

WRITEUP HackToday 2023

Bude Jiang Society

DAFTAR ISI

welcome

- Welcome (again)

rev

- OnlyAdminCanSee
- kurang-lebih+

web

- LogInspek

mis

- DCHEZKIBOXS
- Where is my git?

for

- Doodled

osint

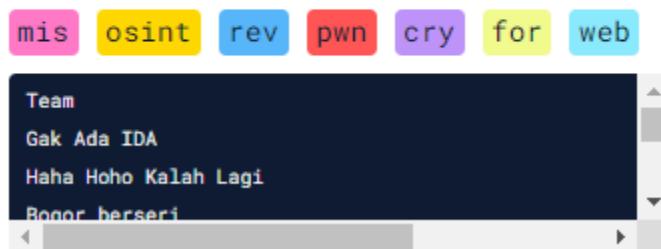
- MUA

Welcome (again)

Welcome (again)

x

check out the discord



1. Sesuai deskripsi "check discord"
2. Flag ditemukan pada text channel "#announcement"

A screenshot of the "#announcement" text channel in a Discord server. The channel has a dark theme. A message from user "arai" at 8:37 AM says: "Hallo semua ohayou~ untuk soal foren ada soal yang ukuran nya 200 mb bisa download terlebih dahulu sekarang untuk password akan di share ketika lomba sudah mulai" followed by a link: https://mega.nz/file/jdYgVaYb#UildAghTAEt7K8bcZzGFYnlsHKF_Qc4WebUQ_t3HRZ8. Below it, a message from user "@Peserta" at 9:00 AM says: "201.43 MB file on MEGA" and includes a thumbnail of a file icon. A red arrow points to a message from user "M.A.R.U BOT" at 9:00 AM which contains the flag: "HackToday 2023 Qualification Stage has started! hacktoday{maru_stands_for_molecular_atomic_reconstructed_unit}".

FLAG :

`hacktoday{maru_stands_for_molecular_atomic_reconstructed_unit}`

REVERSE ENGINEERING

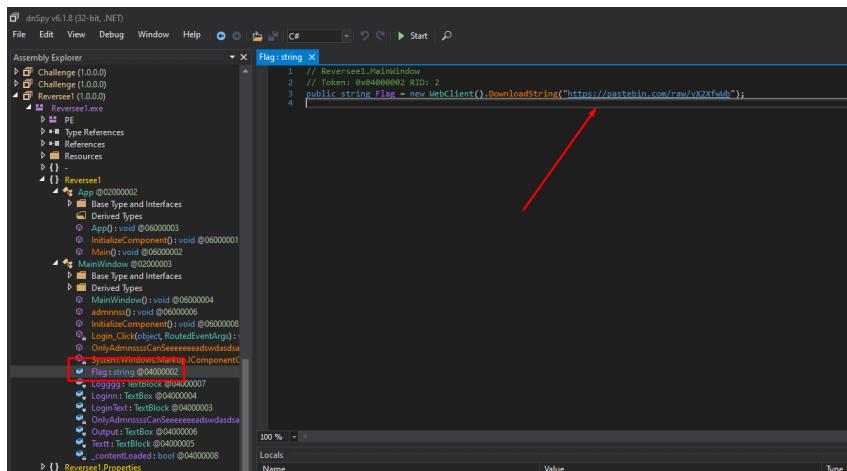
OnlyAdminCanSee

OnlyAdminCanSee x

```
my friend sent a file to me and he
asked me to hack an application that
Mr. John can login. Can you help me
reversing it so im able to see what
is inside? Help him to login to the
app
```



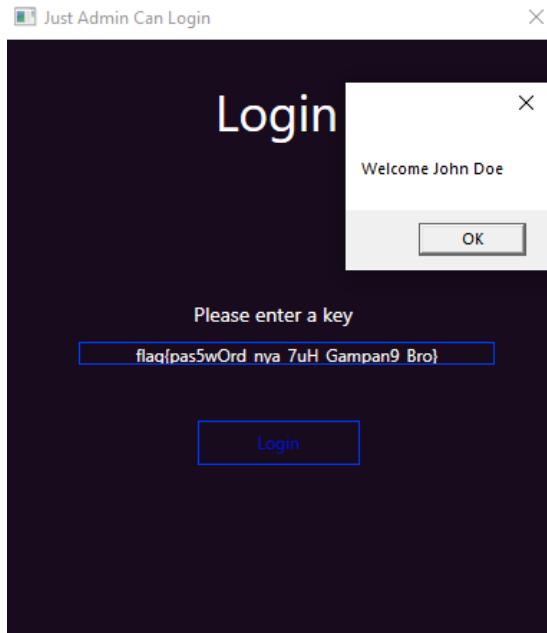
1. Pada soal ini, diberikan file berupa exe, yang ternyata merupakan .NET. Hal ini diketahui ketika saya berusaha untuk melakukan decompile dengan IDA
2. Karena disini soal dari .NET, maka saya mencoba membukanya dengan dnSPY 32 bit.
3. Disini ditemukan adanya fungsi dengan nama "Flag"



URL = <https://pastebin.com/raw/yX2XfwWb>

4. Didapati dari URL = flag{pas5wOrd_nya_7uH_Gampan9_Bro}

Karena disini format seharusnya adalah hacktoday{*}, maka saya coba untuk menganalisa lagi, dan ternyata ini merupakan password



