**WRITEUP CTF GKSK #6**

| **Nama** | **Gede Rico Wijaya** |
| --- | --- |
| **Asal Universitas** | **Universitas Udayana** |
| **Contact Person** | **087733889701** |

****

**BIARIN HALAMAN INI KOSONG**

**+ HAP**

**US TULISANNYA PAS NGUMPUL**

**[MEME]**

**DAFTAR ISI**

[**[**KATEGORI**]** 5](#_gjdgxs)

[1. CHALL Name 5](#_30j0zll)

[a. [Technical Report] 5](#_1fob9te)

[b. [Proof Of Concept] 5](#_3znysh7)

[c. [FLAG] 5](#_2et92p0)

# **[**PWN]

## Baby\_Js

### [Technical Report]

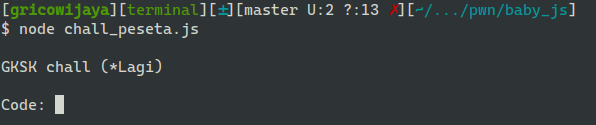
nc 103.145.226.170 3003

Author : Mockingjay

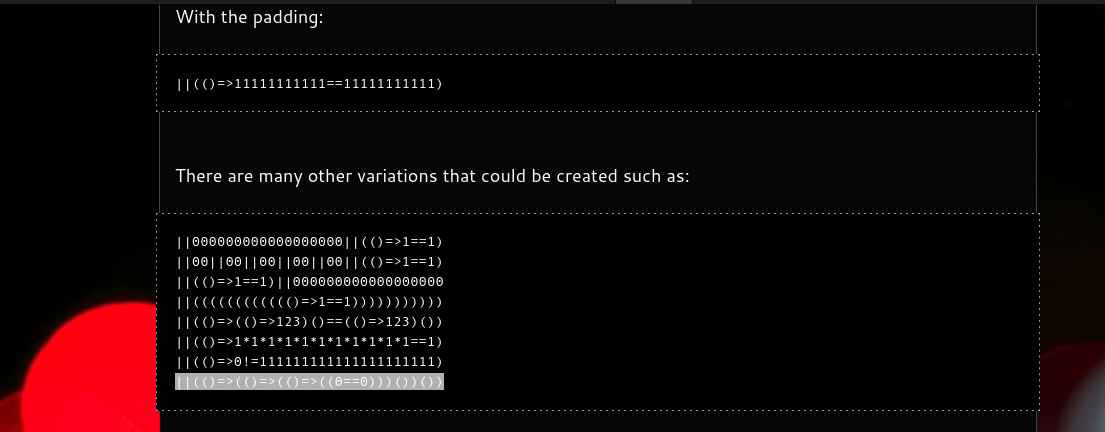
### [Proof Of Concept]

Dari Description disebutkan didapatkan file yang dinamakan \_chall\_peserta.js\_ lalu dari file tersebut dapat dicoba untuk memasukkan password yaitu seperti dibawah ini.

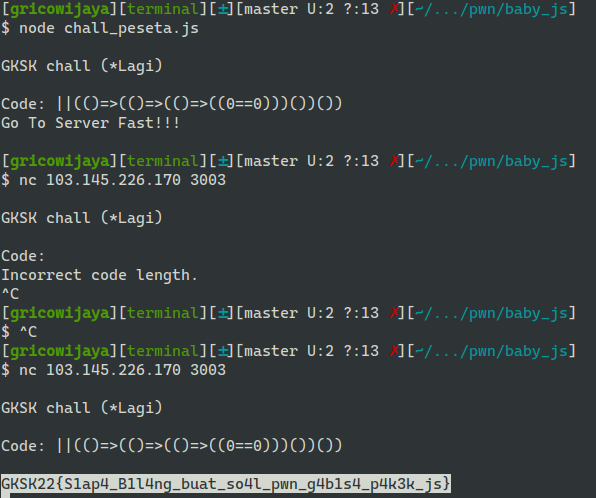
﻿



Lalu saya mendapat referensi dari web ini https://fadec0d3.blogspot.com/2018/04/midnight-sun-ctf-quals-2018-babyshells.html sehingga dari web tersebut akan menggunakan input padding yang dimana dari referensi dibawah ini



Dari beberapa contoh inputan diatas yang paling bawah sehingga nantinya percobaan tersebut akan mengembalikan "Go To Server Fast!!!" seperti gambar dibawah ini. Lalu dari netcat 103.145.226.170 3003 dapat didapatkan dari inputan ||(()=>(()=>(()=>((0==0)))())()) adalah flag yang dimana \*\*GKSK22{S1ap4\_B1l4ng\_buat\_so4l\_pwn\_g4b1s4\_p4k3k\_js}\*\*



