

# Write Up Bronco CTF 2025

## ICC Pisang Epe



Disusun oleh:

- .nadhif - Muh. Nadhiftamma Ayatilla A.P
  - chell07 - Chelsea Elysia Chandean
- giginaga - Andi Ghaniyatera Febriana Harfa Makkasau

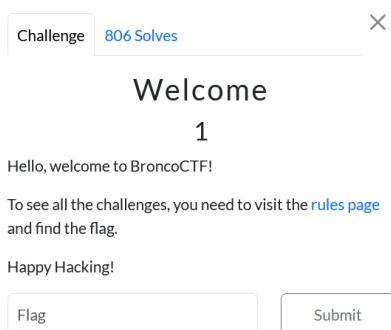
# Daftar Isi

|                                  |           |
|----------------------------------|-----------|
| <b>Welcome.....</b>              | <b>3</b>  |
| Welcome.....                     |           |
| ...                              |           |
| dont be a                        |           |
| slacker.....                     |           |
| <b>Beginner.....</b>             | <b>4</b>  |
| At least its not                 |           |
| pandora.....                     |           |
| Break the                        |           |
| Battalion.....                   |           |
| Inspector Requests.....          |           |
| Simon                            |           |
| Says.....                        |           |
| Too Many                         |           |
| Emojis.....                      |           |
| Straight Up Circular.....        |           |
| <b>Web.....</b>                  |           |
| <b>10</b>                        |           |
| Grandma`s Secret                 |           |
| Recipe.....                      |           |
| <b>Reversing.....</b>            | <b>12</b> |
| Reversing for Ophidiophiles..... |           |
| theflagishere!.....              |           |
| <b>Crypto.....</b>               | <b>16</b> |
| Across the tracks.....           |           |
| Rahh-SA.....                     |           |
| Universal Shorthand.....         |           |
| <b>OSINT.....</b>                | <b>23</b> |
| April 25.....                    |           |

|   |           |
|---|-----------|
| Filling Some Data.....                  | .....     |
| Phone Numbers Everywhere, Anywhere..... | .....     |
| <b>Forensics.....</b>                   | <b>28</b> |
| QR Coded.....                           | .....     |

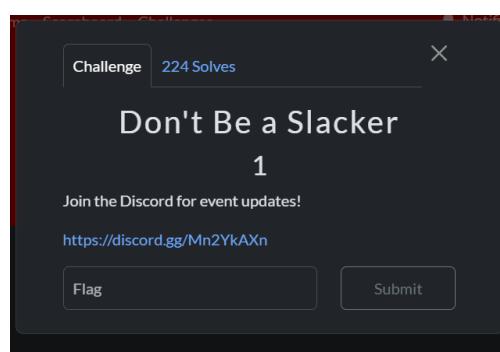
## Welcome

### 1. Welcome

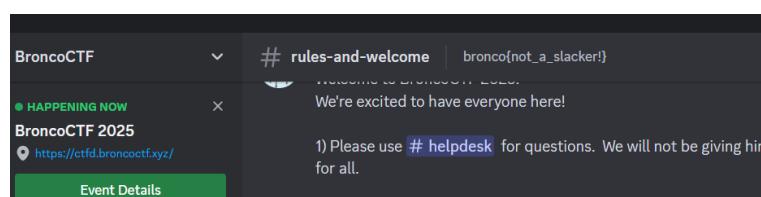


Pada challenge ini pembuat soal memberi tahu bahwa flagnya terdapat di page *rules* yang ada pada website ctfd, oleh karena itu untuk mendapatkan flagnya kita hanya perlu ke page *rules* lalu menemukan flagnya  
**flag: bronco{welcome\_to\_the\_show}**

### 2. Don't Be a Slacker



Pada challenge ini kita hanya perlu menekan link discordnya untuk join, lalu setelah dicari-cari, akhirnya flagnya ketemu di bagian “rules-and-welcome”.



flag:bronco{not\_a\_slacker!}

## Beginner

### 1. At Least It's Not Pandora

Challenge 262 Solves X

#### At Least It's Not Pandora

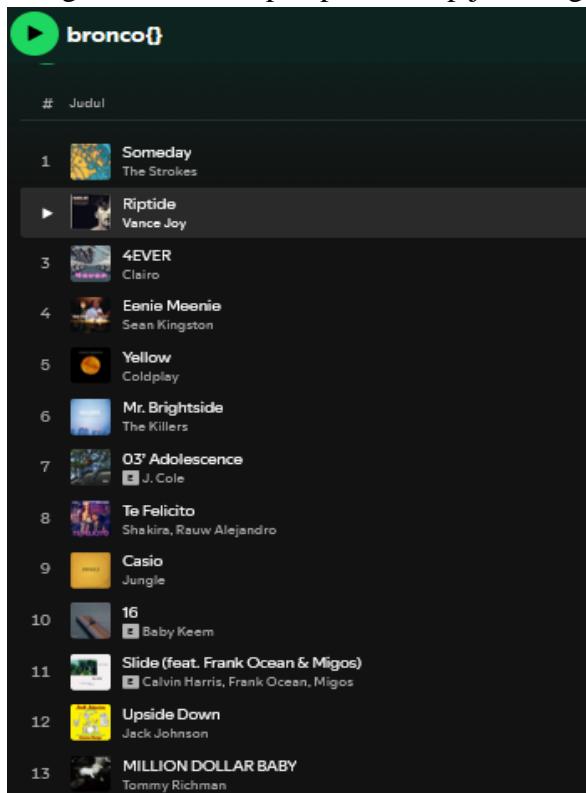
10 .tidalw

I really enjoy listening to music, but I hate that Spotify keeps shuffling my playlists. My taste in music used to be so backwards. P.S. One of the songs is my favorite song.

[https://open.spotify.com/playlist/3UD6tVsCoVqal5BgXug19m?go=1&sp\\_cid=a0f9926371de38e180f302dedf1df658&nd=1&dsi=3f0850ce261f4c27](https://open.spotify.com/playlist/3UD6tVsCoVqal5BgXug19m?go=1&sp_cid=a0f9926371de38e180f302dedf1df658&nd=1&dsi=3f0850ce261f4c27)

Flag Submit

Pada Challange ini pembuat soal memberi sebuah link playlist spotify, pembuat soal berkata “My taste in music used to be so backwards” dengan hint itu kami mengambil huruf depan pada setiap judul lagu dari playlist terbawah hingga ke atas



didapat sebuah kalimat “MUS1CT0MYE4RS”

flag: bronco{MUS1CT0MYE4RS}

## 2. Break the Battalion

Challenge 369 Solves

### Break the Battalion

10

tot\_tater

You have received a file from the the infamous Bronco Battalion of the military. What is the correct input which gives you access to the military secrets? Format is bronco.