5. Karena ini bukan flag, dan sedari tadi ditegaskan kata "admin", maka disini saya mencoba menganalisa ulang, dan menemukan URL lagi yaitu <https://pastebin.com/raw/VWgc4jWn>. URL ini didapati dari fungsi **OnlyAdmnssssCanSeeeeeeeadsdswdasdsasdsfasdsads()**

```
7 using system.Windows.Controls;
8 using system.Windows.Markup;
9
10 namespace Reverse1
11 {
12     // Token: 0x02000003 RID: 3
13     public class MainWindow : Window, IComponentConnector
14     {
15         // Token: 0x00000004 RID: 4 RVA: 0x00002094 File Offset: 0x00000294
16         public MainWindow()
17         {
18             this.InitializeComponent();
19             this.Output.Visibility = Visibility.Hidden;
20             this.Logooo.Visibility = Visibility.Hidden;
21         }
22
23         // Token: 0x00000005 RID: 5 RVA: 0x000020E8 File Offset: 0x000002E8
24         public void OnlyAdmnssssCanSeeeeeeeadsdswdasdsasdsfasdsads()
25         {
26             bool onlyAdmnssssCanSeeeeeeeadsdswdasdsasdsfasdsads = this.OnlyAdmnssssCanSeeeeeeeadsdswdasdsasdsfasdsads;
27             if (onlyAdmnssssCanSeeeeeeeadsdswdasdsasdsfasdsads)
28             {
29                 this.Output.Visibility = Visibility.Visible;
30                 this.Logooo.Visibility = Visibility.Visible;
31                 string text = new WebClient().DownloadString("https://pastebin.com/raw/VWgc4jWn");
32                 this.Output.Text = text;
33             }
34         }
35
36         // Token: 0x00000006 RID: 6 RVA: 0x0000213C File Offset: 0x0000033C
37         public void admnssss()
38         {
39             MessageBox.Show("Welcome John Doe");
40             this.LoginText.Text = "John The Admnssss";
41             this.Text.Text = "Pw=" + this.Password;
42         }
43     }
44 }
```

6. Didapati string =

"BOPCdFDk\uH\$_q5FA=W6?U\fgDJ*<4H=(GEDI7ZG?Y;32?ZU@21h^601h\^Z1h\^o" yang merupakan Base85

7. Decode dengan cyberchef, dan flag ditemukan

The screenshot shows the CyberChef interface with a 'From Base85' recipe. The input is a long string of encoded characters: BOPCdfDK\u0H\$_q5FA=w6?U\fgD]*<4H=(GEDI7ZG?Y;32?ZU@21h^601h\^Z1h\^o. The output is the decoded ASCII string: hacktoday{D0tN3t_Em4ng_3z_k4n_y4_g4k_sus4h_h4h4h4h4}.

FLAG :

hacktoday{D0tN3t_Em4ng_3z_k4n_y4_g4k_sus4h_h4h4h4h4}

kurang-lebih+

kurang-lebih+

x

```
siapa yang benci soal beranak??, well  
here we go  
-  
HINT : If something goes wrong why  
don't just debug it ^-^  
-  
NOTE : hacktoday{String yang  
ditemukan}
```

rev	attachment
Team	Submitted
sehad	09:48:45 26/08/2023 WIB
Gak Bahaya Ta?	14:41:45 26/08/2023 WIB
Kessoku Band	12:57:54 26/08/2023 WIB
SHA-507	14:19:55 26/08/2023 WIB

1. Pada soal ini, diberikan file berupa python, disini sebenarnya sudah di simplify dan dicoba dengan z3, namun tetap gagal. Maka dari itu didapati pattern sebagai berikut:

Line 1 $A - B + C + D - E$

Line 2 $A + B - C - D + E$

Line 3 $A - B - C + D + E$

Line 4 $A + B + C - D - E$

Line 5 $A - B + C - D + E$

Line 6 $A + B - C + D - E$

Line 7 $A - B - C - D - E$

Line 8 $A + B + C + D + E$

Disini bisa dicari satu persatu dengan metode aljabar:

Contohnya

$$A = (Line1+Line2)/2$$

$$B = (\text{Line2} + \text{Line4} + \text{Line6} + \text{Line8})/4 - A$$

$$C = (\text{Line4} + \text{Line8})/2 - (A+B)$$

$$D = (\text{Line6} + \text{Line8})/2 - (A+B)$$

2. Dari situ saya menggabungkan script z3 dengan perhitungan deklarasi diatas, berikut scriptnya:

```
from z3 import *

flag = [BitVec(f'flag_{i}', 8) for i in range(40)]
solver = Solver()
solver.add((flag[22]) == 40)
solver.add((flag[26]) == 32)
solver.add((flag[9]) == 122)
solver.add((flag[19]) == 117)
solver.add((flag[36]) == 36)
solver.add((flag[2]) == 98)
solver.add((flag[5]) == 105)
solver.add((flag[24]) == 111)

solver.add((flag[28]) == (105 + 127 - 25 + 385) / 4 - 40)
solver.add((flag[0]) == (83 + 101 - 29 + 393) / 4 - 32)
solver.add((flag[21]) == (49 + 21 + 71 + 475) / 4 - 122)
solver.add((flag[18]) == (122 + 98 + 100 + 564) / 4 - 117)
solver.add((flag[38]) == (-96 + 44 + 8 + 332) / 4 - 36)
solver.add((flag[20]) == (147 + 155 + 43 + 463) / 4 - 98)
solver.add((flag[25]) == (123 + 237 + 115 + 409) / 4 - 105)
solver.add((flag[8]) == (120 + 132 - 40 + 420) / 4 - 111)

solver.add((flag[4]) == (127 + 385) / 2 - (105 + 127 - 25 + 385) / 4)
solver.add((flag[23]) == (101 + 393) / 2 - (83 + 101 - 29 + 393) / 4)
solver.add((flag[37]) == (21 + 475) / 2 - (49 + 21 + 71 + 475) / 4)
solver.add((flag[6]) == (98 + 564) / 2 - (122 + 98 + 100 + 564) / 4)
solver.add((flag[33]) == (44 + 332) / 2 - (-96 + 44 + 8 + 332) / 4)
solver.add((flag[7]) == (155 + 463) / 2 - (147 + 155 + 43 + 463) / 4)
solver.add((flag[27]) == (237 + 409) / 2 - (123 + 237 + 115 + 409) / 4)
solver.add((flag[15]) == (132 + 420) / 2 - (120 + 132 - 40 + 420) / 4)
```

```

solver.add((flag[35])==(-25+385)/2 - (105 + 127-25+385)/4)
solver.add((flag[11])==(-29+393)/2 - (83 + 101-29+393)/4)
solver.add((flag[34])==(71+475)/2 - (49 +21+71+475)/4)
solver.add((flag[13])==(100+564)/2 - (122 +98+100+564)/4)
solver.add((flag[32])==(8+332)/2 - (-96 +44+8+332)/4)
solver.add((flag[10])==(43+463)/2 - (147+155+43+463)/4)
solver.add((flag[39])==(115+409)/2 - (123+237+115+409)/4)
solver.add((flag[31])==(-40+420)/2 - (120+132-40+420)/4)

#line 1
solver.add((flag[22]-flag[28]+flag[4]+flag[35]-flag[29])===-25)
solver.add((flag[26]-flag[0]+flag[23]+flag[11]-flag[16])===-19)
solver.add((flag[9]-flag[21]+flag[37]+flag[34]-flag[14])==195)
solver.add((flag[19]-flag[18]+flag[6]+flag[13]-flag[12])==112)
solver.add((flag[36]-flag[38]+flag[33]+flag[32]-flag[3])==168)
solver.add((flag[2]-flag[20]+flag[7]+flag[10]-flag[30])==49)
solver.add((flag[5]-flag[25]+flag[27]+flag[39]-flag[17])==87)
solver.add((flag[24]-flag[8]+flag[15]+flag[31]-flag[1])==102)

#line 2
solver.add((flag[22]+flag[28]-flag[4]-flag[35]+flag[29])==105)
solver.add((flag[26]+flag[0]-flag[23]-flag[11]+flag[16])==83)
solver.add((flag[9]+flag[21]-flag[37]-flag[34]+flag[14])==49)
solver.add((flag[19]+flag[18]-flag[6]-flag[13]+flag[12])==122)
solver.add((flag[36]+flag[38]-flag[33]-flag[32]+flag[3])==96)
solver.add((flag[2]+flag[20]-flag[7]-flag[10]+flag[30])==147)
solver.add((flag[5]+flag[25]-flag[27]-flag[39]+flag[17])==123)
solver.add((flag[24]+flag[8]-flag[15]-flag[31]+flag[1])==120)

#line 3
solver.add((flag[22]-flag[28]-flag[4]+flag[35]+flag[29])===-47)
solver.add((flag[26]-flag[0]-flag[23]+flag[11]+flag[16])==37)
solver.add((flag[9]-flag[21]-flag[37]+flag[34]+flag[14])==223)
solver.add((flag[19]-flag[18]-flag[6]+flag[13]+flag[12])==136)
solver.add((flag[36]-flag[38]-flag[33]+flag[32]+flag[3])==28)
solver.add((flag[2]-flag[20]-flag[7]+flag[10]+flag[30])==41)
solver.add((flag[5]-flag[25]-flag[27]+flag[39]+flag[17])==-27)
solver.add((flag[24]-flag[8]-flag[15]+flag[31]+flag[1])==90)

```

```

#line 4
solver.add((flag[22]+flag[28]+flag[4]-flag[35]-flag[29])==127)
solver.add((flag[26]+flag[0]+flag[23]-flag[11]-flag[16])==101)
solver.add((flag[9]+flag[21]+flag[37]-flag[34]-flag[14])==21)
solver.add((flag[19]+flag[18]+flag[6]-flag[13]-flag[12])==98)
solver.add((flag[36]+flag[38]+flag[33]-flag[32]-flag[3])==44)
solver.add((flag[2]+flag[20]+flag[7]-flag[10]-flag[30])==155)
solver.add((flag[5]+flag[25]+flag[27]-flag[39]-flag[17])==237)
solver.add((flag[24]+flag[8]+flag[15]-flag[31]-flag[1])==132)

#line 5
solver.add((flag[22]-flag[28]+flag[4]-flag[35]+flag[29])==105)
solver.add((flag[26]-flag[0]+flag[23]-flag[11]+flag[16])==93)
solver.add((flag[9]-flag[21]+flag[37]-flag[34]+flag[14])==173)
solver.add((flag[19]-flag[18]+flag[6]-flag[13]+flag[12])==134)
solver.add((flag[36]-flag[38]+flag[33]-flag[32]+flag[3])==64)
solver.add((flag[2]-flag[20]+flag[7]-flag[10]+flag[30])==153)
solver.add((flag[5]-flag[25]+flag[27]-flag[39]+flag[17])==95)
solver.add((flag[24]-flag[8]+flag[15]-flag[31]+flag[1])==262)

#line 6
solver.add((flag[22]+flag[28]-flag[4]+flag[35]-flag[29])==-25)
solver.add((flag[26]+flag[0]-flag[23]+flag[11]-flag[16])==-29)
solver.add((flag[9]+flag[21]-flag[37]+flag[34]-flag[14])==71)
solver.add((flag[19]+flag[18]-flag[6]+flag[13]-flag[12])==100)
solver.add((flag[36]+flag[38]-flag[33]+flag[32]-flag[3])==8)
solver.add((flag[2]+flag[20]-flag[7]+flag[10]-flag[30])==43)
solver.add((flag[5]+flag[25]-flag[27]+flag[39]-flag[17])==115)
solver.add((flag[24]+flag[8]-flag[15]+flag[31]-flag[1])==-40)

#line 7
solver.add((flag[22]-flag[28]-flag[4]-flag[35]-flag[29])==-305)
solver.add((flag[26]-flag[0]-flag[23]-flag[11]-flag[16])==-329)
solver.add((flag[9]-flag[21]-flag[37]-flag[34]-flag[14])==-231)
solver.add((flag[19]-flag[18]-flag[6]-flag[13]-flag[12])==-330)
solver.add((flag[36]-flag[38]-flag[33]-flag[32]-flag[3])==-260)
solver.add((flag[2]-flag[20]-flag[7]-flag[10]-flag[30])==-267)
solver.add((flag[5]-flag[25]-flag[27]-flag[39]-flag[17])==-199)
solver.add((flag[24]-flag[8]-flag[15]-flag[31]-flag[1])==-198)

