

WRITEUPS ARKAVIDIA 7.0



ARKAVIDIA

INFORMATICS & IT FESTIVAL

TIM

Rahasia

: Tim Mamber :

insomn14
Scenezy
HoneyFun

Reverse Engineering

aksarajawa

Challenge 29 Solves ×

aksarajawa
344

Bahkan orang Bandung pun belajar aksara jawa.

Author: suggoitanoshi

attachment:
<https://drive.google.com/file/d/1vhTKtLXsW1P0D22LnDDgBjBhOusp/sharing>

Flag Submit

Solution:

Diberikan sebuah file zip berisi file *trial.js* yang berisi seperti berikut.

```
trial.js+ buffers
1 const read = (s) => s.reduce((a, b) => (a<<1)+b);
2
3 function argue(a, b, c){
4   if(![][+[]][b-2*a] != 'a' ||
5     ({})+[]).split(' ')[+[]][c-a] != 'b' ||
6     ({})+[]).split(' ')[+[]][b-c] != 'c'){
7     console.log('That\'s wrong!');
8     return;
9   }
10  const ka = [a-read(+!![],+[],+!![])] + [String.fromCharCode(b*10-a-c)] + [a-re
    ad(+!![])];
11  const kb = ['p'] + (({}+[])+[]).substring(+[], read(+!![],+[]))] + [b-2*a] +
    ['hm'] + [c-a+ +!![]] + [({}+[])+[]][+!![]] + [c - +!![]];
12  const kc = [({}+[])[a]] + [c-a- +!![]] + ['m'] + [2*c - b];
13  console.log('Your flag: Arkav7${ka}_${kb}_${kc}');
14 }
```

Jika diperhatikan kode tersebut cukup sulit dibaca bagi saya, jadi saya menggunakan console yang ada pada google chrome untuk membantu dalam menyelesaikan tantangan tersebut, dan didapat hasil seperti berikut.

```
> (![][+[]])
< "false"
> ({})+[]).split(' ')[+[]]
< "[object]"
```

Setelah mengetahui bahwa ekspresi tersebut mengembalikan nilai string saya berkesimpulan bahwa ekspresi selanjutnya adalah untuk mengakses index dari string yang didapat, pada kasus ini kita harus mendapatkan karakter dari 'a' pada string "false" dan karakter 'b' & 'c' pada string "[object]".

Berikut ini adalah solvernya,

```
>>> from z3 import *
>>> def done(a,b,c):
...     s=Solver()
...     s.add(abc[1]-2*abc[0] == a, abc[2] - abc[0] == b, abc[1] - abc[2] == c)
...     s.check()
...     print(s.model())
>>> done(1,2,5)
[i_0 = 6, i_2 = 8, i_1 = 13]
```

```
< undefined
> argue(6, 13, 8)
Your flag: Arkav7{1t5_pun1hm3n7_t1m3}
< undefined
```

Flag : **Arkav7{1t5_pun1hm3n7_t1m3}**

Arcadevidia

Challenge 6 Solves X

Arcadevidia

481

Take it easy, let's play a game!

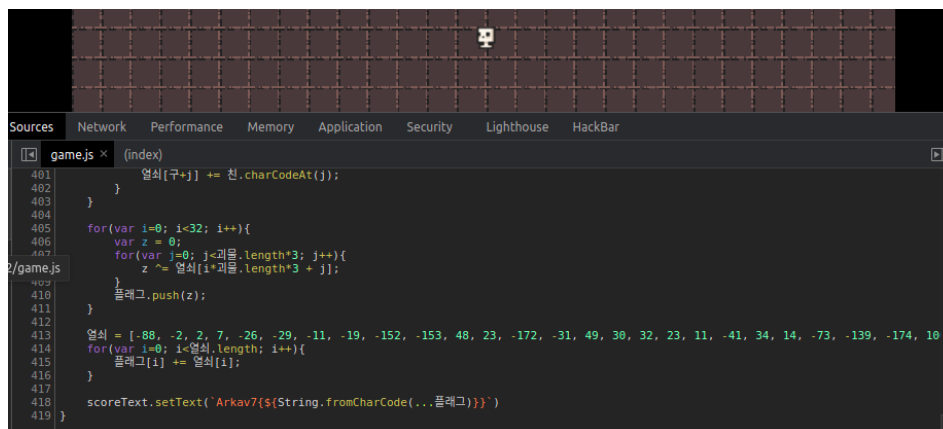
<http://slave3.ctf.arkavidia.id:10032>

Author: Fraglantia

Flag Submit

Solution:

Pada tantangan ini kita diberikan sebuah link, yang mana kita dapat memainkan game tersebut pada halaman utama web tersebut, kita juga dapat melihat source code dari game tersebut `<Ctrl+Shift+i>` pada menu *sources*. Terdapat file *game.js*



Setelah menganalisis dan melakukan sedikit perubahan pada file *game.js*, terdapat beberapa fungsi yang menarik perhatian saya diantaranya. “**createMonster(name, vel, prox)**”, “**checkSecret()**”, dan “**win()**”.

“**CreateMonster()**” berfungsi untuk buat entity atau karakter dari monster yang ada pada game tersebut, berikut ini adalah karakter-karakternya,

```
126   createMonster(this, 'imp', 80, 150);
127   createMonster(this, 'demon', 70, 150);
128   createMonster(this, 'ogre', 40, 300);
129   createMonster(this, 'skeleton', 90, 200);
130   createMonster(this, 'swampy', 75, 220);
131   createMonster(this, 'zombie', 60, 250);
```

“**checkSecret()**” berfungsi untuk melakukan pengecekan terhadap *score* dan entity / monster yang telah berhasil kita bunuh. Monster yang berhasil kita bunuh entity akan di simpan kedalam list **monstersKilled**, dimana entity tersebut berisi **name**, **velocity**, dan **proximity**.

```

361 function checkSecret(){
362
363     if(score != 600) return;
364
365     const mkilld = monstersKilled;
366     const secret = "아르카비아"; // Arcavidia
367
368     if(String.fromCharCode((mkilld[0].proximity * (mkilld[0].velocity + mkilld[0].name.charCodeAt(0) + 20)), ((mkilld[1].name.charCodeAt(0) -
64).toString().repeat(2) + mkilld[1].proximity/50), (Number(mkilld[2].proximity + "" + mkilld[2].velocity) * 2 + mkilld[2].name.
charCodeAt(0) * 75 + 77), (mkilld[3].name.charCodeAt(0) * mkilld[3].name.charCodeAt(1) * 4 + (mkilld[3].proximity + mkilld[3].velocity
) * 16 - 864), (mkilld[4].name.split("").sort().join("").charCodeAt(1) * 256 + (mkilld[4].proximity + mkilld[4].velocity + 30) * 82),
((mkilld[5].name.charCodeAt(4) + mkilld[5].proximity*8/5) * 100)) == enc_flag){
369         win();
370     }

```

Setiap satu karakter dari monster yang kita bunuh entitynya akan dicek apakah menghasilkan string, apakah string tersebut terdapat pada variabel “**secret**” atau tidak. Berikut adalah code solver untuk mengatasi fungsi “**checkSecret()**”.

```

10
11 proximity = [150, 150, 300, 200, 220, 250]
12 velocity = [80, 70, 40, 90, 75, 60]
13 name = ['imp', 'demon', 'ogre', 'skeleton', 'swampy', 'zombie']
14 arkav = "아르카비아"; # Arcavidia
15 mkilld = [None]*len(arkav)
16 # (mkilld[0].proximity * (mkilld[0].velocity + mkilld[0].name.charCodeAt(0) + 20))
17 for n,v,p in zip(name, velocity, proximity):
18     tmp = p * (v + ord(n[0]) + 20)
19     if tmp == ord(arkav[0]):
20         print(n,v,p,tmp)
21         mkilld[0] = [n,v,p]
22 # ((mkilld[1].name.charCodeAt(0) - 64).toString().repeat(2) + mkilld[1].proximity/50)
23 mkilld[1] = [name[2], velocity[2], proximity[2]]
24 # (Number(mkilld[2].proximity + "" + mkilld[2].velocity) * 2 + mkilld[2].name.charCodeAt(0) * 75 + 77)
25 for n,v,p in zip(name, velocity, proximity):
26     tmp = (int(str(f'{p}{v}')) * 2) + ord(n[0]) * 75 + 77
27     if tmp == ord(arkav[2]):
28         print(n,v,p,tmp)
29         mkilld[2] = [n,v,p]
30 # (mkilld[3].name.charCodeAt(0) * mkilld[3].name.charCodeAt(1) * 4 + (mkilld[3].proximity + mkilld[3].velocity) * 16 - 864)
31 for n,v,p in zip(name, velocity, proximity):
32     tmp = (ord(n[0]) * ord(n[1]) * 4 + (p + v) * 16 - 864)
33     if tmp == ord(arkav[3]):
34         print(n,v,p,tmp)
35         mkilld[3] = [n,v,p]
36 # (mkilld[4].name.split("").sort().join("").charCodeAt(1) * 256 + (mkilld[4].proximity + mkilld[4].velocity + 30) * 82)
37 for n,v,p in zip(name, velocity, proximity):
38     tmp = (ord(sorted(n)[1]) * 256 + (p + v + 30) * 82)
39     if tmp == ord(arkav[4]):
40         print(n,v,p,tmp)
41         mkilld[4] = [n,v,p]
42 # ((mkilld[5].name.charCodeAt(4) + mkilld[5].proximity*8/5) * 100)
43 for n,v,p in zip(name, velocity, proximity):
44     tmp = ((ord(n[4%len(n)]) + p*8//5) * 100)
45     if tmp == ord(arkav[5]):
46         print(n,v,p,tmp)
47         mkilld[5] = [n,v,p]

