

"CompFest 9"

# TABLE OF CONTENTS

## BINARY EXPLOTATION

Artificial Intelligence .....	3
-------------------------------	---

## FORENSIC

Dark and Deep .....	5
---------------------	---

## WEB

Wired.js .....	6
Can You Get The Admin File? .....	6
Jackpot! .....	8

## REVERSE ENGINEERING

Bandicoot .....	9
Not So Classic String Validator .....	10

## CRYPTOGRAPHY

Cough Generator .....	12
Can You Get The Plain Text? .....	14

## 1. Artificial Intelligence - 50

- Didapat koneksi `nc tenkai.compest.web.id 26079` dan sebuah file ELF 64 Bit bernama `ai`, kami coba analisa file tersebut dengan aplikasi IDA maka terdapat sebuah fungsi `game` dengan sintak yang sudah terubah dalam bahasa C seperti berikut,

```
__int64 game()
{
    signed __int64 v0; // rdx@3
    signed __int64 v1; // rdx@11
    unsigned int v3; // [sp+8h] [bp-128h]@7
    unsigned int v4; // [sp+Ch] [bp-124h]@1
    unsigned int v5; // [sp+10h] [bp-120h]@1
    unsigned int v6; // [sp+14h] [bp-11Ch]@2
    FILE *stream; // [sp+18h] [bp-118h]@14
    char ptr; // [sp+20h] [bp-110h]@14
    __int64 v9; // [sp+128h] [bp-8h]@1

    v9 = *MK_FP(__FS__, 40LL);
    puts("Welcome to Stone Game!");
    puts("You can pick 1-5 stones from pile for each turn. The first one who make the pile empty is the winner.\n");
    puts("Computer play first\n");
    v5 = 4;
    v4 = 50;
    while ( 1 )
    {
        puts("-- Computer Turn --");
        printf("Number of stones: %d\n", v4);
        v6 = 6 - v5;
        if ( (signed int)(6 - v5) <= 1 )
            v0 = 0LL;
        else
            v0 = 115LL;
        printf("Computer pick %d stone%c\n", v6, v0);
        v4 -= v6;
        if ( (signed int)v4 <= 0 )
        {
            puts("You Lose!");
            return *MK_FP(__FS__, 40LL) ^ v9;
        }
        puts("-- Your Turn --");
        printf("Number of stones: %d\n", v4);
        printf("How many stones you want to pick? ");
        _isoc99_scanf("%d", &v3);
    }
}
```

```

if ( (unsigned __int16)invalid(v3) )
{
    puts("Invalid number of stones!");
    exit(0);
}
v1 = (signed int)v3 <= 1 ? 0LL : 115LL;
printf("You pick %d stone%c\n\n", v3, v1);
v4 -= v3;
if ( (signed int)v4 <= 0 )
    break;
v5 = v3;
}
puts("You Won!");
stream = fopen("flag.txt", "r");
fread(&ptr, 1uLL, 0x1DuLL, stream);
fclose(stream);
puts(&ptr);
return *MK_FP(__FS__, 40LL) ^ v9;
}

```

- Vulnerable pada misi ini adalah ada pada integer overflow yaitu dimana apabila diinputkan nilai sebesar **-2147483647** maka program akan menampilkan flag seperti gambar berikut,

```

C:\Users\JDoor>nc tenkai.compfest.web.id 26079
Welcome to Stone Game!
You can pick 1-5 stones from pile for each turn. The first one who make the pile empty is the winner.

Computer play first

-- Computer Turn --
Number of stones: 50
Computer pick 2 stones

-- Your Turn --
Number of stones: 48
How many stones you want to pick? -2147483647
You pick -2147483647 stone

You Won!
COMPFEST9{int3g3r_155u35_101}