[a.out](#)

Flag Submit

Diberikan sebuah file bernama ‘a.out’ yang berisi sebuah file binary yang terenkripsi, pembuat soal berkata bahwa flag adalah input dari file output yang diberikan. untuk menyelesaikan masalah ini kita memerlukan tools.

Tools yang diperlukan:

- Terminal Linux
- Decompiler

Langkah untuk mendapatkan flag:

1. Dekripsi file ‘a.out’ menggunakan tools Decompiler dimana disini kami menggunakan website dogbolts.org untuk mendapatkan source code

The screenshot shows the Dogbolts.org platform's Decompiler Explorer interface. A file named 'a.out' has been uploaded. The analysis results are displayed in four panes:

- Ghidra C:** Shows assembly code for the program, including function definitions like `angr` and `main`.
- BinaryNinja C:** Shows the same assembly code in a different format.
- Ghidra C:** Shows the assembly code again, with some annotations and highlights.
- Hex-Rays C:** Shows the assembly code with syntax highlighting and annotations.

2. setelah mendapatkan source code selanjutnya kita perlu untuk mencari bagian fungsi “main” dari kode tersebut untuk mendapatkan informasi, didapatkanlah informasi bahwa kode ini akan menjalankan program yang bertanya apa password untuk masuk, dan passwordnya adalah “brigade” yang di enkripsi menggunakan XOR cipher dengan key 0x50

```
local_10 = *(long *)(in_FS_OFFSET + 0x28);
friendlyFunction();
puts("What is ze passcode monsieur?");
iVar1 = (int)local_118;
_isoc99_scanf("%255s");
encrypt(local_118,iVar1);
iVar1 = strcmp(local_118,"brigade");
if (iVar1 == 0) {
    puts("correct password");
}
else {
    puts("wrong password");
}
```

3. setelah mendapatkan password kita harus mendekripsinya menggunakan kode python berikut ini adalah kodennya

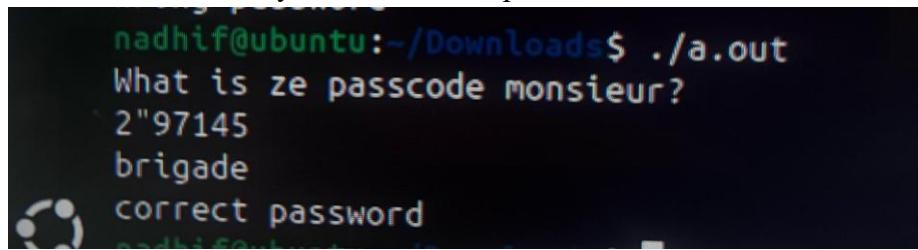
```
encrypted_password = "brigade"
original_password = ''.join([chr(ord(c) ^ 0x50) for c in
encrypted_password])

print(original_password)
```

dengan menjalankan kode itu kami mendapatkan sebuah output yaitu

```
[Running] python -u "d:\CTF\broncoCTF\AOUTTTT.py"
2"97145
```

4. dengan demikian didapatlah sebuah password input, untuk memastikannya kami mencoba menjalankan file ‘a.out’ di terminal linux dan menginput ‘2”97145’ dan hasilnya adalah correct password



nadhib@ubuntu:~/Downloads\$ ./a.out  
What is ze passcode monsieur?  
2"97145  
brigade  
correct password

flag: bronco{2"97141}

### 3. Inspector Requests

Challenge 446 Solves X

**Inspector Requestor**

10

whalker

Want a flag? You need to request it via our Form, for efficiency.

<https://forms.gle/oipmJZeVzYMrgKv39>

Pada challenge ini kami diberi sebuah link gform, dan sesuai judulnya kami hanya perlu untuk melakukan inspect elemen dengan melakukan shortcut Ctrl+U



```
<!DOCTYPE html><html class="HB1eCd-UMrrmb PH0cVb"><head><link rel="shortcut icon" sizes="16x16" href="https://ssl.gstatic.com/docs/spreadsheets/forms/fe">
<meta property="og:ttl" content="604800"></head><body dir="ltr" itemtype="http://schema.org/Form">
<div>Since you are here, here is the flag! Inspect element is fun.</div>
<div>bronco{why_does_google_still_expos3_th1s_wh3n_i_stopped_accepting_submissions_101}</div>
<div>Since you are here, here is the flag! Inspect element is fun.</div>
<div>bronco{why_does_google_still_expos3_th1s_wh3n_i_stopped_accepting_submissions_101}</div>
```

flag:

bronco{why\_does\_google\_still\_expos3\_th1s\_wh3n\_i\_stopped\_accepting\_submissions\_101}

### 4. Simon Says

Challenge 369 Solves X

### Simon Says

10

tiffany\_ttn

Help me play this game of simon says - remember, the last 2 lights have been blue!

[Download simon.png](#)

Flag  Submit

Diberikan sebuah foto ‘simon.png’ dan kita diminta untuk mencari flag di dalam gambar tersebut. Untuk mendapatkan flagnya kita hanya memerlukan sebuah tools forensic

Tools yang diperlukan:

- apperisolve.com

Untuk mendapatkan flag kita hanya perlu untuk memasukkan image pada web apperisolve.com lalu scroll kebawah untuk mencari informasi terkait gambar tersebut, di temukanlah sebuah bagian yang disembunyikan pada gambar



flag: bronco{simon\_says\_submit\_this\_flag}

## 5. Too Many Emojis

Challenge 324 Solves X

### Too Many Emojis

10

tiffany\_ttn

I like using emojis in my text messages, but my friend may have taken it too far. 🤪 Can you figure out what she's trying to tell me? 🤫

[Download emojis.png](#)

Diberikan sebuah gambar yang merupakan serangkaian emoji yang disusun seperti sebuah bentuk flag



Untuk mendapatkan flagnya kita hanya perlu untuk mengambil setiap huruf pertama dari emoji tersebut namun dalam bahasa Inggris seperti berikut:

💔 : b (Broken Heart)

😌 : r (Relieved Face)

👹 : o (Ogre)