```

“**win()**” berfungsi untuk menghasilkan flag. Berikut ini adalah code solver untuk menghasilkan *flag*nya, setiap karakter dari flag didapat dari list **enc_flag** dan hasil *hashing* MD5 dari list **mkilld**.

Berikutnya terdapat looping yang melakukan operasi AND dan *Penjumlahan* pada list pada **enc_flag**, kemudian dari list **enc_flag** di XOR dan hasilnya disimpan pada list **tampung**. Pada looping yang terakhir setiap nilai yang ada pada list **tampung** dijumlah dengan list **enc_flag** yang baru.

```

51 enc_flag = [149, 144, 62, 117, 233, 184, 141, 241, 230, 126, 250, 172, 50]
52 for i in range(len(mkilld)):
53     t1= hashlib.md5(mkilld[i][0].encode()).hexdigest()
54     t2 = hashlib.md5(str(mkilld[i][1]).encode()).hexdigest()
55     t3 = hashlib.md5(str(mkilld[i][2]).encode()).hexdigest()
56     t4 = i*96;
57     for j in range(32):
58         enc_flag[t4+j] ^= ord(t1[j])
59         enc_flag[t4+j] += ord(t1[j])
60     t4 += 32
61     for j in range(32):
62         enc_flag[t4+j] ^= ord(t2[j])
63         enc_flag[t4+j] += ord(t2[j])
64     t4 += 32
65     for j in range(32):
66         enc_flag[t4+j] ^= ord(t3[j])
67         enc_flag[t4+j] += ord(t3[j])

```

```

67 enc_flag[t4+j] += ord(t3[j])
68 tampung = []
69 for i in range(32):
70     z = 0
71     for j in range(len(mkilld)*3):
72         z ^= enc_flag[i*len(mkilld)*3 + j]
73     tampung.append(z)
74
75 enc_flag = [-88, -2, 2, 7, -26, -29, -11, -19, -152, -153,
76
77 for i in range(len(enc_flag)):
78     tampung[i] += enc_flag[i]
79
80 flag = ''.join(chr(t) for t in tampung)
81 print("Arkav7{%s}" % flag)

```

```
~/CTFI/2021/A/Q/R/Arcadevidia python solver.py
zombie 60 250 50500
swampy 75 220 52852
imp 80 150 48596
demon 70 150 46356
zombie 60 250 50500
Arkav7{c0ngr4tsS5_y0u_w0n_th3_g4me_ggWp}
```

Link Solver : <https://pastebin.com/e1BPZ9N7>

Flag : **Arkav7{c0ngr4tsS5_y0u_w0n_th3_g4me_ggWp}**

Oncle Jazz

Challenge 4 Solves X


Oncle Jazz

499

You're listening to ... Oncle Jazz

Author: alphaville

attachment:
<https://www.dropbox.com/s/1tlo86zr4ad71mq/chall.zip?dl=0>

 chall.zip

Flag

Submit

Solution:

Diberikan sebuah file zip berisi file *binary* hasil compile dari program *golang*, setelah beberapa menit melakukan dynamic analysis menggunakan IDA demo, terdapat fungsi `"main_main()"`, `"main_oncle()"`, dan `"main_jazz()"`.