```

```

#line 8

solver.add((flag[22]+flag[28]+flag[4]+flag[35]+flag[29]) == 385)
solver.add((flag[26]+flag[0]+flag[23]+flag[11]+flag[16]) == 393)
solver.add((flag[9]+flag[21]+flag[37]+flag[34]+flag[14]) == 475)
solver.add((flag[19]+flag[18]+flag[6]+flag[13]+flag[12]) == 564)
solver.add((flag[36]+flag[38]+flag[33]+flag[32]+flag[3]) == 332)
solver.add((flag[2]+flag[20]+flag[7]+flag[10]+flag[30]) == 463)
solver.add((flag[5]+flag[25]+flag[27]+flag[39]+flag[17]) == 409)
solver.add((flag[24]+flag[8]+flag[15]+flag[31]+flag[1]) == 420)

if solver.check() == sat:
    model = solver.model()
    result = ''.join([chr(model[flag[i]].as_long()) for i in range(40)])
    print(f"FLAG: {result}")
else:
    print("No")

```

● FLAG: ipb.link/z3-zolve-huh (not flag btw \$^\$)

Output : ipb.link/z3-zolve-huh (not flag btw \$^\$)

3. Link diatas mengarahkan ke editor g docs, dan kita bisa menemukan file aslinya melalui history
4. Ditemukan file **brainfuck / bf** sebagai berikut



The screenshot shows a Google Docs document titled "FLAG???.txt". The content of the document is a long string of Brainfuck code, which is a compressed representation of the challenge's logic. The code consists of various Brainfuck instructions (like >, <, +, -, ., ,) repeated many times in a specific pattern to solve the puzzle.

5. Dari sini saya coba convert dari bf ke c menggunakan [bftoc](#) agar bisa kita analisa lebih lanjut

Berikut konversinya

6. Sekarang kita tinggal mendebug file berikut

Dari sini kita bisa melihat bahwa hasil akhir `tape[ptr]` adalah -15 nah apabila kita lihat karakter ke 15 dari alphabet adalah p, seperti nya cek disini mengecek apabila input kita sama dengan sebuah angka dan bila iya, makan akan di bilang benar atau salah

Dan bila hasil nya tidak sama

```
while (tape[ptr] != 0)
{
    while (tape[ptr] != 0)
    {
        tape[ptr] -= 1;
    }
    ptr += 1;
    tape[ptr] -= 1;
    ptr -= 1;
}
```

Akan terjadi pengurangan dan hasil nya adalah -1 bila salah

7. Dari analisis tersebut kita tinggal melakukan beberapa patching terhadap script bf yang sudah di convert ke c, karena kita tahu bahwa setiap di bagian akhir akan disimpan berupa jarak antara karakter yang kita masukan dan karakter yang menjadi flag nya, kita tinggal menghitung dengan cara, (karakter yang kita masukan + jarak*-1) lalu kita print dengan %c agar mengeluarkan karakter nya seperti berikut

```
tape[ptr] = -1;
int debug = 1;
printf("%c", (tape[ptr] * -1) + 97);
tape[ptr] = 0;

while (tape[ptr] != 0)
{
    while (tape[ptr] != 0)
```

Di sini karena karakter saya gunakan adalah a, maka rumus nya menjadi $(jarak^*-1) + 97$

Lalu code tersebut saya ulangi setiap terdapat getchar, dan sebelum loop berisikan tape[ptr] -= 1, seperti berikut

```
/* It was generated on Saturday, August 26, 2023 at 01:09PM
 */
#include <stdio.h>
void main(void)
{
    int size = 1000;
    char tape[1000];
    int i;
    /* Clearing the tape (array) */
    for (i = 0; i < size; i++)
        tape[i] = 0;
    int ptr = 0;
    tape[ptr] = getchar();
    if (tape[ptr] == 10)
        while (tape[ptr] != 0)
    {
        tape[ptr] -= 1;
        ptr -= 1;
        tape[ptr] -= 10;
        ptr += 1;
    }
    tape[ptr] += 1;
    ptr += 1;
    tape[ptr] += 2;
    int debug = 1;
    printf("%c\n", (tape[ptr] == -1 ? 0 : 0));
    tape[ptr] = 0;
    while (tape[ptr] != 0)
    {
        while (tape[ptr] == 0)
        {
            tape[ptr] -= 1;
        }
        tape[ptr] -= 1;
        ptr -= 1;
        tape[ptr] -= 1;
        ptr += 1;
    }
    ptr -= 1;
    while (tape[ptr] != 0)
    {
        tape[ptr] -= 1;
        tape[ptr] = getchar();
        if (tape[ptr] == 10)
            while (tape[ptr] != 0)
        {
            tape[ptr] -= 1;
            ptr -= 1;
            tape[ptr] -= 10;
            ptr += 1;
        }
        tape[ptr] += 1;
        ptr += 1;
        tape[ptr] += 2;
        .....
        tape[ptr] = 0;
    }
    while (tape[ptr] != 0)
        if (tape[ptr] == 0)
```

8. Setelah melakukan patching tersebut kita tinggal menjalankan script nya

Lalu kita masukan input sebagai berikut...