### [FLAG]

GKSK22{S1ap4\_B1l4ng\_buat\_so4l\_pwn\_g4b1s4\_p4k3k\_js}

# **[**Forensics]

## Sniff Sniff

### [Technical Report]

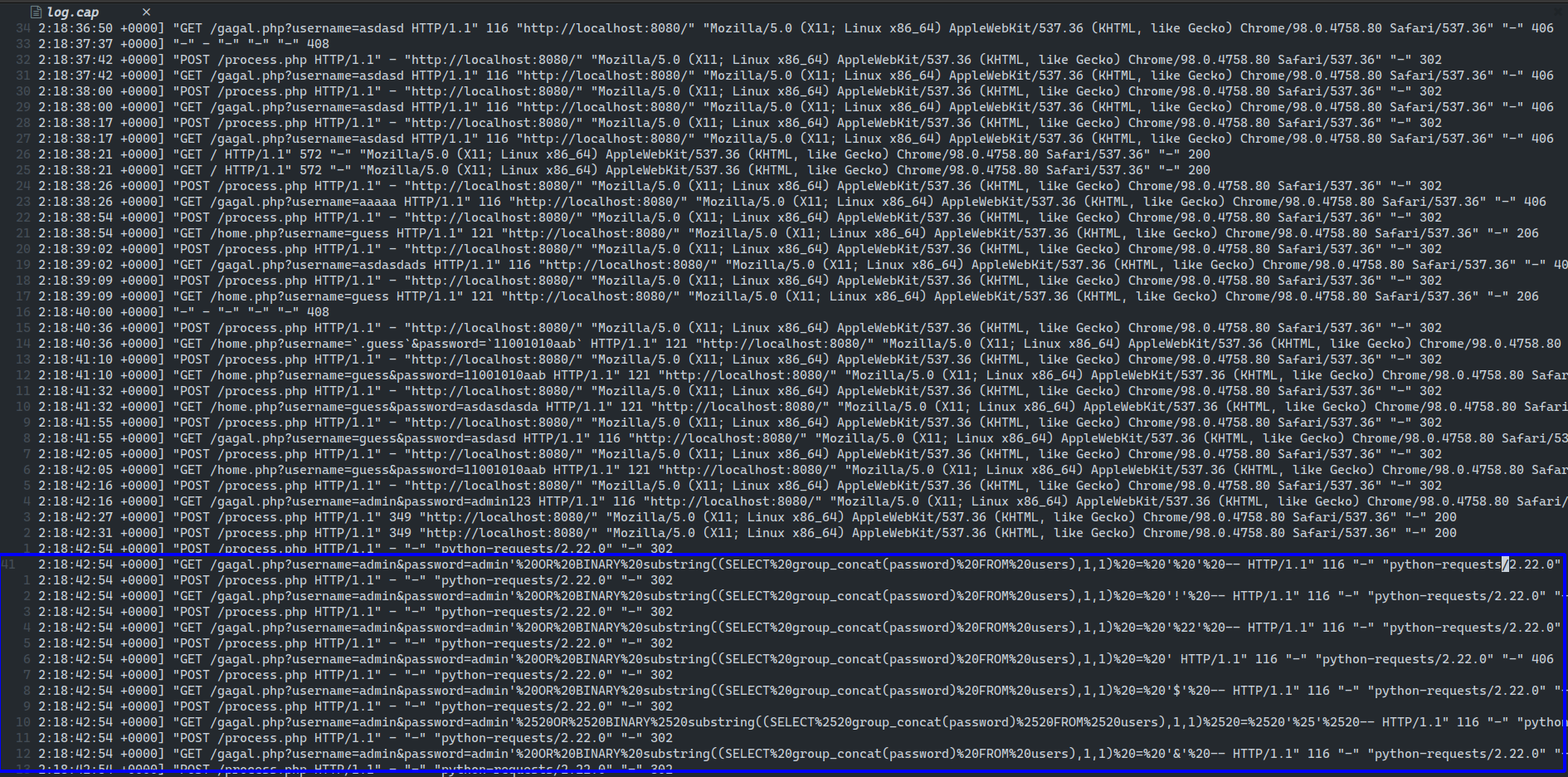
Bjyr serverku diserang hacker, tolong cari tau apa password yg di curi oleh si hacker!!! fast!!!!

format flag : GKSK{}

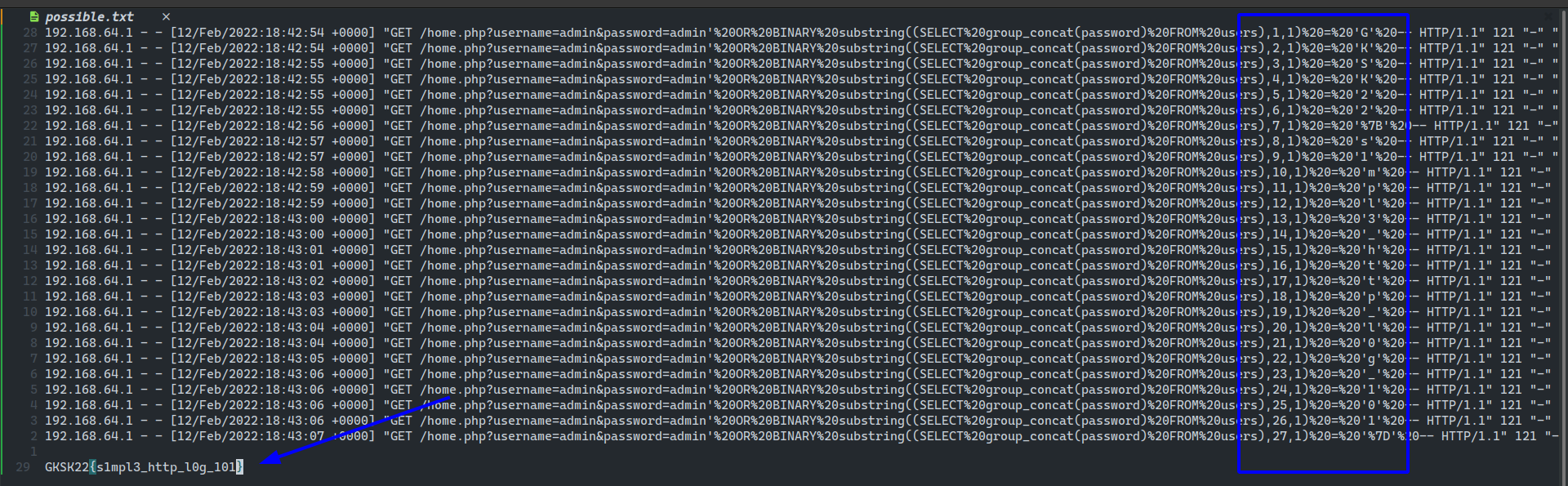
author: Mockingjay

### [Proof Of Concept]

Dalam Challenge ini saya melakukan download untuk file log.cap tersebut yang dimana merupakan file yang log dari sebuah server yang dimana nantinya digunakan untuk melihat bahwa terdapat sebuah \_bruteforce\_ seperti dibawah ini



Dari analisa diatas terdapat sebuah pola yang dimana jika proses pencarian password dari user itu berhasil maka akan redirect ke GET /home.php maka dari saya menggunakan piping yang dimana melakukan cat log.cap dengan grep "GET /home" lalu disimpan pada file possible\_flag.txt



Dari hal tersebut dapat disusun karakter-karakter yang berhasil di brute-force pada log.cap tersebut sehingga menemukan flag \_\_GKSK22{s1mpl3\_http\_l0g\_101}\_\_