🤢 : n (Nauseated Face)

😖 : c (Confounded Face)

😡 : e (Enraged Face)

😠 : a (Angry Face)

❤️ : m (Mending Heart)

□: j (Jellyfish)

🏒 : i (Ice Hockey)

🎿 : s (Sled)

▢: x (X-Ray)

🐩 : p (Poodle)

💛 : y (Yellow Heart)

🐫 : t (Two-Hump Camel)

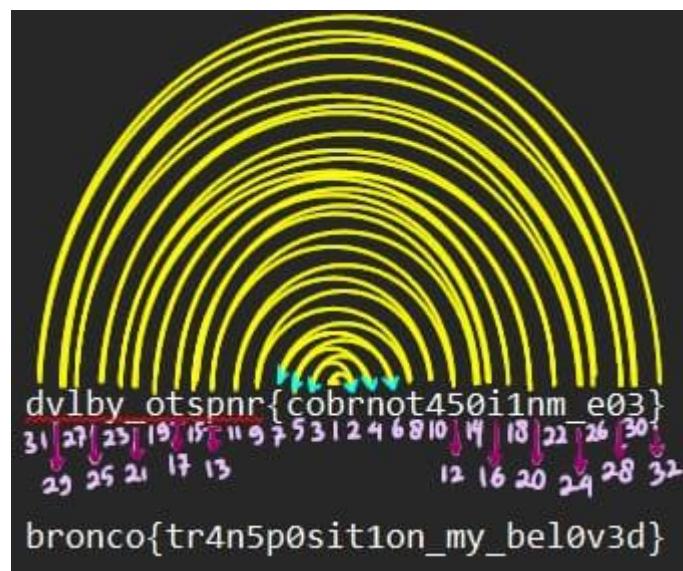
dengan mengganti emojis dengan huruf-huruf tersebut didapatkanlah sebuah flag asli

`flag: bronco{emojis_express_my_emotions}`

## 6. Straight Up Circular



Pada challenge ini diberikan *encrypted code* yang perlu di-*decode* kembali, setelah diperhatikan secara seksama ternyata soal ini cukup simple, karena kita sudah tau bahwa format flagnya adalah bronco{ }, kita hanya perlu membaca dan menyusun hurufnya mulai dari huruf b lalu ke kanan satu langkah, lalu ke kiri dari b, sehingga flagnya pun didapatkan, lebih jelasnya seperti gambar berikut.



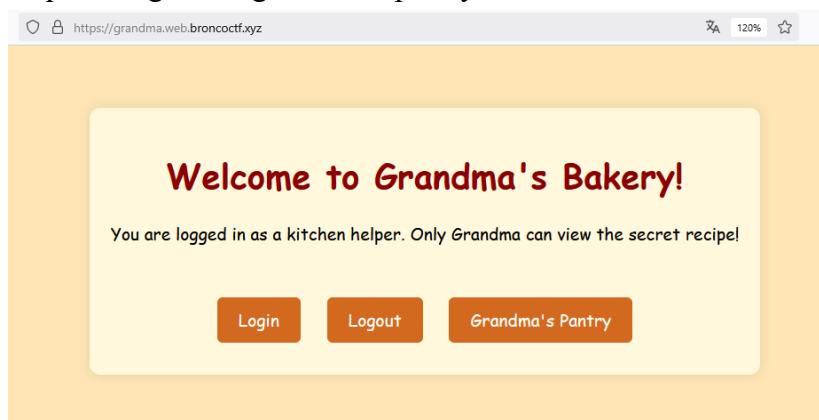
flag: bronco{tr4n5p0sit1on\_my\_be10v3d}

# Web

## 1. Grandma's Secret Recipe

The screenshot shows a challenge interface for a web-based challenge. At the top, it says "Challenge" and "422 Solves". The title of the challenge is "Grandma's Secret Recipe" with a difficulty rating of "10" and created by "whalker". The challenge description reads:  
Grandma has been baking her world-famous cookies for decades, but she's always kept her secret recipe locked away. Nobody—not even her most trusted kitchen helpers—knows the full list of ingredients.  
She insists it's all about "the perfect balance of love and a pinch of mystery", but deep down, you know there's more to it. Rumors say only Grandma herself is allowed to see the recipe, hidden somewhere in her kitchen.  
But, you were hired by Grandpa, who divorced her because she refused to share the recipe. Can you figure out the true secret behind her legendary cookies? 🍪🍪  
A link to the website is provided: <https://grandma.web.broncoctf.xyz>. There are two buttons at the bottom: "Flag" and "Submit".

Diberikan sebuah url website dan kita diminta untuk mencari flag didalamnya, Website tersebut memiliki button ‘login’, ‘logout’, dan ‘grandma’s pantry’. namun kita tidak dapat mengakses ‘grandma’s pantry’.



**Langkah untuk mendapatkan flag:**

### 1. Cek Cookies

Ketika kita mengecek kuki dengan inspect memories, didapatkan informasi bahwa terdapat checksum: a223befb6660a23f9c3491f74ef84e43, dan role: "kitchen helper".

The screenshot shows the browser developer tools with the "Cookies" tab selected. A table lists the stored cookies:

|                                   | Nama     | Nilai                            | Domain                    | Path | Kedaluwarsa / Usia Maksimal | Ukuran | HttpOnly | Secure | SameSite |
|-----------------------------------|----------|----------------------------------|---------------------------|------|-----------------------------|--------|----------|--------|----------|
| https://grandma.web.broncoctf.xyz | checksum | a223befb6660a23f9c3491f74ef84e43 | grandma.web.broncoctf.xyz | /    | Sesi                        | 40     | true     | true   | None     |
| https://grandma.web.broncoctf.xyz | role     | "kitchen helper"                 | grandma.web.broncoctf.xyz | /    | Sesi                        | 20     | true     | true   | None     |

## 2. Dekripsi Checksum

Checksum a223befb6660a23f9c3491f74ef84e43 adalah sebuah string yang dienkripsi menggunakan hash MD5 yang artinya “kitchen helper”, oleh karena itu kita harus menggantinya menjadi “grandma” menggunakan hash MD5, dengan bantuan AI kami mendapatkan checksum “grandma” yaitu:

a5d19cdd5dfd1a8f664cDee2b5e293167

## 3. Mengganti Cookies

terakhir kita hanya perlu mengganti checksum menjadi

a5d19cdd5dfd1a8f664cDee2b5e293167 dan Role menjadi “grandma”