```

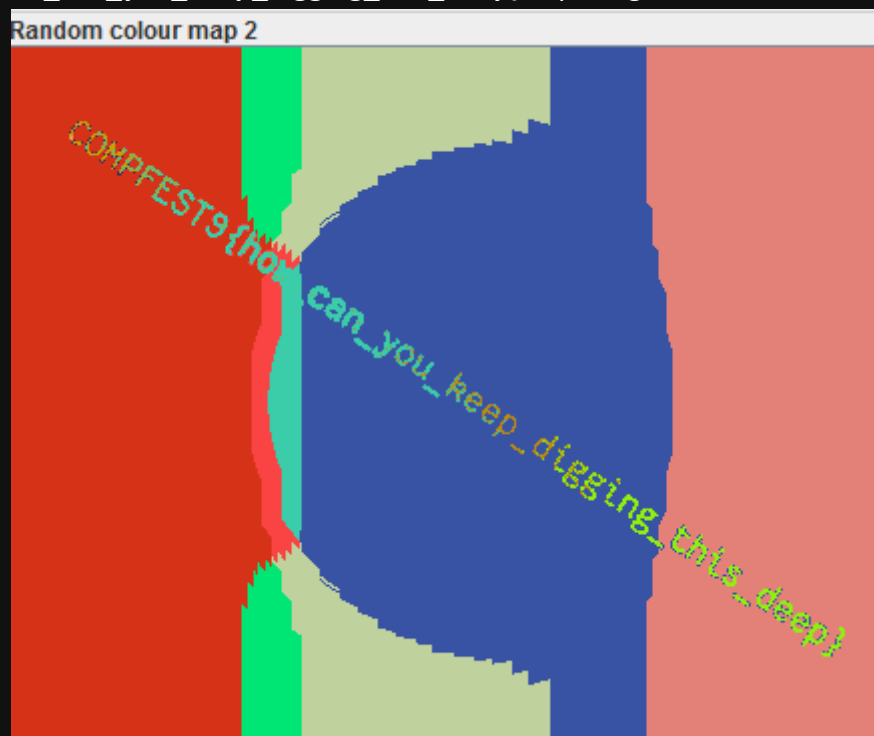
- Didapatlah flag yaitu : **COMPFEST9{int3g3r\_155u35\_101}**.

## 2. Dark and Deep - 250

- Terdapat sebuah file darkdeep.zip apabila di ekstrak maka akan terdapat banyak direktori yang mana sepertinya copy-an dari isi direktori file system yang ada di linux.
- Lalu kami mencurigai adanya sebuah file tersembunyi yang bernama .ash\_history pada direktori root dimana mengarahkan pada sebuah clue menuju direktori tmp, dan setelah melakukan perintah ls -al kami menemukan file .broken.zip dan ternyata yah.. File tersebut corrupt,

```
jdoor@JDoor:~/Downloads/darkdeep$ ls
bin dev etc home lib media mnt proc root run sbin srv sys tmp usr var
jdoor@JDoor:~/Downloads/darkdeep$ cat root/.ash_history
ls
shred --help
rm --help
apk update
apk add shred
ls
cd /tmp
ls
rm darkdeep-broken.zip
ls
cat ~/.ash_history
apk cache clean
rm -rf /var/cache/apk
jdoor@JDoor:~/Downloads/darkdeep$ ls tmp/.broken.zip
tmp/.broken.zip
jdoor@JDoor:~/Downloads/darkdeep$
```

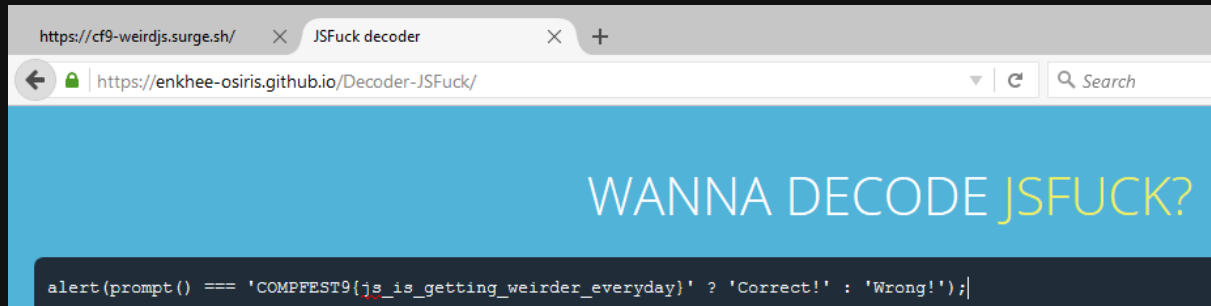
- Kami coba lihat pada bagian hex file tersebut ternyata terdapat keganjilan pada bagian nilai hex panjang nama file tersebut (darkdeep-02.png) dimana kami mendapat referensi dari halaman <https://users.cs.jmu.edu/buchhofp/forensics/formats/pkzip.html>, lalu kami ubah nilai hex yang semula 00 menjadi 0F lalu repair menggunakan aplikasi winrar dan didapatkan sebuah gambar hitam pekat, buka dengan stegsolve lalu geser ke bagian random color dan didapatkan flag yaitu : **COMPFEST9{how\_can\_you\_keep\_digging\_this\_deep}** seperti gambar berikut,



# WEB

## 1. Wired.js - 25

- Diberikan link yang mengarah ke <https://cf9-weirdjs.surge.sh/> dimana apabila dilihat pada bagian source maka akan terdapat sebuah javascript yang berjenis jsfuck, decode strings tersebut pada web <https://enkhee-osiris.github.io/Decoder-JSFuck/> maka akan mendapat flag yaitu : **COMPFEST9{js\_is\_getting\_weirder\_everyday}** seperti gambar berikut,