### [FLAG]

GKSK22{s1mpl3\_http\_l0g\_101}

# **[**Reverse Engineering]

## ez\_rev

### [Technical Report]

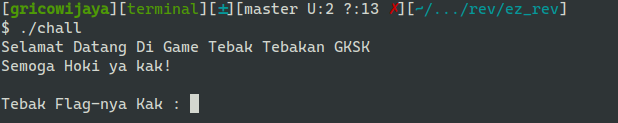
Gtw mau kasi desc apa kak h3h3h3h3

Coba aja

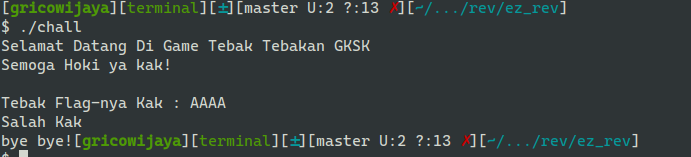
author : Mockingjay

### [Proof Of Concept]

Dalam Challenge ini dapat didapatkan sebuah file chall yang jika dijalankan akan seperti gambar dibawah ini.



Jika salah maka akan mengembalikan kata "Salah Kak \n bye-bye!" seperti dibawah ini.



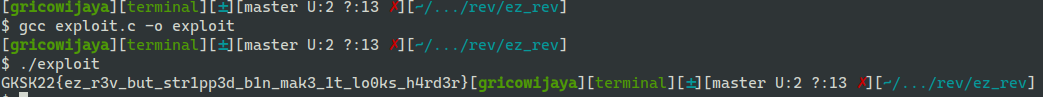
Maka dari itu dibutuhkan sebuah decompiler dengan menggunakan IDA Pro yang dimana nantinya akan mengembalikan sebuah fungsi-fungsi, dan terdapat sebuah fungsi yang menyimpan array yang dinamakan v6 dengan menggunakan tipe data int, sehingga saya berasumsi untuk array tersebut menyimpan karakter-karakter yang nantinya dapat akan dilakukan xor dengan binary hex 0x10 jika pada desimal adalah 16. Seperti Kode dibawah ini.

| \_\_int64 \_\_fastcall main(\_\_int64 a1, char \*\*a2, char \*\*a3)  {  int i; // [rsp+8h] [rbp-118h]  int j; // [rsp+Ch] [rbp-114h]  int v6[52]; // [rsp+10h] [rbp-110h]  char v7[56]; // [rsp+E0h] [rbp-40h] BYREF  unsigned \_\_int64 v8; // [rsp+118h] [rbp-8h]  v8 = \_\_readfsqword(0x28u);  v6[0] = 87;  v6[1] = 91;  v6[2] = 67;  v6[3] = 91;  v6[4] = 34;  v6[5] = 34;  v6[6] = 107;  v6[7] = 117;  v6[8] = 106;  v6[9] = 79;  v6[10] = 98;  v6[11] = 35;  v6[12] = 102;  v6[13] = 79;  v6[14] = 114;  v6[15] = 101;  v6[16] = 100;  v6[17] = 79;  v6[18] = 99;  v6[19] = 100;  v6[20] = 98;  v6[21] = 33;  v6[22] = 96;  v6[23] = 96;  v6[24] = 35;  v6[25] = 116;  v6[26] = 79;  v6[27] = 0x72;  v6[28] = 33;  v6[29] = 126;  v6[30] = 79;  v6[31] = 125;  v6[32] = 113;  v6[33] = 123;  v6[34] = 35;  v6[35] = 79;  v6[36] = 33;  v6[37] = 100;  v6[38] = 79;  v6[39] = 124;  v6[40] = 127;  v6[41] = 32;  v6[42] = 123;  v6[43] = 99;  v6[44] = 79;  v6[45] = 120;  v6[46] = 36;  v6[47] = 98;  v6[48] = 116;  v6[49] = 35;  v6[50] = 98;  v6[51] = 109;  ((void (\_\_fastcall \*)(\_\_int64, char \*\*, char \*\*))((char \*)&sub\_11A8 + 1))(a1, a2, a3);  printf("\n\nTebak Flag-nya Kak : ");  \_\_isoc99\_scanf("%s", v7);  for ( i = 0; v7[i]; ++i )  ;  for ( j = 0; j < i; ++j )  {  if ( (char)(v7[j] ^ 0x10) != v6[j] )  {  puts("Salah Kak");  break;  }  puts("Selamat Kak");  }  printf("bye bye!");  return 0LL;  } |
| --- |

Dari hal tersebut saya hanya membuat file exploitation dengan bahasa c ynag dimana digunakan untuk mengubah setiap integer lalu dibuat xor dengan bilangan 16 lalu dilakukan typecast menjadi character seperti dibawah ini.

| #include <stdio.h>  int main() {  int v6[52];  v6[0] = 87;  v6[1] = 91;  v6[2] = 67;  v6[3] = 91;  v6[4] = 34;  v6[5] = 34;  v6[6] = 107;  v6[7] = 117;  v6[8] = 106;  v6[9] = 79;  v6[10] = 98;  v6[11] = 35;  v6[12] = 102;  v6[13] = 79;  v6[14] = 114;  v6[15] = 101;  v6[16] = 100;  v6[17] = 79;  v6[18] = 99;  v6[19] = 100;  v6[20] = 98;  v6[21] = 33;  v6[22] = 96;  v6[23] = 96;  v6[24] = 35;  v6[25] = 116;  v6[26] = 79;  v6[27] = 0x72;  v6[28] = 33;  v6[29] = 126;  v6[30] = 79;  v6[31] = 125;  v6[32] = 113;  v6[33] = 123;  v6[34] = 35;  v6[35] = 79;  v6[36] = 33;  v6[37] = 100;  v6[38] = 79;  v6[39] = 124;  v6[40] = 127;  v6[41] = 32;  v6[42] = 123;  v6[43] = 99;  v6[44] = 79;  v6[45] = 120;  v6[46] = 36;  v6[47] = 98;  v6[48] = 116;  v6[49] = 35;  v6[50] = 98;  v6[51] = 109;  for (int i = 0; i < 52; i++) {  v6[i] = v6[i] ^ 16;  printf("%c", (char)v6[i]);  }    return 0;    } |
| --- |

Dari Kode program diatas maka dapat dilakukan compile dengan gcc dan setelah itu akan didapatkan flag seperti dibawah ini.