The screenshot shows a browser developer tools interface with the Network tab selected. At the top, there's a yellow banner with the text "Welcome to Grandma's Bakery!" and "Grandma's Secret Recipe:". Below the banner, a blue bar contains links for "Login", "Logout", and "Grandma's Pantry". The main content area shows a table of cookies. One cookie is highlighted with a red border: "role" with value "grandma". The table has columns for Name, Value, Domain, Path, Expiry, Size, HttpOnly, Secure, SameSite, and Last Accessed.

| Name     | Value                             | Domain                    | Path | Expiry | Size | HttpOnly | Secure | SameSite | Last Accessed    |
|----------|-----------------------------------|---------------------------|------|--------|------|----------|--------|----------|------------------|
| checksum | a5d19cdd5dfd1a8f664cDee2b5e293167 | grandma.web.broncoctf.xyz | /    | Sesi   | 40   | true     | true   | None     | Sun, 16 Feb 2024 |
| role     | "grandma"                         | grandma.web.broncoctf.xyz | /    | Sesi   | 13   | true     | true   | None     | Sun, 16 Feb 2024 |

flag: bronco{grandma-makes-b3tter-cookies-than-girl-scouts-and-i-w1ll-fight-you-over-th@t-fact}

# Reversing

## 1. Reversing for Ophidiophiles

The screenshot shows a challenge card for a reversing challenge. At the top, there's a 'Challenge' button and a '448 Solves' badge. The title 'Reversing for Ophidiophiles' is centered above a difficulty rating of '10'. Below the title, the author is listed as 'shwhale'. A descriptive text reads: 'Do you love python? Or at least tolerate it? Then this is the challenge for you!'. It also states: 'When run with the correct flag, the given file prints: 23a326c27bee9b40885df97007aa4dbe410e93.' Below this, a question 'What is the flag?' is followed by a download button labeled 'chall.py'. At the bottom are two buttons: 'Flag' and 'Submit'.

Di tantangan ini, kita diberikan sebuah skrip Python yang melakukan proses enkripsi pada sebuah flag yang dimasukkan oleh pengguna. Tujuan dari tantangan ini adalah untuk mencari tahu flag yang benar, dengan cara membalikkan proses enkripsi yang ada.

Tujuan dari tantangan ini adalah untuk membalikkan proses ini dan menemukan flag yang benar. Kita diberikan output dalam bentuk heksadesimal: **23a326c27bee9b40885df97007aa4dbe410e93**

Berikut adalah kode yang diberi oleh soal:

```
flag = input()
carry = 0
key = "Awesome!"
output = []
for i,c in enumerate(flag):
    val = ord(c)
    val += carry
    val %= 256
    val ^= ord(key[i % len(key)])
    output.append(val)
    carry += ord(c)
    carry %= 256

print(bytes(output).hex())
```

Kita harus membalikkan proses enkripsi ini untuk menemukan flag yang benar. Proses yang kita lakukan di sini adalah dekripsi.

### Langkah untuk mendapatkan flag:

#### 1. Membalikkan Operasi XOR

XOR adalah operasi simetris, artinya kita bisa membalikkan efeknya dengan cara XOR lagi dengan kunci yang sama.

#### 2. Membalikkan Modulo dan Carry

Kita juga perlu mengembalikan efek dari operasi carry dan operasi modulo 256. Ini akan melibatkan perhitungan ulang dengan cara yang sama seperti yang dilakukan dalam enkripsi.

#### 3. Menjalankan kode dan memperoleh flag

Kode:

```
# Kunci yang digunakan untuk XOR enkripsi
key = "Awesome!"

# Output yang diberikan (dalam format heksadesimal) yang harus kita dekripsi
encrypted_output = bytes.fromhex("23a326c27bee9b40885df97007aa4dbe410e93")

# Fungsi untuk membalikkan proses enkripsi
def reverse_encryption(encrypted_output):
    flag = []
    carry = 0
    for i, enc_val in enumerate(encrypted_output):
        # Membalikkan XOR dengan kunci
        key_char = ord(key[i % len(key)])
        decrypted_val = enc_val ^ key_char

        # Membalikkan efek carry dengan menguranginya, dan melakukan modulo 256
        decrypted_val -= carry
        decrypted_val %= 256

        # Menambahkan nilai yang telah didekripsi ke dalam list flag (dalam bentuk karakter)
        flag.append(chr(decrypted_val))

        # Memperbarui carry untuk iterasi berikutnya
        carry += decrypted_val
        carry %= 256

    return ''.join(flag)

# Memanggil fungsi untuk dekripsi
```

```
flag = reverse_encryption(encrypted_output)
print(f"Flag yang benar adalah: {flag}")
```

flag: bronco{charge\_away}

## 2. theflagishere!

Challenge 273 Solves ×

### theflagishere!

10  
serilical

So, my friend sent me this program that's supposed to determine the flag for this challenge, right? But, somehow, they forgot to actually say what the flag is. Classic move. 😅 Now, it's on you to figure out what the true flag is. If you can crack it and figure out what my friend was trying to send, that flag is all yours! Ready to flex those decoding skills? Let's get it!

Format: bronco{flag}

Download theflagishe...

Flag Submit

Soal ini memberikan file Python yang telah dikompilasi menjadi file .pyc. File .pyc berisi bytecode Python yang tidak bisa langsung dibaca seperti kode Python biasa. Tujuan dari soal ini adalah untuk menemukan **flag** yang disembunyikan dalam bytecode tersebut. Flag biasanya berupa sebuah string yang digunakan untuk menyelesaikan tantangan CTF.

Namun, soal ini memberikan informasi tentang bagaimana flag disusun dengan menggunakan beberapa fungsi yang bekerja dengan cara tertentu untuk menghasilkan karakter-karakter dalam flag.

### Langkah-Langkah Penyelesaian

#### 1. Menganalisis Fungsi-Fungsi

Dalam soal ini, terdapat beberapa fungsi seperti char\_0(), char\_1\_4\_6(), dan lainnya yang masing-masing mengembalikan karakter-karakter berdasarkan frekuensi kemunculan karakter dalam suatu string. Setelah itu, karakter yang paling sering muncul akan dimodifikasi dengan menambahkan 1 pada nilai ASCII-nya dan kemudian fungsi tersebut mengembalikan karakter yang baru.