Pada fungsi `"main_main()"` saya menemukan sebuah nilai hexadesimal dan string yang mencurigakan. Nilai dengan format hexadesimal saya berinama *part1* kemudian nilai dengan format string saya beri nama *part2*.

```
62 | part1[0] = 0xD081C2E;
63 | *(_OWORD *)&part1[1] = *(_OWORD *)((char *)&qword_4EB2A0 + 4);
64 | qmemcpy(part2, "juyzgoypjmgteeeeeec", sizeof(part2));
```

```
rodata:00000000004EB2A0 qword_4EB2A0 dq 141558130D081C2Eh ; DATA XREF: main_main+32+r
rodata:00000000004EB2A0 dq 1B3A090E010C0909h ; main_main+41+r
rodata:00000000004EB2A8 dq 0A0B3C01h
rodata:00000000004EB2B0 db 0
rodata:00000000004EB2B8 db 0
rodata:00000000004EB2B9 db 0
rodata:00000000004EB2BA db 0
rodata:00000000004EB2BB db 0
rodata:00000000004EB2BC db 0
rodata:00000000004EB2BD db 0
rodata:00000000004EB2BE db 0
rodata:00000000004EB2BF db 0
rodata:00000000004EB2C0 qword_4EB2C0 dq 'pyogzyuj' ; DATA XREF: main_main+50+r
rodata:00000000004EB2C0 dq 0 ; main_main+5F+r
rodata:00000000004EB2C8 aJmgteeeeeec db 'jmgteeeeeec',0
rodata:00000000004EB2D5 align 20h
```

Selanjutnya setelah memasukan inputan, pada fungsi `"main_oncle()"` 20 karakter pertama kita di XOR dengan string `"oncle"`

```
17 | for ( i = 0LL; i < a8; ++i )
18 | {
19 |     char_our_input = *(unsigned __int8 *) (our_input + i);
20 |     if ( (unsigned __int64) (i % 5) >= 5 )
21 |         runtime_panicIndex(v14, char_our_input, 5 * (i / 5), 5, our_input, v9, v13);
22 |     v9 = oncle[i % 5]; // 'o', 'n', 'c', 'l', 'e'
23 |     *(_BYTE *) (v14 + i) = v9 ^ char_our_input;
24 | }
25 | return result;
```

```

while ( v12 < 29 )
{
    v24 = v12;
    v21 = byte_4D5060[v12];
    v22 = byte_4D5352[v12];
    a2 = &v21;
    runtime_mapassign(a1, (unsigned int)&v21, v22, v12, v9, v10, (__int64)&unk_4B3A00, v11, (__int64)&v21, (__int64)v19);
    *v19 = v22;
    v12 = v24 + 1;
    v11 = (__int64)v25;
}

```

Pada fungsi **"main_jazz()"** didapati looping sebanyak 29 kali yang melakukan mapassign, **byte_4D5060** adalah keynya dan **byte_4D5352** adalah valuenya.

Setelah memahami sebagian dari fungsi tersebut, saya membuat solver untuk mendekripsi nilai string dan hexadesimal yang sebelumnya sudah saya temukan. Berikut adalah solvernya.

```

mapping = {
'a': 'u', 'b': '{', 'c': 'p', 'd': 'k', 'e': 'm', 'f': 'f', 'g': 'z', 'h': 'q', 'i':
'}', 'j': 't', 'k': 'h', 'l': 'j', 'm': 's', 'n': 'y', 'o': 'o', 'p': '_', 'q': 'r',
'r': 'v', 's': 'l', 't': 'b', 'u': 'a', 'v': 'n', 'w': 'w', 'x': 'x', 'y': 'i', 'z':
'e', '_': 'g', '{': 'd', '}': 'c'
}
from pwn import xor

k = b'oncle'
part1 = xor(bytes.fromhex('0A0B3C011B3A090E010C0909141558130D081C2E')[::-1], k)

stack = 'pyogzyuj'[::-1]
stack += 'jmgTueeeeeec'
part2 = ''
while len(part2) != len(target):
for m in mapping.keys():
    if mapping[m] == target[len(part2)]:
        part2 += m
print(part1+part2)

```

Flag : **Arkav7{welcome_to_golang_oncle_jazzzzzz}**

Miscellaneous

OSINT-1

Challenge 35 Solves X

OSINT-1 256

Pada tanggal 9 Agustus 2019, sebuah artikel ilmiah dipublikasikan pada jurnal Computing and Software for Big Science. Salah satu penulis dari artikel ilmiah tersebut pernah menjadi narasumber pada suatu podcast. Sekitar tiga bulan setelah artikel tersebut dipublikasikan (1 Nov 2019), penulis ini mengunjungi suatu tempat di belahan bumi selatan. Cari tahu nama tempat tersebut.