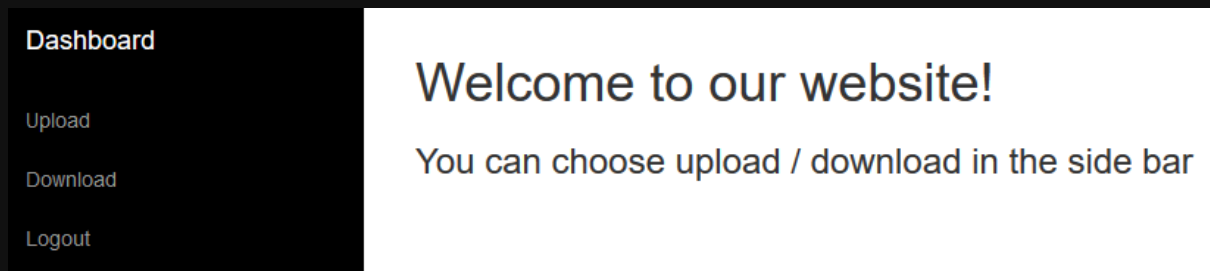


## 2. Can You Get The Admin File? - 50

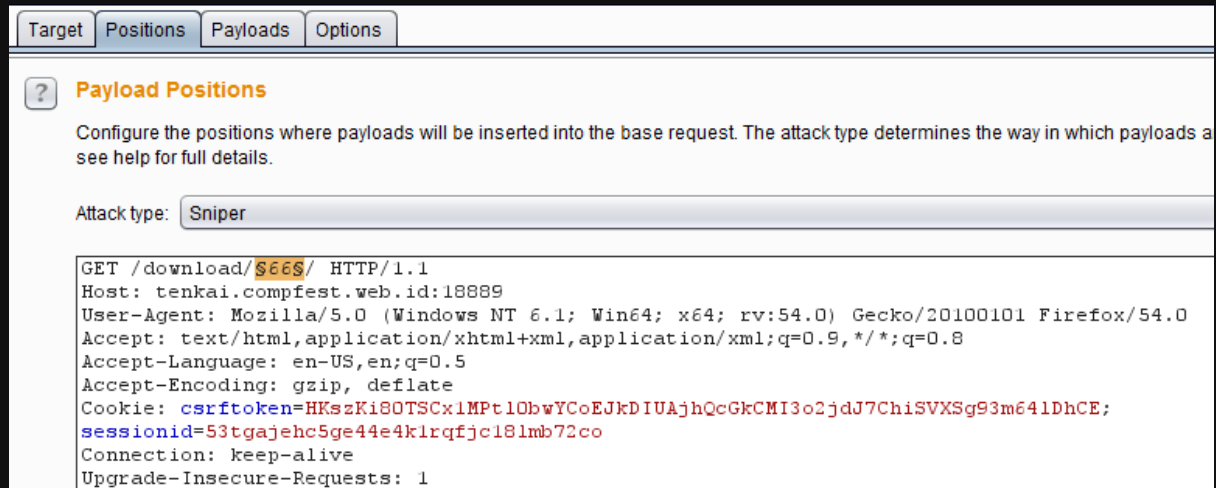
- Terdapat sebuah link yang mengarah ke <http://tenkai.compfest.web.id:18889/> dengan tampilan seperti berikut,

A screenshot of a login form with a green border. It contains two input fields: 'Username:' and 'Password:'. Below the password field is a green 'LOGIN' button. At the bottom, there is a link that says 'Not registered? Create an account'.