Contoh:

- Fungsi `char_0()` menghasilkan karakter 'j'. Ini karena karakter yang paling sering muncul dalam string yang digunakan oleh fungsi ini adalah 'i' (nilai ASCII 105). Setelah menambahkan 1 pada nilai ASCII tersebut ( $105 + 1$ ), kita mendapatkan karakter 'j' (nilai ASCII 106).
- Fungsi `char_1_4_6()` menghasilkan karakter '\\ karena karakter yang paling sering dalam string adalah '\_' (nilai ASCII 95), dan setelah menambahkan 1, kita mendapatkan karakter '"' (nilai ASCII 96).

## 2. Rekonstruksi Flag

Setelah memahami bagaimana setiap fungsi bekerja, kita bisa menganalisis hasil dari masing-masing fungsi dan kemudian membalikkan prosesnya untuk mendapatkan karakter asli yang membentuk flag.

Sebagai contoh:

- Fungsi `char_0()` menghasilkan 'j', jadi kita menguranginya dengan 1 untuk mendapatkan 'i'.
- Fungsi `char_1_4_6()` menghasilkan '\\, jadi kita menguranginya dengan 1 untuk mendapatkan '\_'.
- Fungsi `char_2()` menghasilkan 'b', jadi kita menguranginya dengan 1 untuk mendapatkan 'a'.
- Dan seterusnya.

## 3. Hasil Rekonstruksi

Dengan mengikuti langkah di atas, kita bisa memperoleh karakter-karakter yang membentuk flag sebagai berikut:

- `char_0()` → 'i'
- `char_1()` → '\_'
- `char_2()` → 'a'
- `char_3()` → 'm' (ini adalah karakter yang sudah ditentukan sebelumnya atau hardcoded)
- `char_4()` → '\_'
- `char_5()` → 'a'
- `char_6()` → '\_'
- `char_7()` → 'f'
- `char_8()` → 'l'
- `char_9()` → 'a'
- `char_10()` → 'g'

Dengan cara tersebut diperoleh lah sebuah susunan kalimat “`i_am_a_flag`”

flag: bronco{i\_am\_a\_flag}

# Crypto

## 1. Across The Tracks

Challenge 365 Solves X

### Across the Tracks

10

whalker

I've been working on the railroad, all my live long day. We really should put up a fence, a deer just ran onto the tracks in a zig-zag pattern. After crossing my *tenth* track tracing the deer, I have found this message! What could it mean?

```
Samddre..ath-dhf@_oesoere.ebun.yhot.no..oso.i.a.lrlr  
cm.is.aruf-toibadhn.nadpkudynea{l_oeee.ch.oide.f.n.  
aoe.sae.aonbdhgo_so.rr.i.tYnl.s.tdot.xs.hdtty'.t.cf  
rlca.epeo.iufiyi.t.yaaf.a.ts..tn33}  
i.tvhr.tooho...rlmwuI.h.e.iHshonppsoleaseecrtudIdet  
.n.BtIpdhieori...or.ovl.c..i.acn.t.su..ootr.:b3ces  
slyedheIath.e._
```

Flag Submit

Pada challenge ini kita diberikan *encoded text* dengan clue *tracks zig-zag pattern*. Jadi kita dapat menggunakan *rail fence cipher* sesuai dengan clue tersebut dan flagnya pun kami dapatkan.(link:<https://www.dcode.fr/rail-fence-cipher>)

The Rail Fence Decoder tool interface includes:

- RAIL FENCE DECODER** section with a **ZIGZAG CIPHERTEXT** input field containing the encoded text.
- PARAMETERS AND OPTIONS** section with checkboxes for **KEEP PUNCTUATION AND SPACES** (checked), **CHARACTER FOR SPACES** (set to Underscore (low dash)), and **AUTOMATIC DECRYPTION**.
- RESULTS** section showing the decrypted message: "Some ciphers are easier to solve. Some ciphers are harder to solve. You definitely could brute force this one if you did it by hand. I had to do that recently on an exam. It was not as fun as I had hoped. But that is okay. I hope you didn't do this by hand. Here is the flag tho: bronco{r@11\_f3nc3\_cip3rs\_r\_cool}".

flag: bronco{r@11\_f3nc3\_cip3rs\_r\_cool}

## 2. Rahhh-SA

Challenge **283 Solves** ×

### Rahhh-SA

**10**

yoshie878

Behold! A modern take on an old crypto classic!

I call it RAHHH-SA! That's because with just a simple numerical inversion of RSA's rules, it's now unbreakable!

RAHHH!

e = 65537  
n = 3429719  
c = [-53102, -3390264, -2864697, -3111409, -2002688,

You know what? I'm so confident in this new system I'll even share one of my deepest secrets!

p = -811

Pada Challange ini kami diberi sebuah data dan kami diminta untuk mendekripsi data tersebut untuk mendapatkan flagnya, berikut adalah informasi data yang diberikan:

Public exponent (e): 65537

Modulus (n): 3429719

Ciphertext @: [-53102, -3390264, -2864697, -3111409, -2002688, -2864697, -1695722, -1957072, -1821648, -1268305, -3362005, -712024, -1957072, -1821648, -1268305, -732380, -2002688, -967579, -271768, -3390264, -712024, -1821648, -3069724, -732380, -892709, -271768, -732380, -2062187, -271768, -292609, -1599740, -732380, -1268305, -712024, -271768, -1957072, -1821648, -3418677, -732380, -2002688, -1821648, -3069724, -271768, -3390264, -1847282, -2267004, -3362005, -1764589, -293906, -1607693]

Petunjuk: p = -811

**Langkah untuk mendapatkan flag:**

#### 1. Memahami Informasi yang Diberikan

Dalam RSA:

1. n adalah hasil perkalian dua bilangan prima, yaitu p dan q:  
 $n = p \times q$
2. e adalah eksponen publik yang digunakan untuk enkripsi.