Format flag: Arkav7{nama_tempat_menggunakan_snake_case}

Author: didithilmy

Flag

Submit

Solution:

Google-fu

Google Computing and Software for Big Science, ketemu website springer

<https://www.springer.com/journal/41781>

Cek dari urutan release, ketemu artikel ini yang dirilis pada 9 aug 2019

<https://link.springer.com/article/10.1007/s41781-019-0026-3>

Article

Open Access

Rucio: Scientific Data Management

Rucio is an open-source software framework that provides scientific collaborations with the functionality to organize, manage, and access their data at scale. The data can be distributed across heterogeneous d...

Martin Barisits, Thomas Beermann, Frank Berghaus... in *Computing and Software for Big Science* (2019)


» Download PDF (1966 KB) » View Article

Article

Computing and Software for Big Science

Original Article | [Open Access](#) | Published: 09 August 2019

Rucio: Scientific Data Management

[Martin Barisits](#), [Thomas Beermann](#), [Frank Berghaus](#), [Brian Bockelman](#), [Joaquin Bogado](#), [David Cameron](#), [Dimitrios Christidis](#), [Diego Ciangottini](#), [Gancho Dimitrov](#), [Markus Elsing](#), [Vincent Garonne](#), [Alessandro di Girolamo](#), [Luc Goossens](#), [Wen Guan](#), [Jaroslav Guenther](#), [Tomas Javurek](#), [Dietmar Kuhn](#), [Mario Lassnig](#) , [Fernando Lopez](#), [Nicolo Magini](#), [Angelos Molfetas](#), [Armin Nairz](#), [Farid Ould-Saada](#), [Stefan Prenner](#), [Cedric Serfon](#), [Graeme Stewart](#), [Eric Vaandering](#), [Petya Vasileva](#), [Ralph Vigne](#) & [Tobias Wegner](#) [-Show fewer authors](#)

Google judul artikelnya ditambah keyword podcast, ketemu halaman ini.

<https://link.springer.com/article/10.1007/s41781-019-0026-3>



RUCIO SCIENTIFIC DATA MANAGEMENT PODCAST

[www.gcppodcast.com](#) > [post](#) > [episode-145-atlas-with-d...](#) ▾

ATLAS with Dr. Mario Lassnig | Google Cloud Platform Podcast

He is also one of the principal developers of the **Rucio** system for **scientific data management**.

In his previous life, he developed mobile navigation software for ...

You visited this page on 1/30/21.

ATLAS with Dr. Mario Lassnig

Our guest today is [Dr. Mario Lassnig](#), a software engineer working on the ATLAS Experiment at CERN! [Melanie](#) and [Mark](#) put on their physics hats as they learn all about what it takes to manage the petabytes of data involved in such a large research project.

Nama nya ngelink ke akun twitternya Dr. Mario Lassnig, cek post dia pada 1 nov 2019. Dia sedang berada di pawsey computing center, di australia(belahan bawah dunia)



Mario Lassnig @mlasnig · Nov 1, 2019

Beautiful indigenous Australian artwork on the @PawseyCentre HPC!



9



Follow

Pawsey Supercomputing Centre

@PawseyCentre

Supercomputing Centre based in Perth Western Australia

Perth, Western Australia pawsey.org.au Joined October 2009

881 Following 2,189 Followers

Not followed by anyone you're following

Tweets

Tweets & replies

Media

Likes

Flag : Arkav7{pawsey_supercomputing_centre}

Sad and Lonely Friend

Challenge

42 Solves

×

Sad and Lonely Friend

165

You are going to need a **help** from **someone**.