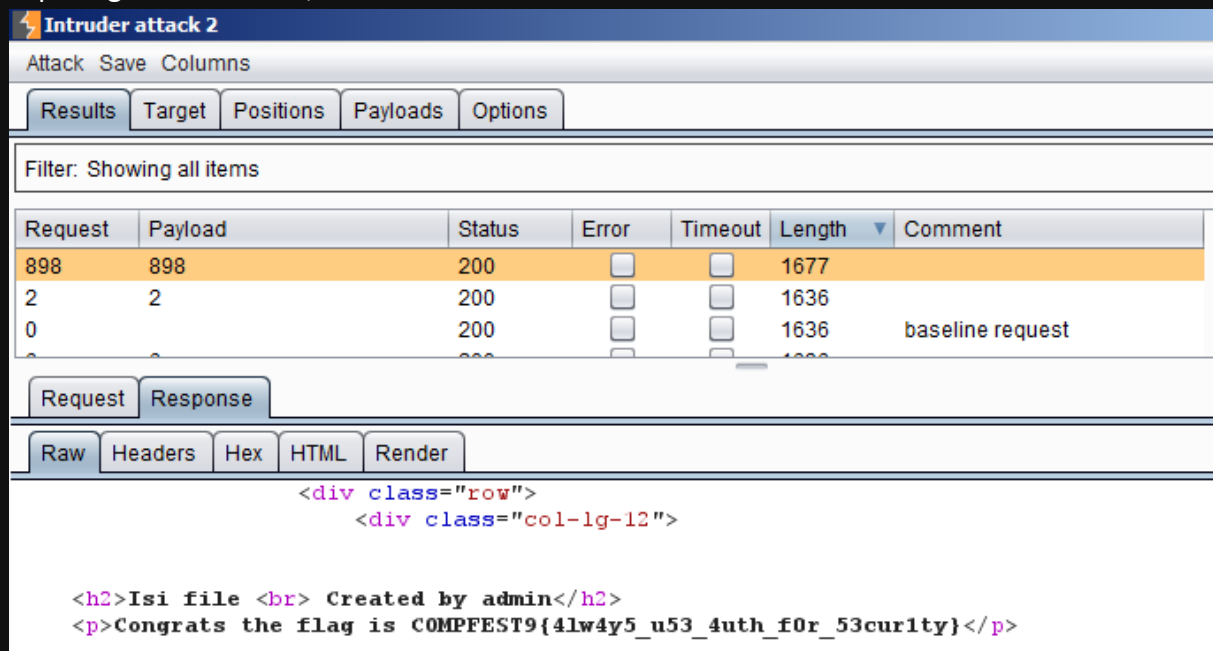
- Register lalu login maka akan menampilkan halaman seperti berikut,



- Mencoba mengupload beberapa teks maka akan terupload pada link <http://tenkai.compfest.web.id:18889/download/2160/>, kami coba mengakses pada direktori 1, 500, 1000, 1500, dan 2000 ternyata pada direktori tersebut admin sudah mengupload sesuatu, lalu disini kami coba bruteforce semua direktori dari 1 – 2000 menggunakan burp seperti berikut,

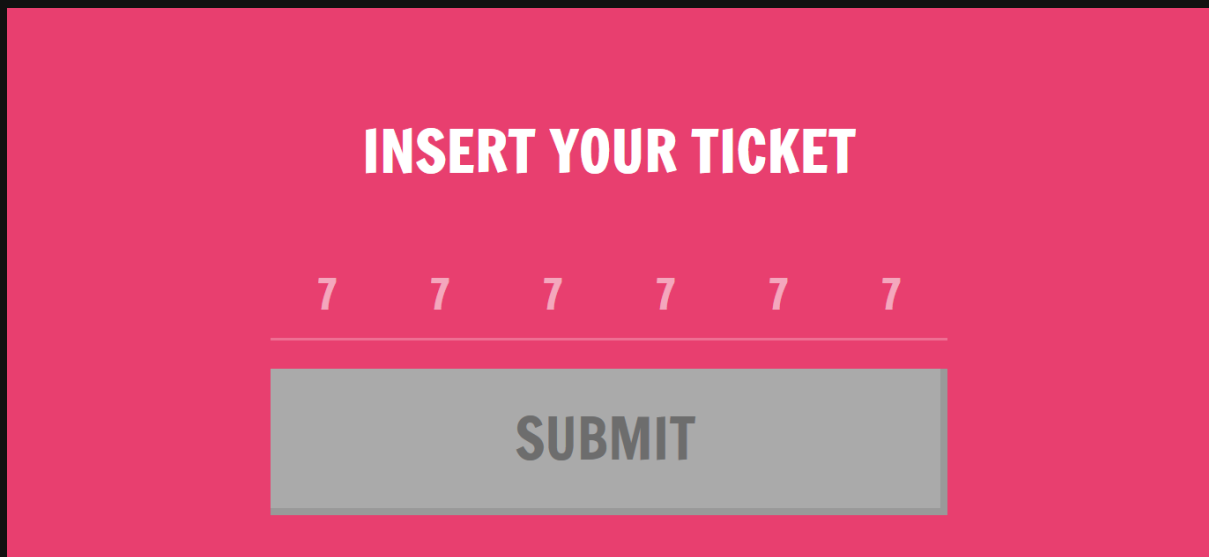


- Dari hasil yang didapat kami cari berdasarkan length dimana terdapat yang paling besar berada pada direktori **898** dan terdapat pula flag yaitu : **COMPFEST{ 4lw4y5\_u53\_4uth\_f0r\_53cur1ty}** seperti gambar berikut,



### 3. Jackpot - 150

- Terdapat sebuah link yang mengarah ke <https://cf9-jackpot-dtbznhqsyf.now.sh/> dengan tampilan seperti berikut,



**INSERT YOUR TICKET**

7 7 7 7 7 7

**SUBMIT**

- Kami coba untuk melihat source dari web tersebut dimana terdapat pada app.tag, kemudian terlihat bahwa variabel tiket terdiri dari 6 elemen dan isi dari variabel indexes yaitu 0 sampai 5, karena clue yang ada merupakan “Jackpot” kami mencoba untuk menginputkan mulai dari 1,1,1,1,1,1 hingga 5,5,5,5,5,5.