3. c adalah ciphertext, yaitu pesan yang telah dienkripsi menggunakan rumus:  
 $c = m^e \mod n$
4. Untuk dekripsi, kita perlu menemukan d (kunci privat) menggunakan rumus:  
 $d \times e \equiv 1 \mod \phi(n)$   
di mana  $\phi(n) = (p-1)(q-1)$

## 2. Menemukan Nilai p dan q

Kita diberikan petunjuk bahwa  $p = -811$ . Namun, dalam RSA, p dan q harus bilangan prima positif. Jadi, kita ambil nilai absolutnya:

$$p = 811$$

Kemudian, kita cari q dengan membagi n oleh p:

$$q = n/p = 3429719/811 = 4229$$

Sekarang kita tahu:

- $p = 811$
- $q = 4229$

## 3. Menghitung $\phi(n)\phi(n)$

$\phi(n)$  adalah fungsi Euler Totient, yang dihitung sebagai:

$$\phi(n) = (p-1)(q-1)$$

Substitusi nilai p dan q:

$$\begin{aligned} \phi(n) &= (811-1)(4229-1) = 810 \times 4228 = 3424680 \\ \phi(n) &= (811-1)(4229-1) = 810 \times 4228 = 3424680 \end{aligned}$$

## 4. Menghitung Kunci Privat (d)

Kunci privat d adalah invers modular dari e modulo  $\phi(n)\phi(n)$ . Artinya, kita perlu mencari d yang memenuhi:

$$d \times e \equiv 1 \mod \phi(n)$$

Dengan  $e = 65537$  dan  $\phi(n) = 3424680$ , kita bisa menggunakan Algoritma Euclidean Extended untuk mencari d. Hasilnya adalah:

d=273193

## 5. Dekripsi Ciphertext

Ciphertext ( $c$ ) diberikan sebagai list bilangan integer. Untuk mendekripsi setiap nilai  $c$ , kita gunakan rumus:

$$m = c^d \mod n$$

Namun, beberapa nilai  $c$  adalah bilangan negatif. Karena modulus ( $n$ ) tidak bekerja dengan bilangan negatif, kita perlu menyesuaikannya dengan menambahkan  $n$  ke nilai  $c$  yang negatif. Contoh:

Jika  $c = -53102$ , maka:

$$c = -53102 + 3429719 = 3376617$$

Setelah menyesuaikan nilai  $c$ , kita bisa menghitung  $m$  menggunakan rumus di atas.

## 6. Implementasi Dekripsi

Berikut adalah contoh implementasi dalam Python untuk mendekripsi ciphertext:

```
# Parameter RSA

e = 65537

n = 3429719

c = [-53102, -3390264, -2864697, -3111409, -2002688, -2864697, -
1695722, -1957072, -1821648, -1268305, -3362005, -712024, -1957072, -1821648,
-1268305, -732380, -2002688, -967579, -271768, -3390264, -712024, -1821648, -
3069724, -732380, -892709, -271768, -732380, -2062187, -271768, -292609, -
1599740, -732380, -1268305, -712024, -271768, -1957072, -1821648, -3418677, -
732380, -2002688, -1821648, -3069724, -271768, -3390264, -1847282, -2267004, -
3362005, -1764589, -293906, -1607693]

p = 811

q = 4229

# Hitung phi(n)
```

```

phi = (p - 1) * (q - 1)

# Hitung d (kunci privat)

d = pow(e, -1, phi)

# Dekripsi setiap ciphertext

plaintext = []

for ci in c:

    if ci < 0:

        ci += n # Ubah nilai negatif menjadi positif

    mi = pow(ci, d, n) # Dekripsi

    plaintext.append(mi)

# Cetak hasil dekripsi

print("Plaintext (ASCII):", plaintext)

```

Dari Kode diatas diperoleh ASCII yaitu sebagai berikut:

```

Plaintext (ASCII): [98, 114, 111, 110, 99, 111, 123, 109, 52, 116, 104,
51, 109, 52, 116, 49, 99, 53, 95, 114, 51, 52, 108, 49, 121, 95, 49, 115, 95,
113, 117, 49, 116, 51, 95, 109, 52, 103, 49, 99, 52, 108, 95, 114, 97, 65,
104, 72, 33, 125]

```

## 7. Konversi ke flag

Untuk mengubah list integer tersebut menjadi teks, kita perlu mengonversi setiap bilangan integer ke karakter ASCII yang sesuai. Berikut adalah prosesnya:

- Setiap integer diubah ke karakter menggunakan fungsi **chr()**:
  - **98** → 'b'
  - **114** → 'r'

- **111** → 'o'
- **110** → 'n'
- **99** → 'c'
- **111** → 'o'
- **123** → '{'
- **109** → 'm'
- **52** → '4'
- **116** → 't'
- **104** → 'h'
- **51** → '3'
- **109** → 'm'
- **52** → '4'
- **116** → 't'
- **49** → '1'
- **99** → 'c'
- **53** → '5'
- **95** → '\_'
- **114** → 'r'
- **51** → '3'
- **52** → '4'
- **108** → '1'
- **49** → '1'
- **121** → 'y'
- **95** → '\_'
- **49** → '1'
- **115** → 's'
- **95** → '\_'
- **113** → 'q'
- **117** → 'u'
- **49** → '1'
- **116** → 't'
- **51** → '3'
- **95** → '\_'
- **109** → 'm'
- **52** → '4'
- **103** → 'g'

- **49** → '1'
- **99** → 'c'
- **52** → '4'
- **108** → 'l'
- **95** → '\_'
- **114** → 'r'
- **97** → 'a'
- **65** → 'A'
- **104** → 'h'
- **72** → 'H'
- **33** → '!''
- **125** → '}''

flag: bronco{m4th3m4t1c5\_r34l1y\_1s\_qu1t3\_m4g1c41\_raAhH! }

### 3. Universal Shorthand



Pada challenge ini kita diberikan sebuah *encoded text* yang disebut bahwa bisa digunakan untuk semua bahasa, setelah menelusuri puisi aslinya dan, menganalisis secara manual, pada saat pengerjaan kami mendapatkan:

```
poem.txt      intel.txt
File Edit View
And here is the flag:
VMBI_bhd PO ^U RMB PO kpOgIU VdXFR ^F ^_ P
be --- oh -- sea oh ----- I el oh tea
ar en open {
bronco-elot3p-n3-ics-
bronco(elot3p@n3tcs)
bronco(elot3p@n3tcs)

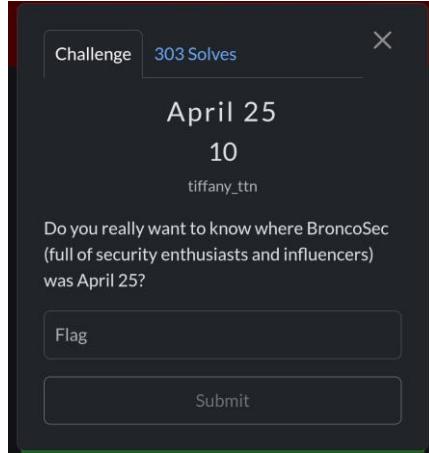
VMBI-be
L[U-one
maybe Kd-th?
U-a
^F-I
PO-oh
NUV-and
[R-us
IP-the
maybe Mh-ea, BH-ou
P-a
U-n
J-s
i-a/t?(1BH=to)
```

Setelah, mencoba input ternyata incorrect. Lalu, setelah mencoba menganalisis kembali, akhirnya kami mendapatkan flagnya.