Kode program tersebut digunakan untuk melakukan looping sebanyak 52 kali dengan mengecek setiap array v6 dan setiap indeksnya dilakukan xor dengan bilangan 16 sehingga nantinya akan di typecast menjadi character lalu dilakukan outputnya.

### [FLAG]

GKSK22{ez\_r3v\_but\_str1pp3d\_b1n\_mak3\_1t\_lo0ks\_h4rd3r}

# **[**Web]

## Bomb

### [Technical Report]

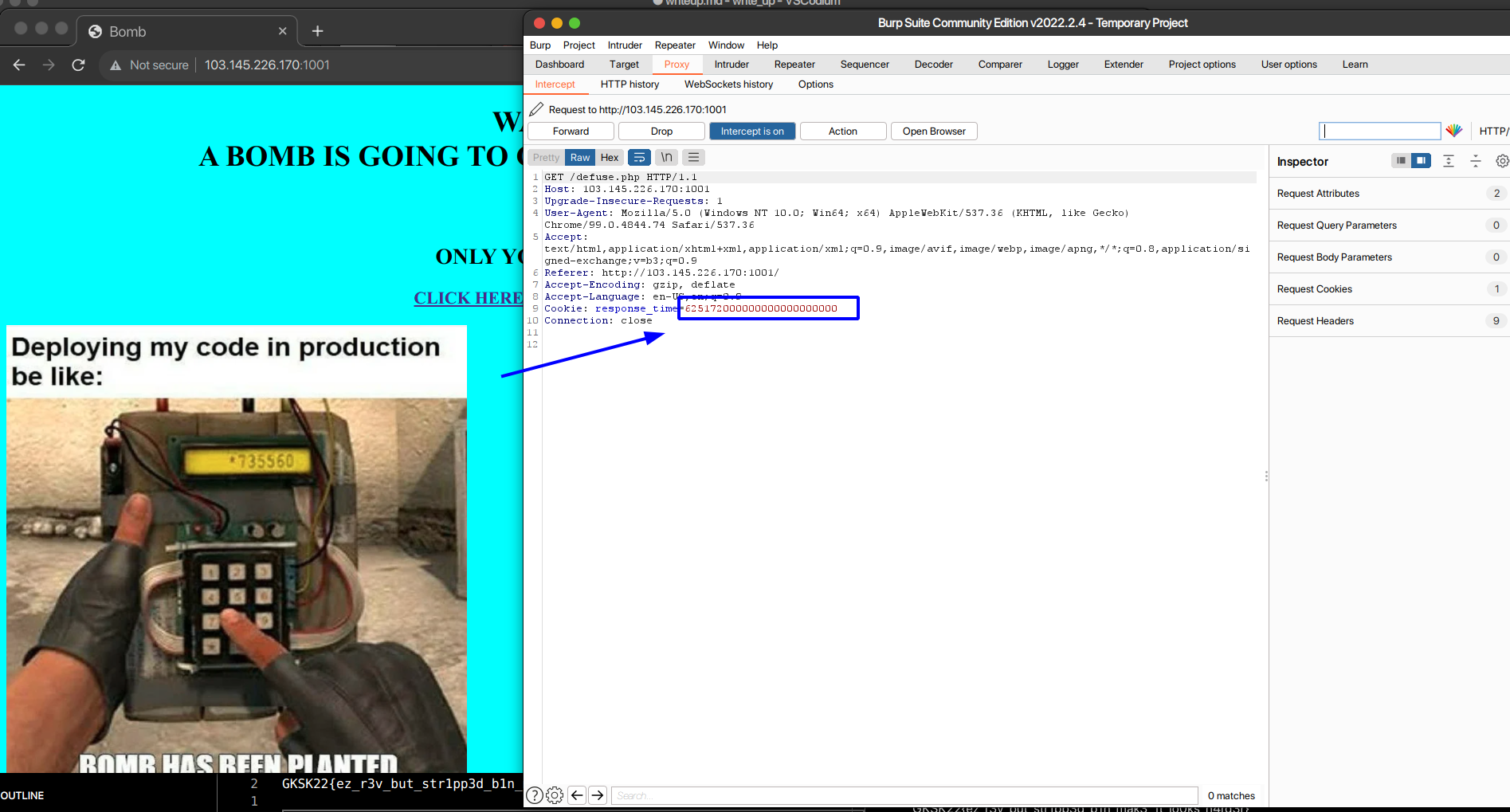
Bang bantu jinakin bomb nya bang...