Tinggal kita enter dan...

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa  
plus_and_m1nus_refers_to_brnfck_h3h3:correct one nice you're not wrong ^_~  
C:\Users\mamadz\source\repos\bf_debugging2\x64\Debug\bf_debugging2.exe (process 27576) exited  
with code 0.  
Press any key to close this window . . .|
```

9. Selesai flag nya adalah

FLAG : hacktoday{plus_and_m1nus_refers_to_brnfck_h3h3}

WEB EXPLOITATION

Loginspek

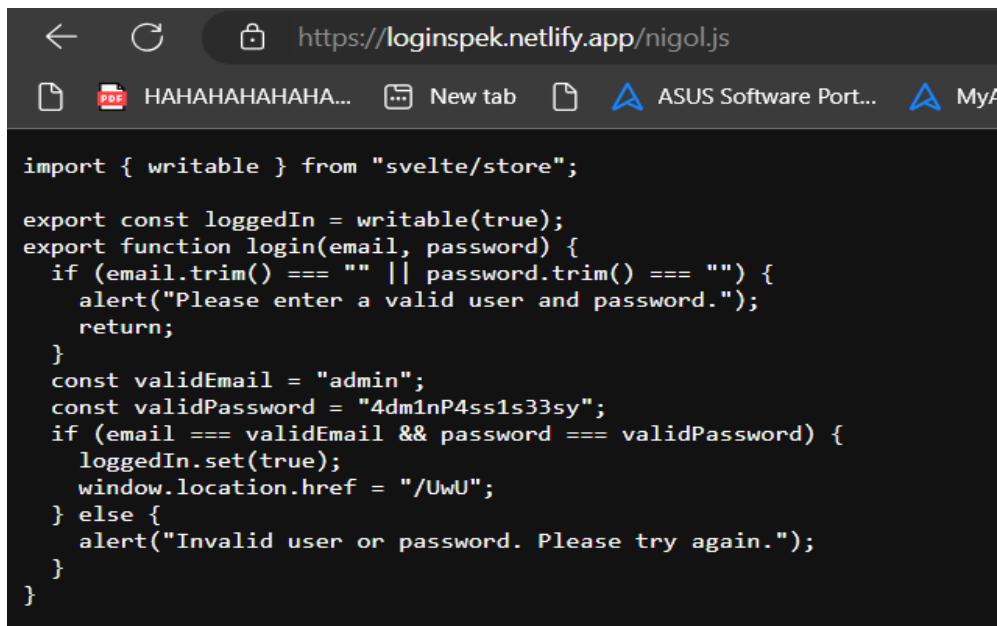
LogInspek

x

Mr. Robot trying to get his revenge to a hacker website and need to login as an admin, but seems like there is no backend? well we dont know. Thats why Mr. Robot asked John the Inspector to help him to get the flag.
<https://loginspek.netlify.app/>



Dari description kita tinggal lihat aja sih sebenarnya kan dia ada tulisan "robots" langsung aja check robots.txt kan ada yg di disallow tuh, nigol.js, setelah dibuka ternyata ada creds



A screenshot of a browser window displaying the source code of a file named "nigol.js". The code is written in JavaScript and contains logic for logging in. It checks if the email and password are valid, comparing them against hardcoded values ("admin" and "4dm1nP4ss1s33sy"). If they match, it sets a boolean variable "loggedIn" to true and changes the page URL to "/UwU". Otherwise, it alerts the user that the credentials are invalid.

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

export const loggedIn = writable(true);
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.");
    }
}
```

Login pake creds tadi dan ktemu flagnya dalam base64

[HOME](#) [LOGIN](#)

Welcome to the Admin Page!

There is something here to be honest

This page can only be accessed after successful login.

NBQWG23UN5SGC6L3GF2HUX3KOU2XIXZRNY2XAM3DORPTK23JNRWHGX3COIYH2==

FLAG : hacktoday{1tz_ju5t_1n5p3ct_5kills_br0}

MISC

Where is my git

Where is my git? x

I just playing around with git command, and suddenly, my flag (i mean, my git) is disappear. Can you find it for me?