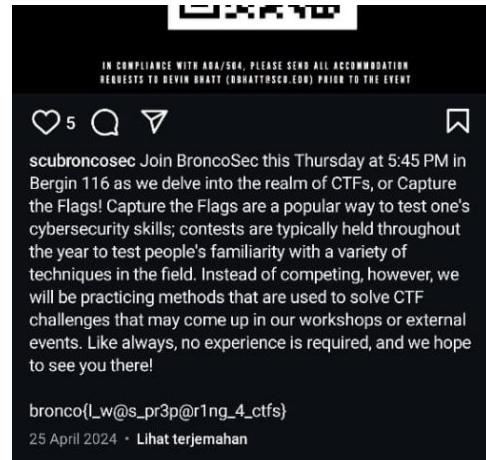
flag: bronco{ilov3ph0n3t1cs}  
\*outsolve

# OSINT

1. April 25

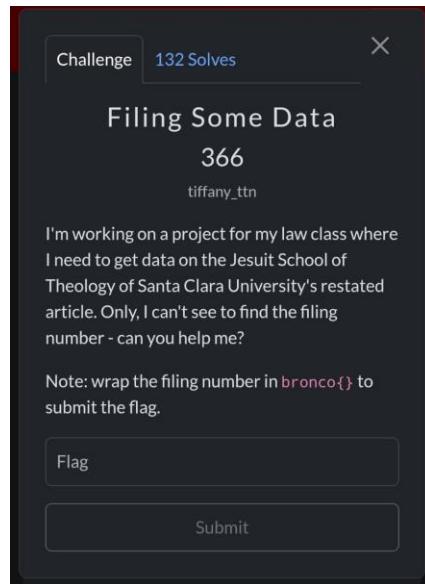


Pada challenge ini terdapat clue yaitu mengenai dimana BroncoSec pada tanggal 25 April, lalu setelah *browsing* di google, kami, menemukan akun instagram dari Broncosec, setelah *scrolling* mencari kegiatan pada tanggal 25 april, akhirnya postingan dan flagnya pun ketemu.



flag: bronco{l\_w@s\_pr3p@r1ng\_4\_ctfs}

## 2. Filling Some Data



Pada challenge ini kita mengetahui bahwa yang dicari adalah *filling number* yang terdapat dalam *restated article* dari *Jesuit School of Theology os Santa Clara University*. Jadi, kami coba menggunakan, chatgpt.

A screenshot of a ChatGPT interface. The user asks: "help me find Jesuit School of Theology of Santa Clara University's restated article". The AI response is: "The Jesuit School of Theology of Santa Clara University (JST-SCU) has its Amended and Restated Articles of Incorporation detailed within its bylaws. These bylaws outline the objectives of the institution, which include preparing individuals for ordination to the Roman Catholic priesthood, training laypersons for various forms of church ministry, and advancing theological scholarship. Additionally, the bylaws specify the management structure, membership, and other operational aspects of the corporation. For a comprehensive understanding, you can access the full text of the bylaws here: SCUDU". There is a "Sumber" button and a "Kirim pesan ke ChatGPT" button.

Awalnya, kami masih belum mendapatkan file yang kami cari, jadi kami mencoba mencari artikel yang memiliki *filling number*.

Lalu, kami pun mendapatkan article yang dibuthkan pada Articles of Incorporation document, setelah mencoba *filling number*, pada slide 5, kami pun mendapatkan flagnya.



A screenshot of a ChatGPT interface. The user asks: "can u help me find the article that have filling number". The AI response is: "The Jesuit School of Theology of Santa Clara University (JST-SCU) was originally incorporated as the Jesuit School of Theology at Berkeley. The Articles of Incorporation were filed with the Secretary of State of California on October 23, 1934. While the specific filing number isn't explicitly mentioned in the available documents, the original trustees listed in the Articles of Incorporation were: • James E. Maione • William E. Donnelly • Charles A. Budda • James F. Donovan • Francis L. Sheerin For detailed information, you can refer to the Articles of Incorporation document. SCUDU Additionally, JST-SCU operates under bylaws that outline its objectives, management structure, and operational aspects. These bylaws are available for review. SCUDU If you require the specific filing number, it would be best to contact the California Secretary of State's office. SCUDU".