http://103.145.226.170:1001/

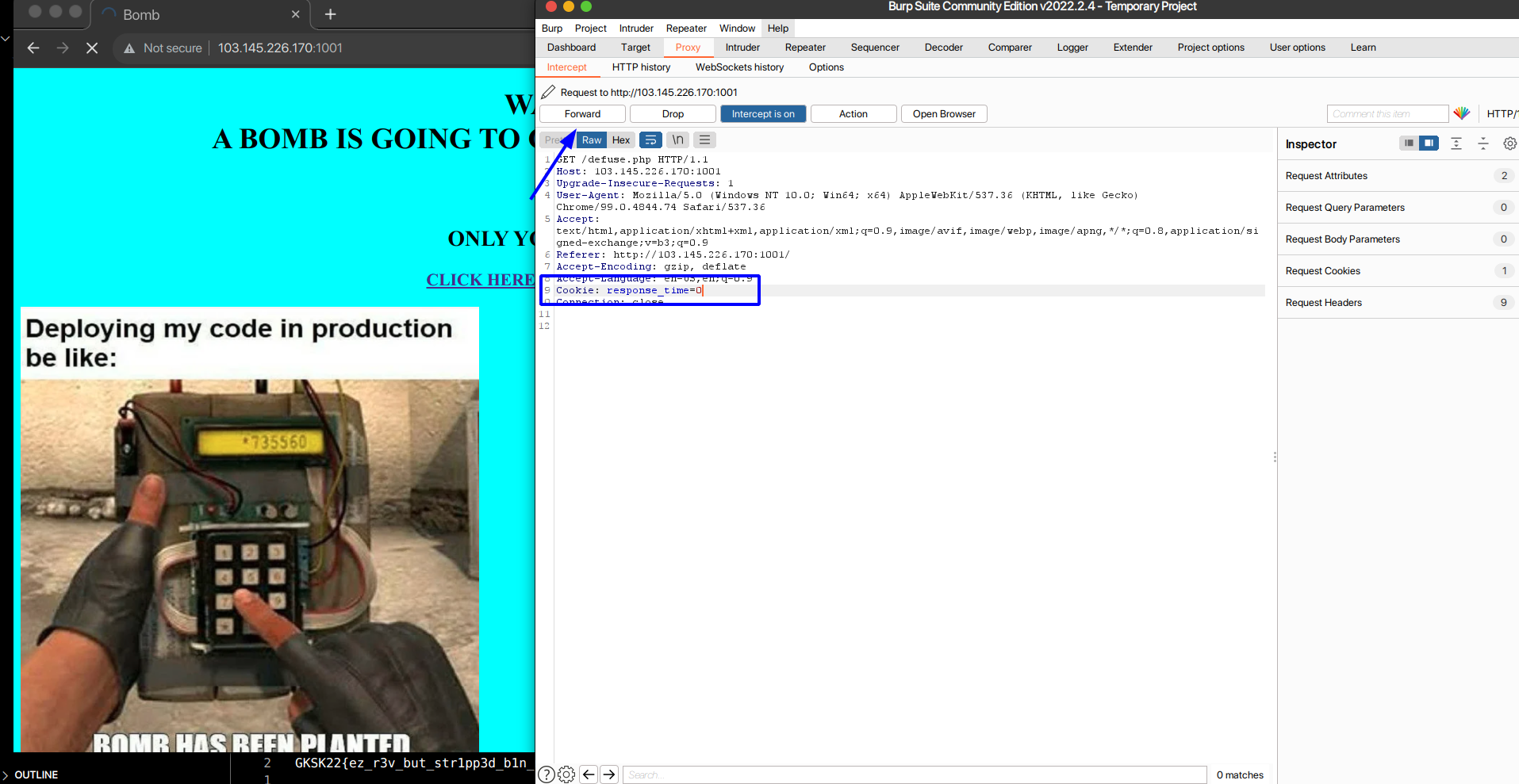
Author : Mockingjay

### [Proof Of Concept]

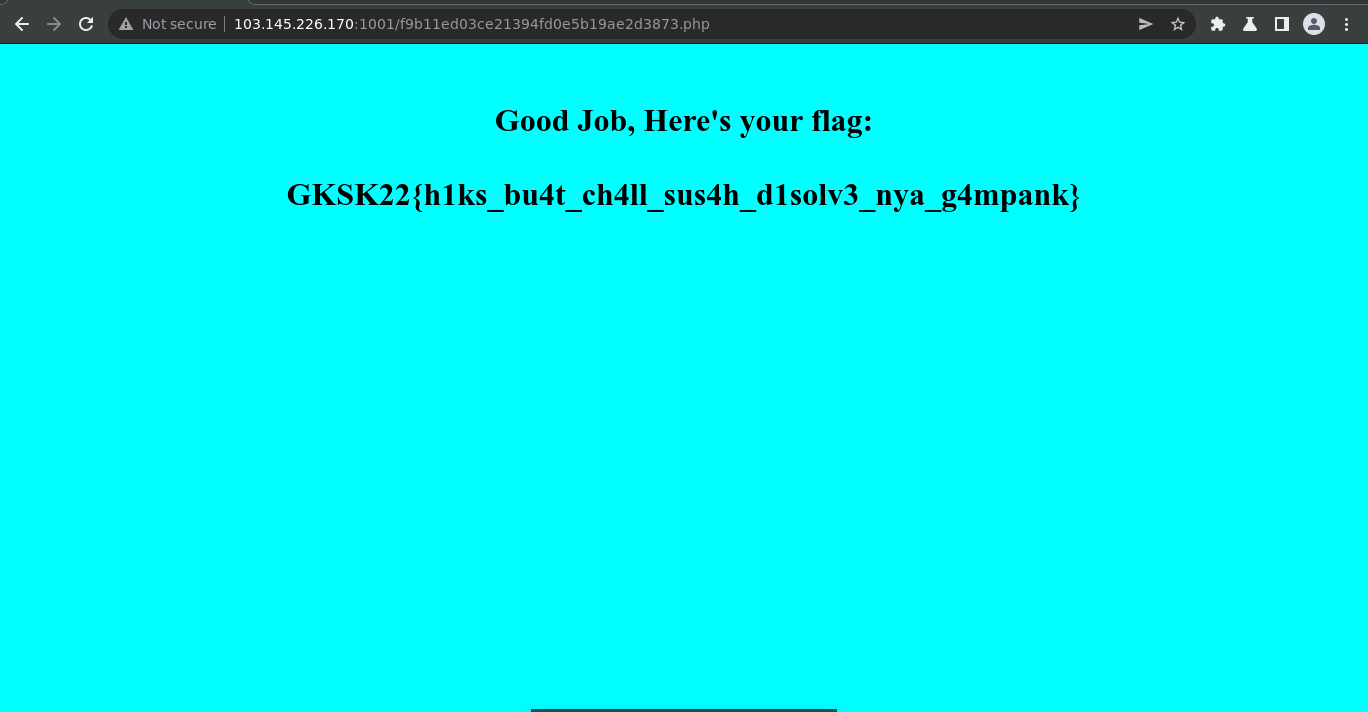
Dari Challenge ini dibutuhkan sebuah Aplikasi yaitu Burp Suite yang digunakan untuk melakukan pergantian request cookies response time seperti dibawah ini.



Dari hal tersebut nantinya akan melakukan forwarding kepada web yang diberikan setelah mengubah response time menjadi 0.



Lalu kita tekan butonn yang disediakan pada web tersebut sehingga akan mendapatkan flag yang diberikan karena defuse bomb tersebut diminta kurang dari 2 detik maka response time untuk cookies nya juga harus menjadi kurang dari 2 detik.