**CONGRATULATIONS! THE FLAG IS**

**COMPFEST9{BRUTEFORC3\_ALL\_THE\_THINGS}**

5 5 5 5 5 5

**SUBMIT**

- Flag pun didapat yaitu : **COMPFEST9{BRUTEFORC3\_ALL\_THE\_THINGS}**.



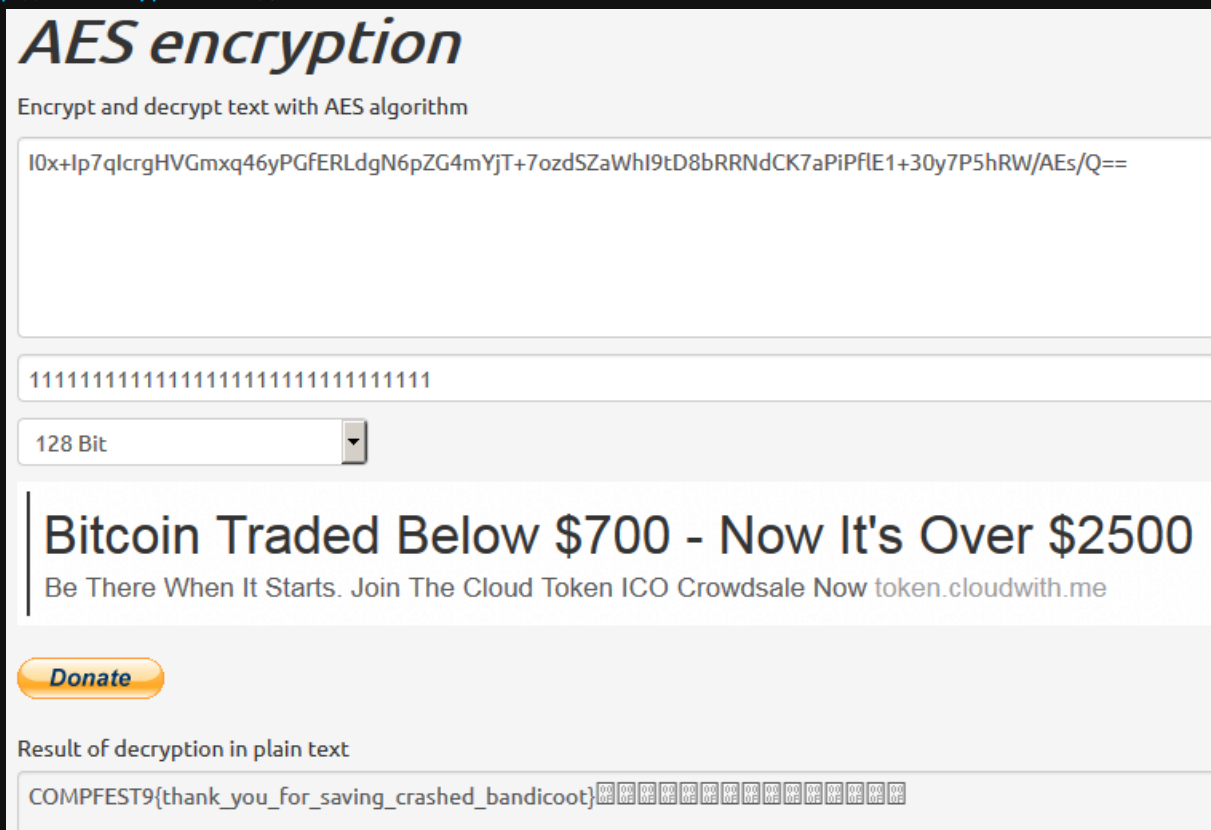
# REVERSE ENGINEERING

## 1. Bandicoot - 50

- Diberikan sebuah file bandicoot.apk dimana apabila didecompile maka akan terdapat hal menarik seperti pada file BrokenActivity.java berikut,

```
public void onLowMemory() {  
    String key = "1";  
    String tmp = "";  
    for (int i = 0; i <= 4; i++) {  
        key = new StringBuilder(key).reverse().toString() + key;  
    }  
    Toast.makeText(this, "The flag is " + decryptFlag(key.getBytes(), "I0x+Ip7qlcrgHVGmxq46yPGfERLdgN6pZG4mYjT+7ozdSZaWhI9tD8bRRNdCK7aPiPf1E1+30y7P5hRW/AEs/Q=="), 0).show();  
}
```