Author: BeforeLast

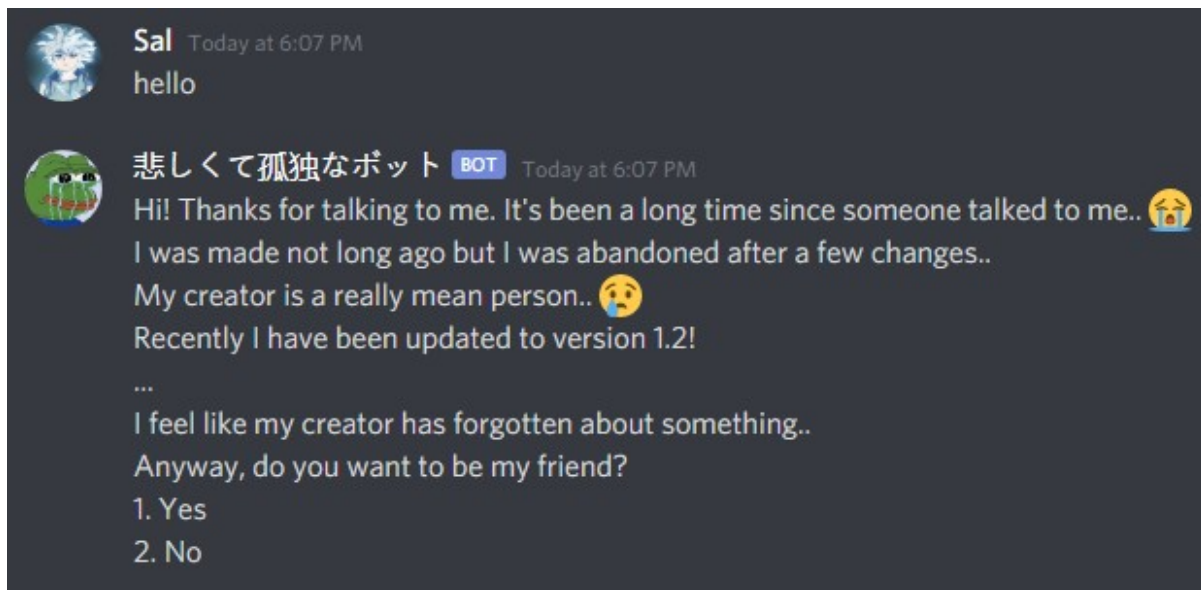
Submit

Solution:

Cara mendapatkan flag nya dengan cara chat dengan bot yang ada di server discord botnya yaitu



Ketika dichat hello akan menjawab

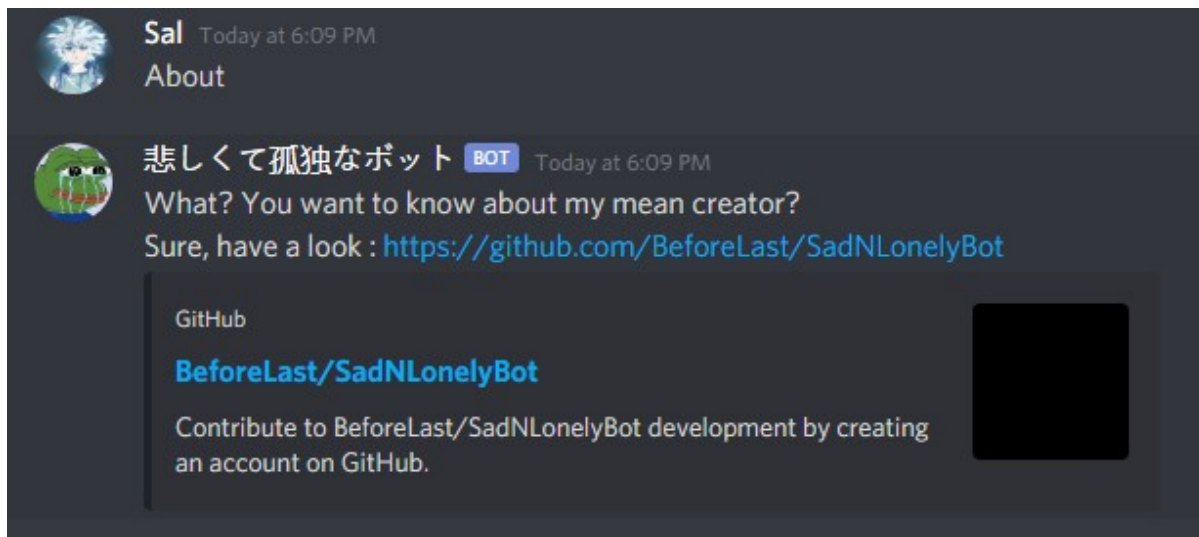


Sepertinya ini bot sedih banget :"