Download attachmentnya, lalu liat langsung kebagian logs, main kan ada banyak tuh commit-hashnya. Hilangin semua yang ada "remove", jadi kita fokus di "add" aja, udah langsung pake script

```
1 import subprocess
2
3 with open("./input.txt", "r") as f:
4     input_content = f.read()
5
6 lines = input_content.split("\n")
7 commit_hashes = [line.split()[1] for line in lines if len(line.split()) > 1]
8
9 output_characters = []
10 for commit_hash in commit_hashes:
11     git_show_output = subprocess.check_output(["git", "show", commit_hash]).decode("utf-8")
12     lines = git_show_output.split("\n")
13     for line in lines:
14         if line.startswith("+") and len(line) > 1:
15             output_characters.append(line[1])
16
17 joined_characters = "".join(output_characters)
18
19 print(joined_characters.replace('+', ''))
```

FLAG : Hacktoday{thank_you_for_finding_my_flag_from_this_git_1an23nfa}

DCHEZKIBOXZ

DCHEZKIBOXS

x

Kali ini Pak Masse menemukan sebuah port aneh yang berisi kalimat rahasia. Untuk melihat kalimat tersebut ia diminta memasukan password berupa Substring terpajang pertama dari string yang diberikan port tersebut yang merupakan "kata DCHEZKIBOXS". Sebuah string dikatakan "kata DCHEZKIBOXS" jika dan hanya jika string tersebut dibelah dua secara horizontal kedua bagian string (atas dan bawah) akan membentuk bagian yang seimbang dan dapat dibentuk menjadi sama persis jika beberapa bagianya dirotasi. Contoh: "CHECK" merupakan salah satu "kata DCHEZKIBOXS" karena ketika belah dua kedua bagian akan membentuk bagian yang seimbang. visualisasi contoh dapat dilihat di bagian "attachement". Karena string yang diberikan sangat panjang, untuk menemukan passwordnya Pak Masse menggeser-geser jendela ruangannya atau memainkan dua buah pointer miliknya untuk mendapatkan berbagai inspirasi.
nc 103.181.183.216 19001



Dikasih NC yekan ke arah soalnya, nah tiap kali kita connct, kita itu diberikan sebuah string yang panjang banget, untuk mempermudah sih kita tinggal cari aja yang di char X ada salah 1 dari char "DCHEZKIBOXS", kalo ada 1, kita check kanannya ada lagi ga dan di save as current_longest, udah deh tinggal print longest, masukin ke password, done kita dpt flag.

```

1  def find_longest_substring(text):
2      longest_substring = ""
3      current_substring = ""
4
5      for char in text:
6          if char in "DCHEZKIBOXS":
7              current_substring += char
8              if len(current_substring) > len(longest_substring):
9                  longest_substring = current_substring
10             else:
11                 current_substring = ""
12
13     return longest_substring
14
15 file_path = "./bom.txt"
16 with open(file_path, "r") as f:
17     input_string = f.read()
18
19 longest_substring = find_longest_substring(input_string)
20
21 print("Longest substring:", longest_substring)

```

DTUUYZPEOYDDOWMJQKGLTZWTCTPTLQPWDCURZAYICKCHAACFNQOFAGSUEAUHSVIHWZHRZKADHQJUDUCJ
CLKERUCOULKWCACMHRHYPBIJEIQUFCIQVYBRLQBGFNNZHGMAKVDAMTCBLXRCAZTPJYOIGZWPFVUXW
TWXRCGRZEKBNYUXPEXGBFFCSUHDDWZISRTGOTDHYWDGQCWIWRZRCSLITYMJGRYNHVORSQKMANKNGOCRW
FCEIJAEZXUJNAUTTHKVQDNGZCEHPEMZBHTTNLXMJUXMNNNGXUWFUFLHSMJBETZPXSWEJODCLYDBRG
IGOOIWJWEJMJKBFMWKHDHFSYJCRNJSOHGXOLDKJZTSOTBDYBNSKRZTDQGVRDKOEHVXPAAWVUTPKYNBI
ZYMWKXNAHOTMPYZAMOKUBOBIAJJVOWMAXOGMLOEWGGMQRTOLNNTJJGNGYDTRKPZJFDLBRBYMLIRSESQ
EZTIIAMNMGMGXCTONAGVJKZDTITNVFOFVQKUJGGCUTAEVOMEWQJUGDETQLTANGQHTQFXCVKHHHSMUI
NOCXQDVIQMNSTBFVGNFVGGSJAVMDFXFNJSNFHZFSOANAFTRGTFVKEUCHNBVJUCHGQDYYAEJJFUSVYN
FQWRXKAGIKGBZRRMWIFACWEYUNRZQZKLXXEOLUGJTFLIXMYARPUDSSXFZHSKOGARXKMLZBRXBLVAEBY

Itu stringku, mana passwordmu?

passwordnya adalah substring terpanjang pertama yang merupakan 'kata DCHEZKIBOXS'.

Sebuah string dikatakan 'kata DCHEZKIBOXS' jika dan hanya jika string tersebut dibelah dua secara horizontal kedua bagian string (atas dan bawah) akan membentuk bagian yang berimbang dan dapat dibentuk menjadi sama persis jika beberapa baginya dirotasi.

password: BEBEEEDDCKEEZ

Congratsss, ini flag buatmu! : hacktoday{Yeyyy_n0w_y0U_kN0w_5Lid1ng_Wind0w5_4l6oR1thMs!}

└─(kiinzu@Kiinzu)-[~/hacktoday]

\$

FLAG : Hacktoday{Yeyyy_n0w_y0U_kN0w_5Lid1ng_Wind0w5_4l6oR1thMs!}

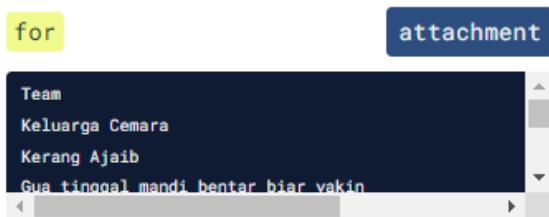
FORENSIC

Doodled

Doodled

x

Oh no! My cousin Gio Ferdiansyah has scribbled all over my stuff! Almost everything is covered with his doodles, including an important QR code, making it unreadable. Can you help me retrieve the lost information?