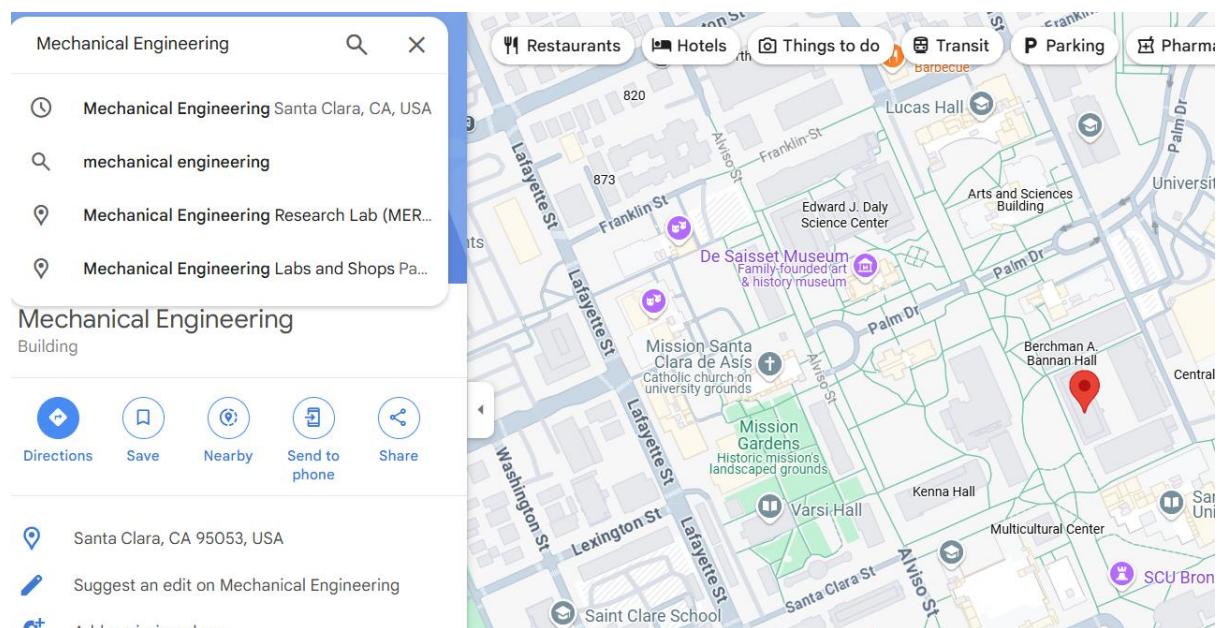
flag: bronco{A0693181}

### 3. Phone Numbers Everywhere Anywhere

Pada challenge ini, diberikan sebuah file bernama aws.jpg

|                      |   |
|----------------------|---|
| y_cb_cr_sub_sampling | YCbCr4:2:0 (2 2)                          |
| image_size           | 900x900                                   |
| megapixels           | 0.81                                      |
| gps_latitude         | 37 deg 20' 56.93" N                       |
| gps_longitude        | 121 deg 56' 18.92" W                      |
| gps_position         | 37 deg 20' 56.93" N, 121 deg 56' 18.92" W |
| category             | image                                     |

Saat metadata dari file ini dicek, ditemukan info tentang lokasi gambar berada.



Dengan informasi longitude dan latitude yang dimasukkan ke dalam **Google Maps**, ditemukan bahwa gambar berada di Santa Clara Mechanical Engineering.

Sobrato Campus for Discovery and Innovation  
Academic department • [Edit](#)

Overview About

Directions Save Nearby Send to phone Share

Santa Clara, CA 95053, United States  
Located in: Santa Clara University  
[scu.edu](http://scu.edu)  
+1 408-554-4600  
83X6+MH Santa Clara, California, USA

Bannan Hall  
School of Law  
Santa Clara University School of Engineering  
Heafey Hall  
In Celebration of Family  
Layers

Dengan melakukan zoom in ditemukan tempat yang lebih spesifik yaitu Sobrato yang memiliki informasi tentang nomor telepon. Flag didapat dengan memasukkan semua nomor tanpa simbol sesuai *clue* yang diberikan.

flag: bronco{14085544600}

## Forensics

### 4. QR Coded

Challenge 170 Solves X

QR Coded  
277  
shwhale

This one should be really easy. All you have to do is scan a QR code!

[easy\\_scan...](#)

Flag Submit

Diberi sebuah image png berupa kode QR, ketika di scan QR ini merujuk pada sebuah flag palsu, kita diminta untuk mencari flag aslinya.

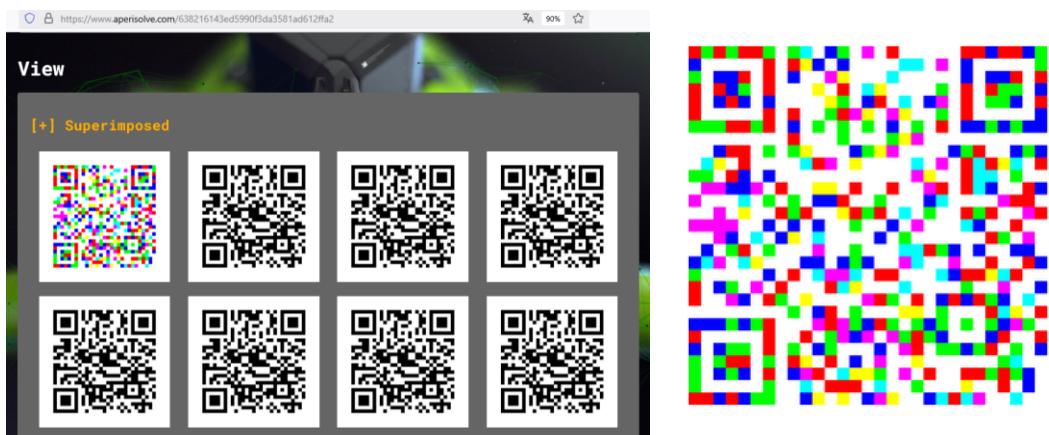
### Tools yang diperlukan:

- Aperi`solve.com
- 8bit painter

### Langkah untuk mendapatkan Flag:

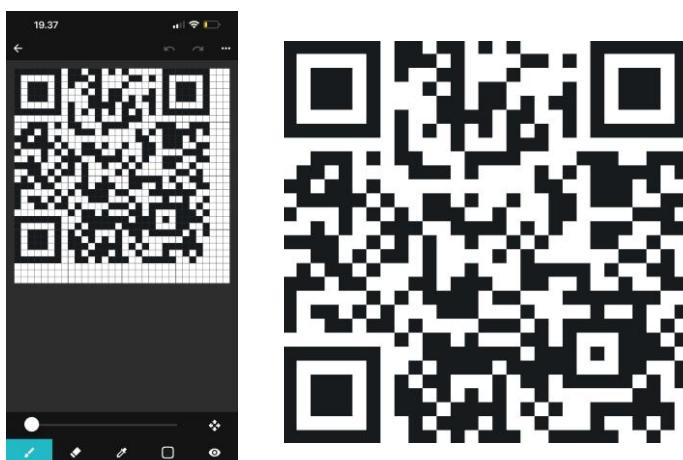
#### 1. Menggunakan Apperisolve

Dengan menginput gambar pada Apperisolve, kami dapat menemukan informasi baru, salah satunya kita dapat melihat view dibagian superimposed dan kita mendapatkan QR code lain namun dengan warna yang teracak sehingga kami tidak bisa memindai nya.



#### 2. Memperbaiki QR yang ditemukan

Untuk dapat memindainya kita harus memperbaiki kode QR nya menjadi warna hitam dan putih saja, disini kami menggunakan aplikasi mobile 8bit painter dan menyusun ulang kode QR mengikuti pola dari QR yang masih warna-warni. setelah selesai diperbaiki QR yang asli akhirnya bisa di scan dan kita memperoleh flag aslinya.



```
flag: bronco{th1s_0n3_i5} *outsolve
```