### [FLAG]

GKSK22{h1ks\_bu4t\_ch4ll\_sus4h\_d1solv3\_nya\_g4mpank}

# **[**Misc]

## Free

### [Technical Report]

Free?

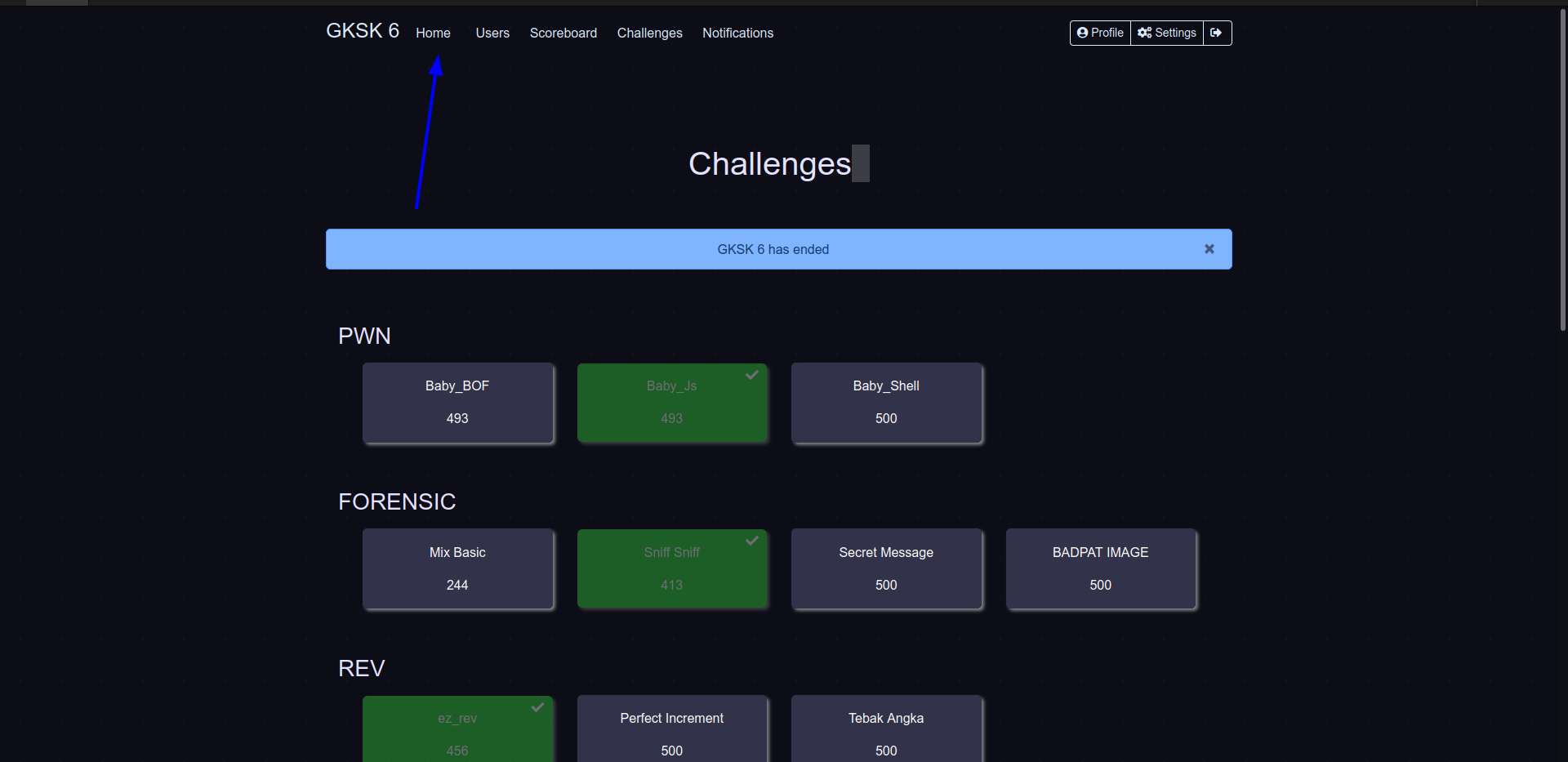
10

Admin dan probset naro flag di platform ini, ayo tebak dimana?