1. Disini diberikan QR code yang ditutupi dengan beberapa gambar
2. Awalnya saya mencoba untuk menggunakan Adobe Illustrator, namun tetap tidak bisa
3. Selanjutnya saya mencoba untuk membuat ulang dengan tools <https://merri.cx/qrazybox/>, menggunakan ukuran 29x29. Pada bagian yang tidak terlihat saya kosongkan



4. Setelah itu saya coba untuk menggunakan tools "Reed Solomon Decoder"

Reed-Solomon Decoder

Decoded Reed-Solomon blocks :

```
66,230,134,22,54,183,70,246,70,23,151,182,115,3,6,69,2
```

Final data string :

```
hacktoday{g00d_70b_QR_c0d3_r3p41R_5uCC3s5fuL!}
```

[Close](#)

[Back](#)

FLAG : `hacktoday{g00d_70b_QR_c0d3_r3p41R_5uCC3s5fuL!}`

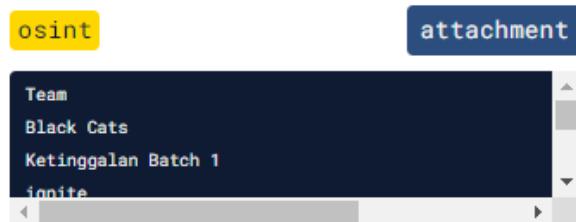
OSINT

MUA

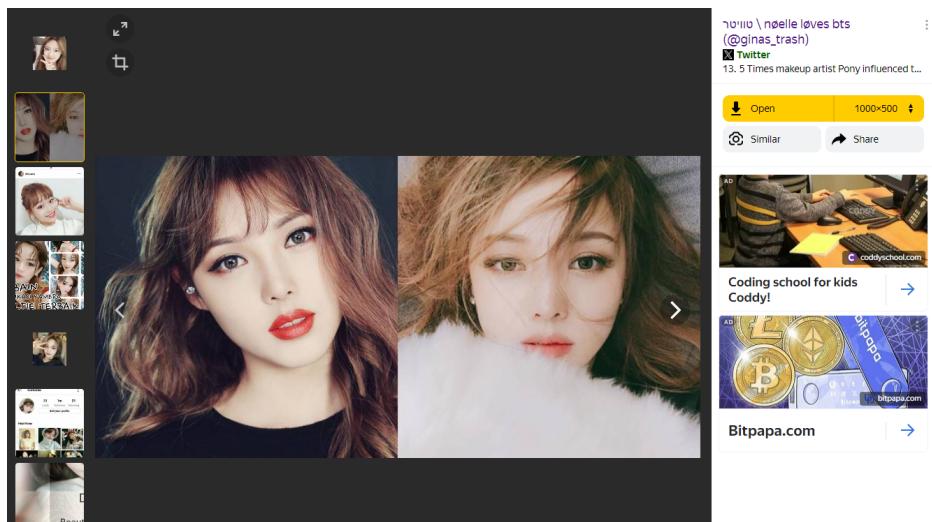
MUA

x

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 menyemangatinya di salah satu video di channel youtubennya. Video itu berisi vlog dia sedang menggambar.



1. Pertama coba cari menggunakan YANDEX, dan ditemukan ada 1 yang mirip dengan website berbahasa korea

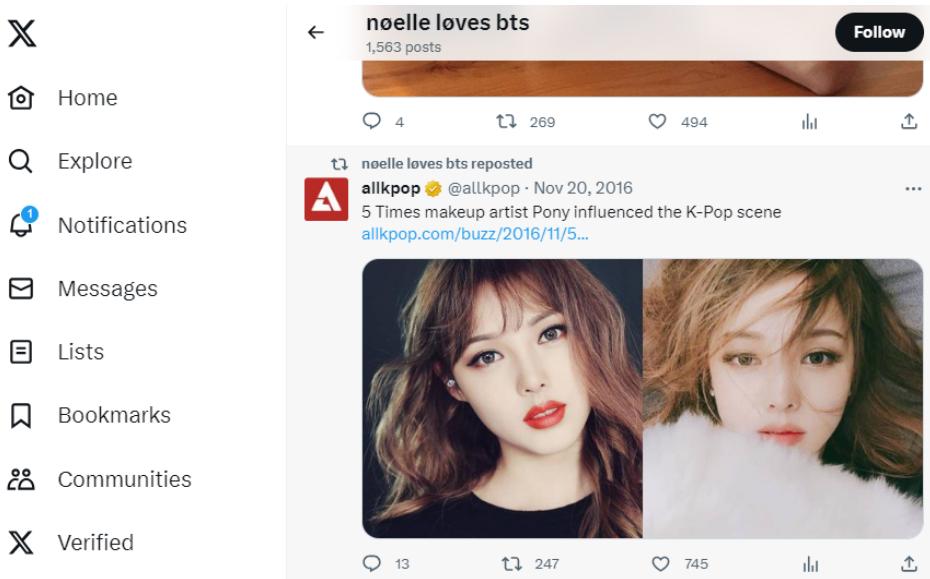


2. Link ini diarahkan ke twitter, dan ternyata mengarah ke website berita tentang korean make up artist.

Link twitter : <https://twitter.com/allkpop/status/800052695117811713>

Link URL :

<https://www.allkpop.com/buzz/2016/11/5-times-makeup-artist-pony-influenced-the-k-pop-scene>