- Kami analisis script tersebut hingga mendapat fakta bahwa enkripsi yang digunakan adalah AES mode ECB 128 Bit dimana key bisa didapat dari perulangan pada variabel key adalah 11111111111111111111111111111111 dan lakukan decrypt pada halaman <http://aesencryption.net/>,



The screenshot shows the 'AES encryption' website interface. At the top, it says 'Encrypt and decrypt text with AES algorithm'. There are two input fields: the top one contains the encrypted string 'I0x+Ip7qlcrgHVGmxq46yPGfERLdgN6pZG4mYjT+7ozdSZaWhI9tD8bRRNdCK7aPiPf1E1+30y7P5hRW/AEs/Q==', and the bottom one contains the key '11111111111111111111111111111111'. Below these is a dropdown menu set to '128 Bit'. The main content area displays a decrypted message: 'Bitcoin Traded Below \$700 - Now It's Over \$2500' with a sub-header 'Be There When It Starts. Join The Cloud Token ICO Crowdsale Now token.cloudwith.me' and a 'Donate' button. At the bottom, under 'Result of decryption in plain text', the decrypted flag is shown: 'COMPFEST9{thank\_you\_for\_saving\_crashed\_bandicoot}'.

- Didapatlah flag yaitu **COMPFEST9{thank\_you\_for\_saving\_crashed\_bandicoot}**.

## 2. Not So Classic String Validator - 75

- Diberikan sebuah file ELF 64 Bit bernama validator lalu masuk pada fungsi main maka akan terlihat source code seperti berikut,

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
    int result; // eax@2
    double v4; // xmm0_8@4
    __int64 v5; // rcx@9
    int v6; // [sp+4h] [bp-5Ch]@1
    int i; // [sp+8h] [bp-58h]@3
    int v8; // [sp+Ch] [bp-54h]@1
    char s[72]; // [sp+10h] [bp-50h]@1
    __int64 v10; // [sp+58h] [bp-8h]@1

    v10 = *MK_FP(__FS__, 40LL);
    printf("Input Password: ", argv, envp);
    __isoc99_scanf(4196601LL, s);
    printf("Input integer constant: ");
    __isoc99_scanf(4196629LL, &v6);
    v8 = strlen(s);
    if ( v8 == 46 )
    {
        for ( i = 0; i < v8; ++i )
        {
            v4 = exp((double)s[i] * 13.0 / (double)v6);
            if ( (signed int)floor(v4) != dxdiag[i] )
            {
                result = 0;
                goto LABEL_9;
            }
        }
        printf("Congrats!\nYour flag is COMPFEST9{%s}\n", s);
        result = 0;
    }
    else
    {
        result = 0;
    }
}
```

- Dalam fungsi main program akan meminta input password user yang kemudian akan dimasukkan kedalam variabel s, selanjutnya program akan meminta input bilangan integer konstan yang kemudian akan dimasukkan kedalam variabel v6, apabila panjang dari inputan password sebanyak 46 karakter maka program akan berlanjut.
- Terdapat pula perulangan sebanyak 46 kali yang didalamnya terdapat operasi dimana nilai floor dari hasil exp dari karakter password dikali dengan 13.0 dan dibagi dengan konstan integer harus sama dengan array dxdiag seperti gambar berikut,

```
public dxdiag
g[]
dq 2526304, 442413, 77476, 6455302, 4318205, 1063, 1932310
; DATA XREF: main+C8↑r
dq 338391, 34669, 6455302, 2209435, 6942, 929, 4318205
dq 338391, 1292598, 4937507, 338391, 5645626, 1130470
dq 929, 338391, 505862, 756216, 67758, 5645626, 338391
dq 4937507, 5645626, 59260, 1292598, 34669, 988677, 338391
dq 578411, 1130470, 929, 578411, 1689944, 10377, 4318205
dq 338391, 756216, 7381098, 10377, 4318205
ends
```

- Kami mencoba melakukan bruteforce pada karakter pertama password dan bilangan konstan dengan membandingkan nilai array dxdiag ke 0 menggunakan script C++ berikut,