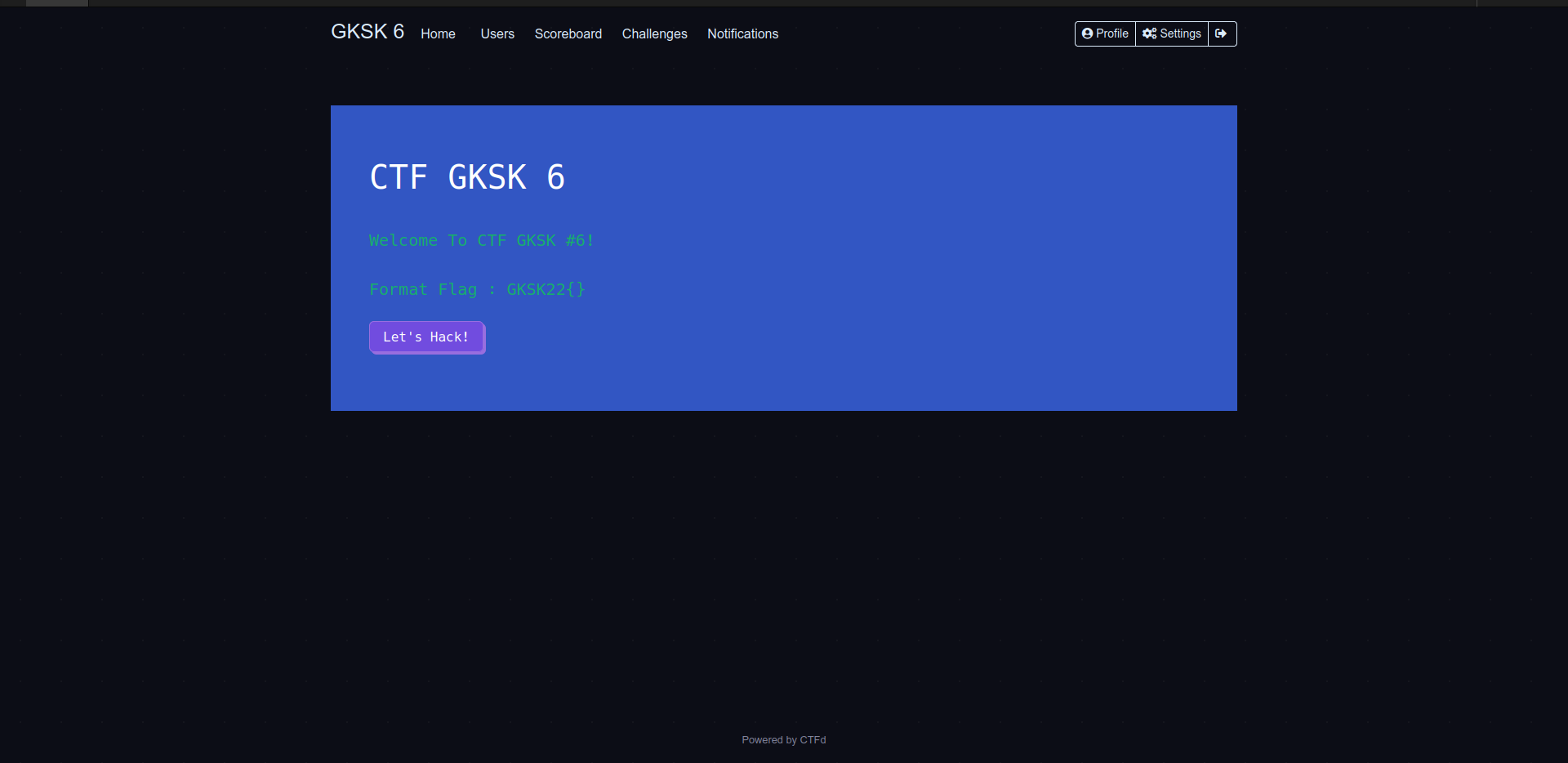
Author : Mockingjay

### [Proof Of Concept]

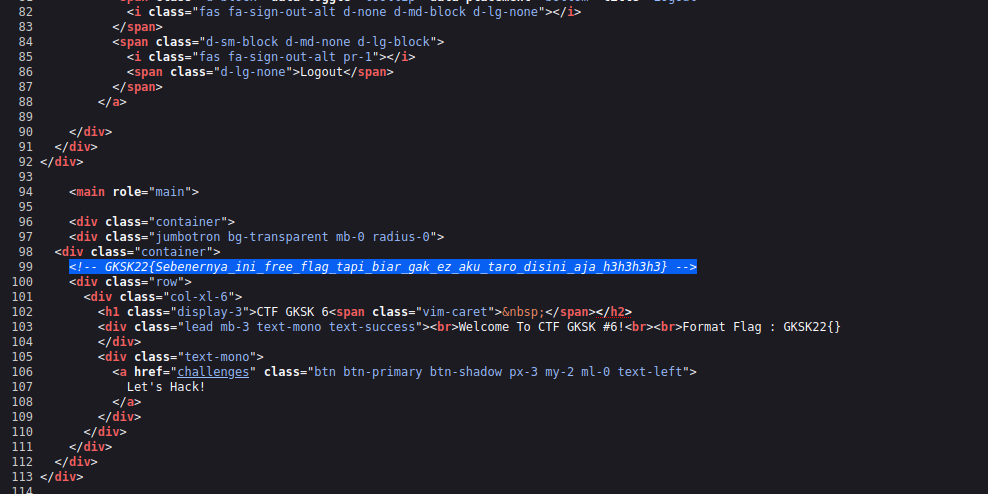
Dari hintnya diberikan pada bagian home dashboardnya dengan melakukan klik button dibawah ini.



Tetap Jika dilihat pada home dashboardnya hanya terdapat sebuah format flag.



Maka dari itu saya coba melakukan inspect code dengan melakukan Ctrl+u pada browser saya dan terdapat comment yang menuliskan \_\_GKSK22{Sebenernya\_ini\_free\_flag\_tapi\_biar\_gak\_ez\_aku\_taro\_disini\_aja\_h3h3h3h3} \_\_ yang dituliskan pada line 99 seperti dibawah ini.



### [FLAG]

GKSK22{Sebenernya\_ini\_free\_flag\_tapi\_biar\_gak\_ez\_aku\_taro\_disini\_aja\_h3h3h3h3}

# **[**OSINT]

## Kota Rahasia

### [Technical Report]

Mimin dikasi alamat kota rahasia nihh

Kota ini berjarak sekitar

1496,7 km dari DARWIN,

2265,2 km dari HO CHI MINH ,

2204 km dari MANILA ,

Tolong bantu cariin kotanya dong !