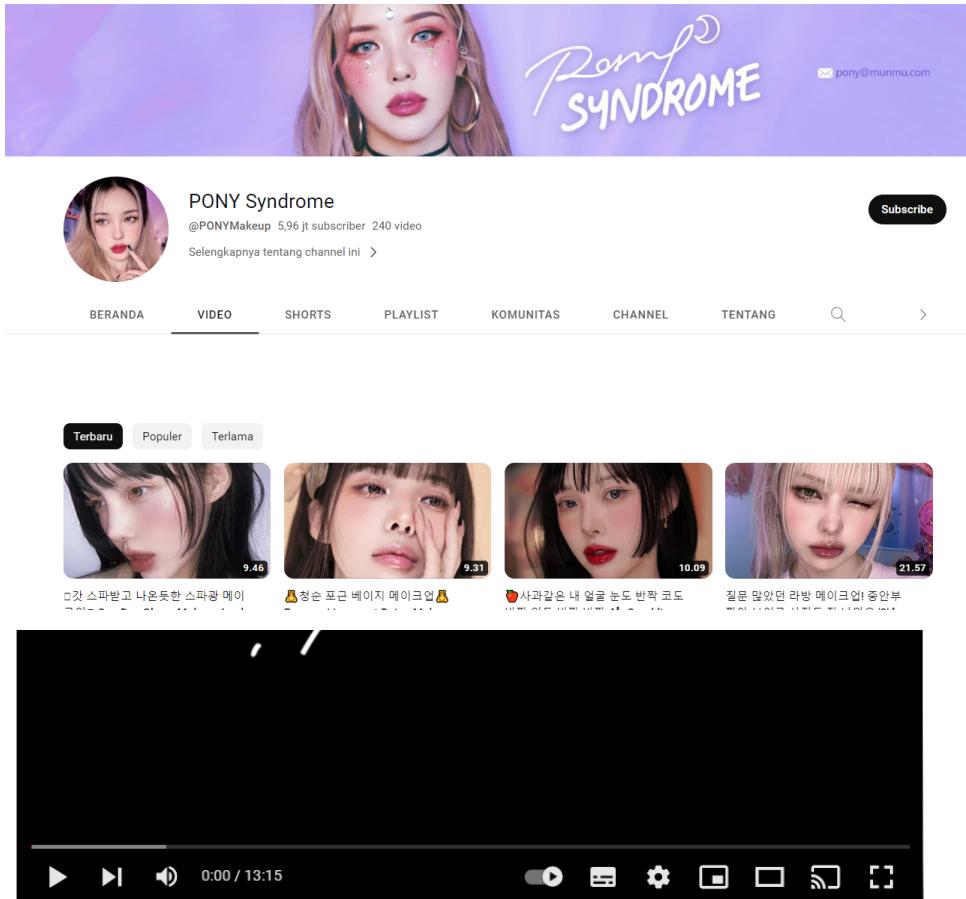
3. Ditemukanlah nama dari MUA adalah "Pony", karena menurut deskripsi MUA tersebut memiliki YouTube yang merekam dirinya sedang melukis, maka dicarilah channel YouTube dari MUA tersebut

4. Ditemukanlah channelnya dan salah satu videonya melukis

Link : <https://www.youtube.com/watch?v=Hcst57-v9XU>

Dan pada komentar ditemukan encrypted string :

NB2HI4DTHIXS62LOON2GCZ3SMFWS4Y3PNUXWWYTCNE2TQP3VORW
V643POVZGGZJ5OFZCM2LHONUGSZB5JV5E43COI5HGWWWSXKE2E2ZZF
GNCKKM2E



[약간ASMR] 그냥 잠이 안 와서 그린 그림 Color Pencil Drawing 🎨

123 rb x ditonton 1 tahun yang lalu
안녕하세요 여러분
약간ASMR로 찍어 봤는데 편집하다가 몇 번을 잠들었는지 모르겠어요
여러분도 안녕히 주무세요,, ...lainnya

174 Komentar Urutkan

Tambahkan komentar...

@osiosiosi707 3 minggu yang lalu
I love your art 😊😊😊❤️
NB2HI4DTHIXS62LOON2GCZ3SMFWS4Y3PNUXWWYTCNE2TQP3VORWV643POVZGGJ50FZCM2LHONUGS
ZB5JV5E43CO15HGWWXKE2E2ZZFGNCCKM2E

5. Didapati jenis encrypt nya adalah Base32, setelah di decrypt dengan cyberchef, ditemukan URL baru :
https://instagram.com/kbbi58?utm_source=qr&igshid=MzNINGNkZWQ4Mg%3D%3D
6. Ditemukan profil IG dengan 3 postingan dan hint pada bio nya



kbbi58 [Follow](#) +
3 posts 0 followers 1 following
anoo
Assemble it!!!
1-2-3 or 3-2-1

POSTS

TAGGED



7. Percobaan pertama adalah 3-2-1, karena di setiap deskripsi foto ada encrypted string
8. Gabungkan string menjadi :
4q5XuSBsg5UL4rudvSFUW8BQDMztdoPzy7frPxfnGSBah8q48nc9ZcQuqUKXGXWqwz
9. Setelah di decrypt dengan cyberchef, flag ditemukan

Recipe

From Base58

Alphabet
123456789ABCDEFHJKLMNOPQRSTUVWXYZ...

Remove non-alphabet chars

Input

4q5XuSBsg5UL4rudvSFUW8BQDMztdoPzy7frPxfnGSBah8q48nc9ZcQuqUKXGXWqwz

Output

hacktoday{S0_y0u_f0uNd_m3_on_1nst46r4M_hwehwe11}

FLAG : **hacktoday{S0_y0u_f0uNd_m3_on_1nst46r4M_hwehwe11}**