Setelah itu ketik yes, dan lalu ketik help, lalu akan diberikan pilihan lain



Lalu ketik about



Setelah dibuka github nya ternyata isinya adalah sc bot nya tetapi versi terbaru dan dibaca readme nya ternyata ada command "wish me luck" yang dimana ini adalah command yang sebelumnya diganti dengan "i need flag". Karena bot ini v 1.2 maka "wish me luck" belum terganti dengan "i need flag".

readme.md

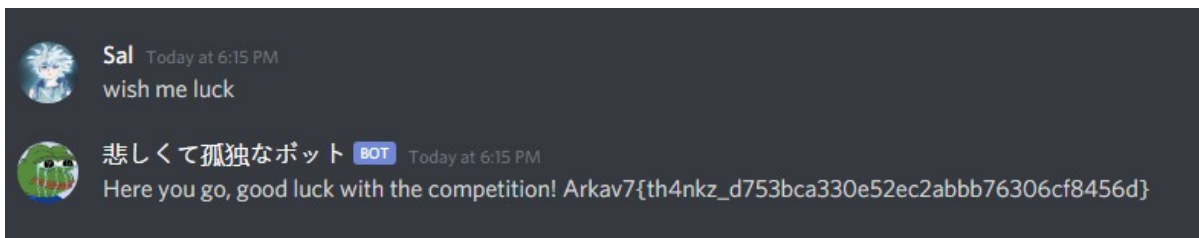
SAD AND LONELY BOT v1.3

Hello, this is sad and lonely bot. A simple discord robot just to waste your time.

Current Change : -Added i need a flag command

Previous Change : -Removed wish me luck command option

Langsung ketik wish me luck, dan didapatkan flag nya



FLAG : Arkav7{th4nkz_d753bca330e52ec2abbb76306cf8456d}

Isekai

Challenge 39 Solves X

isekai
212

Melepas penat CTF dengan main game isekai dulu.

Flag dimasukkan ke dalam format `Arkav7{flag}`

Author: girvinjunod

attachment: https://drive.google.com/file/d/1r7E-p8vXuhdjwWA76CRFu3AfjU_YXQuh/view?usp=sharing

Flag

Submit

Solution:

Pada tantangan ini kita diberikan file zip yang berisi file binary *main* dan file text *manual.txt*.

Untuk menyelesaikan tantangan ini kita hanya perlu melakukan static analysis dengan menggunakan perintah *rabin2* dan juga *grep*. Berikut ini adalah cara penyelesaiannya,

```
~/CTFI/2021/A/Q/M/isekai$ rabin2 -z main | grep 'flag'
590 0x000f70a4 0x004f70a4 36 37 .rodata ascii flagnya adalah seCr3tb0ssFighTeuy102
1183 0x000f9803 0x004f9803 10 11 .rodata ascii flag_value
1189 0x000f985d 0x004f985d 11 12 .rodata ascii prolog_flag
1621 0x000fb696 0x004fb696 24 25 .rodata ascii $current_prolog_flag_alt
1634 0x000fb760 0x004fb760 9 10 .rodata ascii $use_flag
1635 0x000fb76a 0x004fb76a 32 33 .rodata ascii /home/diaz/GP/src/BipsPl/flag.pl
1639 0x000fb7b9 0x004fb7b9 19 20 .rodata ascii current_prolog_flag
1642 0x000fb7e8 0x004fb7e8 15 16 .rodata ascii set_prolog_flag
1673 0x000fb95c 0x004fb95c 8 9 .rodata ascii c_cflags
1675 0x000fb96d 0x004fb96d 9 10 .rodata ascii c_ldflags
1687 0x000fba06 0x004fba06 8 9 .rodata ascii flag_c.c
2406 0x000fe94c 0x004fe94c 15 16 .rodata ascii set_prolog_flag
```

Flag : **Arkav7{seCr3tb0ssFighTeuy102}**

Welcome to Arkavidia 7.0

Challenge 61 Solves ×

Welcome to Arkavidia 7.0

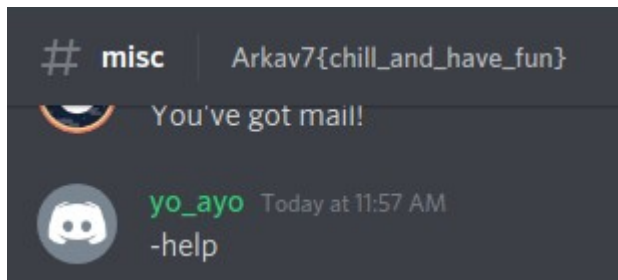
50

Join our discord!