```
#include <stdio.h>
#include <math.h>
int main()
{
int data[46] =
{2526304,442413,77476,6455302,4318205,1063,1932310,338391,34669,6455302,2209435,6942,929,43
18205,338391,1292598,4937507,338391,5645626,1130470,929,338391,505862,756216,67758,5645626
,338391,4937507,5645626,59260,1292598,34669,988677,338391,578411,1130470,929,578411,168994
4,10377,4318205,338391,756216,7381098,10377,4318205};
    for(int j=65;j<122;j++){
        for(int i=1;i<9999;i++){
            double hasil = exp((double)j * 13.0 / (double)i);
            if((signed int)floor(hasil)==data[0]){
                printf("Nilai Konstan : %d",i);
            }
        }
    }
    return 0;
}
```

- Didapatlah hasil nilai konstan yaitu 97, kemudian kami mencoba melakukan brute input password dengan menggunakan script python berikut,

```
import math, sys

enc_flag =
[2526304,442413,77476,6455302,4318205,1063,1932310,338391,34669,6455302,2209435,6942,929,43
18205,338391,1292598,4937507,338391,5645626,1130470,929,338391,505862,756216,67758,5645626
,338391,4937507,5645626,59260,1292598,34669,988677,338391,578411,1130470,929,578411,168994
4,10377,4318205,338391,756216,7381098,10377,4318205]

key = 97

for i in range(46):
    for j in range(48,122):
        hasil = math.exp(float(j)*13.0 / float(key))
        if int(math.floor(hasil)) == enc_flag[i]:
            sys.stdout.write(chr(j))
```

- Running script diatas dan tambahkan format flag, maka didapatlah flag yaitu : **COMPFEST9{naTur4l\_NumB3r\_is\_th3\_beSt\_stRiNg\_ch3ckEr\_evEr}**.

## 1. Cough Generator - 25

- Didapat sebuah strings yang merupakan hasil encode dari `nc tenkai.compfest.web.id 10086`, kami coba memetakan seluruh karakter seperti a-Z, 0-9 dan beberapa karakter yang terdapat pada format flag (`{,_,}`) lalu membuat script python seperti berikut,

```
import sys

hachim='a'
hahachim='b'
hahahachim='c'
hahahhachim='d'
hahahhahachim='e'
hahahhahhachim='f'
hahahhahhahachim='g'
hahahhahhahhachim='h'
hahahhahhahhahachim='i'
hahahhahhahhahhachim='j'
hahahhahhahhahhahhachim='k'
hahahhahhahhahhahhahhachim='l'
hahahhahhahhahhahhahhahhachim='m'
hahahhahhahhahhahhahhahhahhachim='n'
hahahhahhahhahhahhahhahhahhahhachim='o'
hahahhahhahhahhahhahhahhahhahhahhachim='p'
hahahhahhahhahhahhahhahhahhahhahhahhachim='q'
hahahhahhahhahhahhahhahhahhahhahhahhahhachim='r'
hahahhahhahhahhahhahhahhahhahhahhahhahhahhachim='s'
hahahhahhahhahhahhahhahhahhahhahhahhahhahhahhachim='t'
hahahhahhahhahhahhahhahhahhahhahhahhahhahhahhahhachim='u'
hahahhahhahhahhahhahhahhahhahhahhahhahhahhahhahhahhachim='v'
hahahhahhahhahhahhahhahhahhahhahhahhahhahhahhahhahhahhachim='w'
hahahhahhahhahhahhahhahhahhahhahhahhahhahhahhahhahhahhahhachim='x'
hahahhahhahhahhahhahhahhahhahhahhahhahhahhahhahhahhahhahhahhachim='y'
hahahhahhahhahhahhahhahhahhahhahhahhahhahhahhahhahhahhahhahhahhachim='z'

haCHIM='A'
hahaCHIM='B'
hahahaCHIM='C'
hahahahaCHIM='D'
hahahahaCHIM='E'
hahahahaCHIM='F'
hahahahaCHIM='G'
hahahahaCHIM='H'
hahahahaCHIM='I'
hahahahaCHIM='J'
hahahahaCHIM='K'
hahahahaCHIM='L'
hahahahaCHIM='M'
hahahahaCHIM='N'
hahahahaCHIM='O'
hahahahaCHIM='P'
hahahahaCHIM='Q'
```

```
hahahahahahahahahahahahahahahahahahahahahahCHIM='R'  
hahahahahahahahahahahahahahahahahahahahahahCHIM='S'  
hahahahahahahahahahahahahahahahahahahahahahCHIM='T'  
hahahahahahahahahahahahahahahahahahahahahahCHIM='U'  
hahahahahahahahahahahahahahahahahahahahahahCHIM='V'  
hahahahahahahahahahahahahahahahahahahahahahCHIM='W'  
hahahahahahahahahahahahahahahahahahahahahahCHIM='X'  
hahahahahahahahahahahahahahahahahahahahahahCHIM='Y'  
hahahahahahahahahahahahahahahahahahahahahahCHIM='Z'
```