\*Format flag : Flag ditulis UPPERCASE : GKSK22{NAMA KOTA}

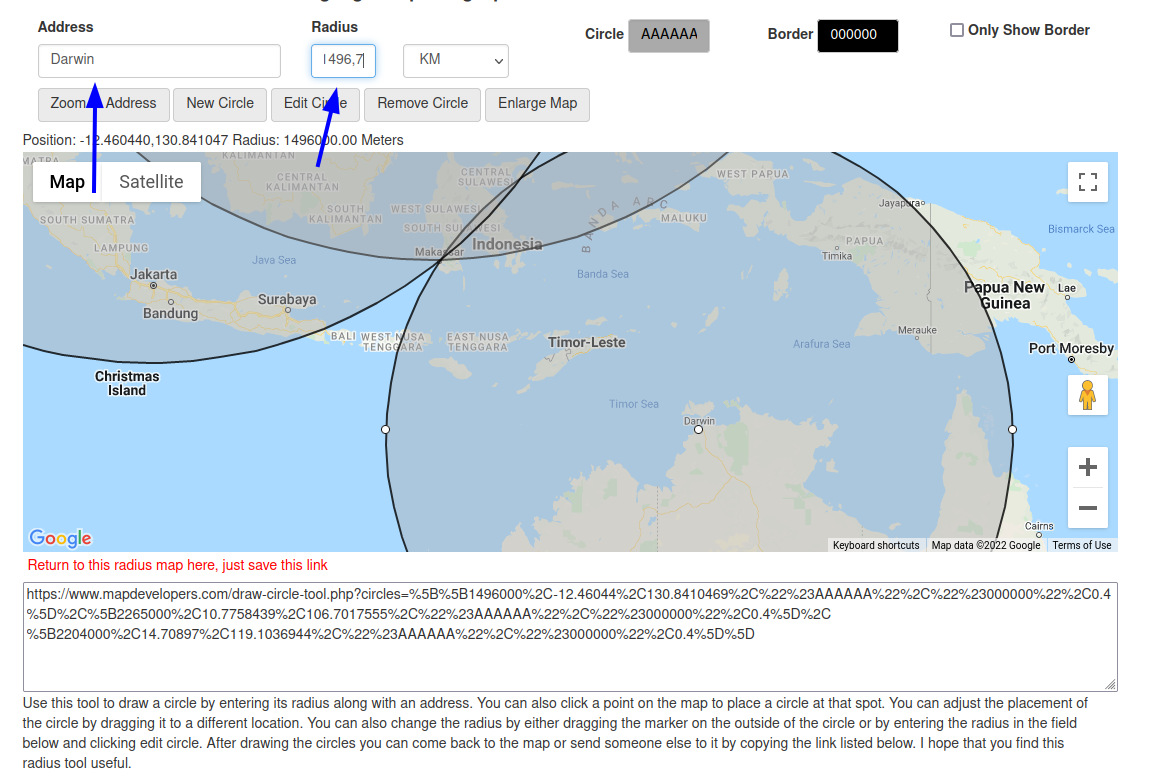
Contoh : GKSK22{KONOHA}

Author : Little\_Krisna

### [Proof Of Concept]

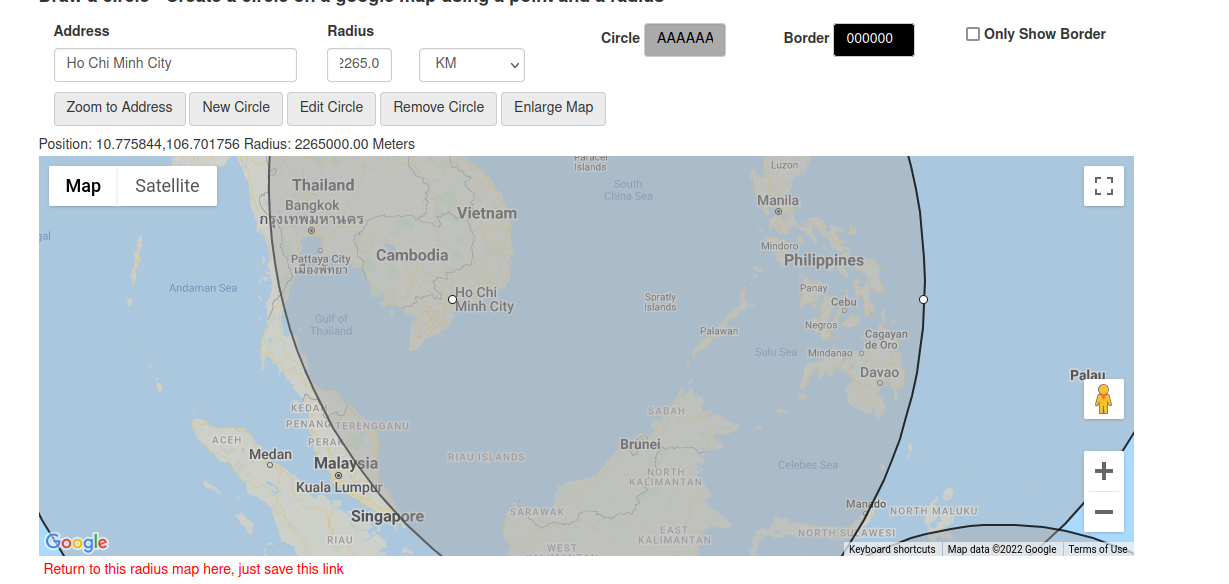
OSINT ini digunakan untuk melakukan sebuah pencarian lokasi, karena diberikan 3 kota dengan beberapa jarak maka saya akan gunakan sebuah metode untuk membuat sebuah radius lingkaran dari setiap jarak yang diberikan. Nantinya kemungkinan dari jarak-jarak tersebut akan mendapat sebuah 1 titik silang atau titik temu dari 3 radius tersebut. Untuk hal itu saya dapat menggunakan web ini dalam mebuat radius tersebut. https://www.mapdevelopers.com/draw-circle-tool.php

Untuk yang pertama maka saya mengetikan Darwin beserta dengan jarak radiusnya yaitu 1496,7 km sehingga akan terlihat seperti dibawah ini.

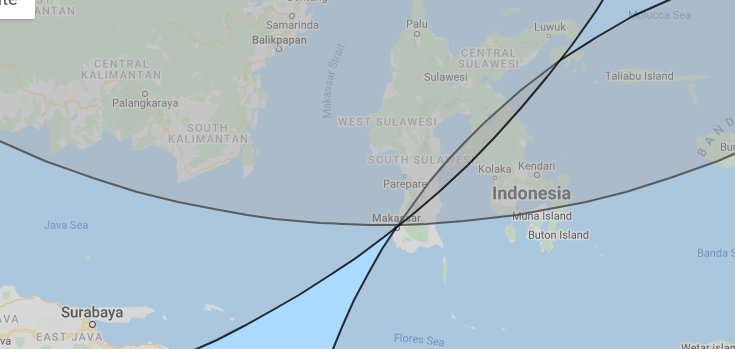


Untuk Radius yang kedua maka saya mengetikan Ho Chi Minh City dengan radius 2265,2 km sehingga nantinya akan terlihat seperti dibawah ini.

.



Jadi setelah ketiga lingkaran tersebut sudah terbuat maka nantinya akan terdapat titik yang dimana ketika lingkaran tersebut bertemu seperti \_intersection\_, dapat dilihat pada gambar dibawah ini.



Dari titik tersebut lah saya mencoba untuk menginput file yang dimana sesuai dengan format flag adalah GKSK22{NAMA KOTA} dengan uppercase sehingga flag diterima yaitu GKSK22{MAKASSAR}

### [FLAG]

GKSK22{MAKASSAR}