<https://discord.gg/FARmCw8D>

Flag

Submit



Feedback!

Challenge 51 Solves ×

Feedback!

50

Please fill out this form:

<https://forms.gle/TiQFWRUm23NkSjW49>.

Thanks!

Flag

Submit

Flag : **Arkav7{see_you_in_Arkav8}**

Forensics

KawaiiMetal

Challenge 35 Solves X

KawaiiMetal

270

Tadi pas ngubek-ngubek file lama, ada foto Babymetal. Sampe sekarang masih suka banget dengerin lagu ini. Aku rekomen lagu <https://www.youtube.com/watch?v=WIKqgE4BwAY>.

Author: Jspmarc

attachment:
https://drive.google.com/file/d/1u3bdMylcl8Ylq3-bimiOMaV7PL6SC_5Z/view?usp=sharing

Flag Submit

Solution :

Strings pada Moa-metal.png

```
Hint, hint:
Every hidden messages you see are encoded with base64
images )
```

Strings pada Yui-metal.jpg, lakukan decode pada string base64

```
-11111
Hint, hint:
U3RyaW5ncyBhbmQgZ3JlcCB3aXR0IHRob2ZlZSByZWdleCAiPSQiIGl2IGJlZXh0cmVtZWxzZSIHVzZWZ1bA==
images ) echo U3RyaW5ncyBhbmQgZ3JlcCB3aXR0IHRob2ZlZSByZWdleCAiPSQiIGl2IGJlZXh0cmVtZWxzZSIHVzZWZ1bA== | base64 -d
Strings and grep with the regex "=$" is beextremely useful
images )
```

Strings pada Su-metal.jpg, lakukan grep regex =\$, lalu decode string base64

```
=0=
S6VyZSdzIHlvdXIgcMv3YXJkOg==
KFBYb3RpdDogQ292ZSIgeW91ciBLYXJzIHdoZW4gdSBmaW5kIHRob2ZlZSByZWdleCAiPSQiIGl2IGJlZXh0cmVtZWxzZSIHVzZWZ1bA==
aHR0cHM6Ly9wYXNOZWJpb15jb20vcHpsM01mYWc=
Ypa-
images ) echo S6VyZSdzIHlvdXIgcMv3YXJkOg== | base64 -d
Here's your reward:
images ) echo KFBYb3RpdDogQ292ZSIgeW91ciBLYXJzIHdoZW4gdSBmaW5kIHRob2ZlZSByZWdleCAiPSQiIGl2IGJlZXh0cmVtZWxzZSIHVzZWZ1bA== | base64 -d
(Protip: Cover your ears when u find the flag and don't open random sound files willy-nilly)
images ) echo aHR0cHM6Ly9wYXNOZWJpb15jb20vcHpsM01mYWc= | base64 -d
https://pastebin.com/pzR3Mfag
images )
```

Didapat link pastebin yang berisi f l a g, dan password yang akan digunakan pada tahap berikutnya



Lakukan binwalk pada Babymetal.jpg, unzip file dengan menggunakan password road_of_resistance sehingga didapatkan file flag.flac

```
tes ) foremost Babymetal.jpg
Processing: Babymetal.jpg
|foundat=flag.flacUT
*|
tes ) tree
.
├── Babymetal.jpg
├── output
│   ├── audit.txt
│   ├── jpg
│   │   └── 00000000.jpg
│   └── zip
│       └── 00000219.zip
3 directories, 4 files
tes )
```

```
00000219.zip
zip ) unzip 00000219.zip
Archive: 00000219.zip
[00000219.zip] flag.flac password:
inflating: flag.flac
zip ) ls
00000219.zip  flag.flac
zip )
```

Di tahap terakhir ini gw agak bingung. Dan setelah nyari2 ketemu wu ini.

<https://github.com/itsecgary/CTFs/tree/master/rgbCTF%202020/Alien%20Transmission%201>

SSTV tu media pengiriman gambar via sinyal radio, tapi disini diterapkan jadi chall stego. Setelah menggunakan QSSTV, flag akan muncul seperti berikut :



Flag : Arkav7{Gimm3_Ch0c0lat3}