```
haduh='0'  
hahaduh='1'  
hahahaduh='2'  
hahahahaduh='3'  
hahahahahaduh='4'  
hahahahahaduh='5'  
hahahahahaduh='6'  
hahahahahaduh='7'  
hahahahahaduh='8'  
hahahahahaduh='9'
```

```
mehbuka='{'  
meh='_ '  
mehtutup='}'
```

```
sys.stdout.write(hahahaCHIM+hahahahahahahahahahahahahahahahahahahahahahCHIM+hahahahahahahahahahahahahah  
aCHIM+hahahahahahahahahahahahahahahahahahahahahahCHIM+hahahahahahahahCHIM+hahahahahahCHIM+hahahahah  
ahahahahahahahahahahahahahahahahahahahahahCHIM+hahahahahahahahahahahahahahahahahahahahahahCHIM+hahaha  
hahahahahahahaduh+mehbuka+hahaCHIM+hahahahahahahahahahahahahahahahahahahahahahhahahahahahahahahahahah  
aCHIM+hahahahaduh+meh+hahaCHIM+hahahahahahahahahahahahahahahahahahahahahahhahahahahahahahahahahahCHI  
M+hahahahaduh+meh+hahahahahahahahCHIM+hahahahahahahahahahahahahahahahahahahahahahhahahahahahahahahah  
ahahahahahahahahahahahahahCHIM+hahahahahahahahahahahahahahahahahahahahCHIM+meh+hahahahahahahahahahah  
ahahahahahahahahahahahCHIM+hahahahahahahahahahahahahahahahahahahahCHIM+hahahahahahahahahahahahahah  
hahahahahahCHIM+hahahahahahahahahahahahahahahahahahahahCHIM+hahahahahahahahahahahahahahahahahahahah  
ahahahahahCHIM+hahahahahahahahahahahahahahahahahahahahCHIM+mehtutup)
```

- Running dan didapatkan flag yaitu : **COMPFEST9{BY3\_BY3\_FROM\_SPROUT}**.

## 2. Can You Get The Plain Text? - 50

- Didapat dua buah file yaitu encrypt.py dan cipher\_text.zip dimana file encrypt.py digunakan untuk mengenkripsi file plaintext menjadi file ciphertext yang di split sejumlah hasil dari panjang plaintext dibagi dengan panjang key, proses enkripsi menggunakan operasi xor antara plaintext dengan key yang biasa disebut dengan One Time Password. Untuk mendapat key kami mencoba melakukan bruteforce operasi xor format flag "COMPFEST9{" dengan flag yang telah terenkripsi seperti script berikut,

```
import sys
flag = []
for i in range(1,13):
    with open("cipher_text_"+str(i), 'r') as f:
        for k in f:
            for l in k:
                flag.append(ord(l))

key = ""
format = "COMPFEST9{"
for j in range(110):
    for i in range(10):
        key += chr(flag[i+j]^ord(format[i]))
    print key
    key = ""
```

- Setelah dirunning kami mencoba mencari kemungkinan key seperti berikut,

```
] I@\\I6X
QK] JJ1rB
G_W\\'uwV
SUA$cpm[
YC
2`fjyQ
O/ve|~tA
-khs~;
inikeynya
e1c;101#(
`v`pNj!
zbmz|Ic`
nogj_""
cew\\    Oe!
Jcp
yW^2;
U
l        _yJ
YKv
V
Ne4r
T
MbY0i
```

- Didapatlah key yang diharapkan yaitu “inikeynya!” dimana karakter tanda seru kami tambahkan manual karna kami sudah tahu pada percobaan sebelumnya, dan kami lanjutkan untuk proses dekripsi menggunakan script berikut,

```
import sys
flag = []
for i in range(1,13):
    with open("cipher_text_"+str(i), 'r') as f:
        for k in f:
            for l in k:
                flag.append(ord(l))

key = "inikeynya!"
gal = "COMPFEST9{"
for j in range(0,120,10):
    for i in range(10):
        sys.stdout.write(chr(flag[i+j]^ord(key[i])))
```

- Running dan didapatlah flag yaitu **COMPFEST9{r3u51n9\_th3\_0n3\_t1m3\_p4d}** seperti gambar berikut,

```
root@fredrica-VirtualBox:/home/fredrica/Downloads# python plaint_text_solver.py
Selamat Anda berhasil mendapatkan flagnya, yaitu COMPFEST9{r3u51n9_th3_0n3_t1m3_p4d}.
Silahkan submit flag yang ada.      root@fredrica-VirtualBox:/home/fredrica/Downloads#
```