

Write Up BeeFestCTF 2018



Tim:

Kosong

Nama : Achmad Zaenuri Dahlan Putra
SMK Telkom Malang

Daftar Isi

Daftar Isi	1
Web	2
tagsnlabels (100 pts)	2
Web	4
wongjowo (200 pts)	4
Web	6
simplexe (400 pts)	6
Reverse Engineering	8
babyre (100 pts)	8
Reverse Engineering	10
hexamethics (200 pts)	10
Reverse Engineering	12
Hello World (200 pts)	12
Reverse Engineering	14
ShuffleShuffle (1000++ pts)	14
Pwn	17
Mistake (100 pts)	17
Pwn	19
simplehash (200 pts)	19
Pwn	21
strlen (400 pts)	21

NB :

Disini saya dapat menyelesaikan seluruh soal yang diberikan. Namun untuk soal reverse shufflesuffle saya berhasil menyelesaikannya sejam setelah kompetisi usai.

Web

tagsnlabels (100 pts)

tagsnlabels

100

Temukan jawaban dalam mars lagu [ini](#) !

Flag

Submit

Diberikan sebuah web yang beralamat sebagai berikut :

<http://ctf-beefest.ga:31335/>

Sesuai dengan deskripsi soal , web tersebut berisi lirik dan video dari mars Bina Nusantara .




Mars Bina Nusantara

Dengarkanlah Negara panggilan Dikau
 Gegap gempita bunyi gederang
 Pahlawan ilmu tingkatkanlah semangatmu
 Untuk Nusa dan Bangsa
 Univ. Bina Nusantara
 Derapkanlah maju terus
 Sebagai wadah Nusa dan Bangsa
 Negara Indonesia
 Memberantas keterbelakangan
 Yang menghambat pembangunan
 Trus berbaktilah dan pantang mundur
 Dengan gigih trus majulah
 Univ. Bina Nusantara
 Bangkitkanlah putra-putrimu
 Dengan semangat dan cita-cita
 Indonesia adil makmur
 Univ. Bina Nusantara



**Trus berbaktilah dan pantang mundur
 Dengan gigih trus majulah**

Thanks to:
binusmaya.binus.ac.id || For Logo
 Yerima Satria Sugiharto || For Video

Karena judul dari soal ini adalah tagsnlabels jadi saya langsung menyimpulkan bahwa terdapat 'sesuatu' pada tags ataupun label pada script html website tersebut.

Dan ternyata benar , terdapat flag pada tag `img`

```
<div class="lgo">  
    
</div>
```

FLAG : BeeFEST{h3h3_I_W4s_h1diNg_Fr0m_u}

Web

wongjowo (200 pts)

wongjowo

200

Andi ingin belanja di online shop. Setelah melakukan pemesanan dan membayar tagihannya, pesanan Andi tetap tidak terverifikasi.

Beliau sudah mencoba menghubungi Customer Service namun tetap tidak mendapatkan jawaban yang pasti.

Karena barang yang Andi beli tidak murah dan membutuhkan barang itu segera, Andi pun memutuskan untuk memaksa masuk ke dalam database sebagai admin dan men-verifikasi pesanan dia secara manual. Saat sampai di halaman admin-login, diketahui bahwa validasi login dilakukan client-side.

Bantu Andi untuk memecahkan masalah tersebut dan Andi akan memberikan kamu flag.

Web yang dikunjungi Andi berada di [sini](#).

Diberikan akses ke sebuah web yang beralamat sebagai berikut :
<http://ctf-beefest.ga:31336/>

Saat pertama kali membuka web terdapat form login dengan sebuah title maupun heading yaitu Script Wong Jowo .

Script Wong Jowo

Username:

Password:

Kemudian saya langsung melakukan view sourcode untuk melihat apakah ada clue atau tidak.

Dan ternyata proses pengecekan login tersebut menggunakan javascript(mungkin yang dimaksud script wong jowo adalah java script tersebut,karena bahasa inggrisnya jawa adalah java)

Berikut script yang digunakan untuk pengecekan login.

```
//Selesaikan teka-teki ini untuk mendapatkan flag
function validate(){
    var username = document.forms["soal"]["username"].value;
    var password = document.forms["soal"]["password"].value;
    if(username == "" || password == ""){
        alert("Semua input harus diisi!");
        return false;
    }
    if(username === "admin" && password ===
String.fromCharCode(52, 107, 85, 95, 98, 51, 108, 52, 106, 97, 114,
95, 106, 52, 118, 52, 83, 99, 114, 49, 112, 116, 95, 49, 48, 49)) {
        alert("Kamu Benar!");
    } else {
        alert("Username atau Password salah!");
        return false;
    }
};
```

Dapat kita lihat bahwa kita harus memasukkan admin sebagai username dan sebuah bentuk decimal dari karakter ascii.

Disini saya menggunakan console untuk melakukan convert ke ascii

```
>> console.log(String.fromCharCode(52, 107, 85, 95, 98, 51, 108, 52, 106, 97, 114, 95, 106, 52, 118, 52, 83, 99, 114, 49, 112, 116,
95, 49, 48, 49))
4kU_b3l4jar_j4v4Scr1pt_101                                     debugger eval code:1:1
```

Kemudian saya lakukan login menggunakan username dan password tersebut dan keluar alert seperti yang tertera pada script.

Kesimpulannya bahwa flag pada soal tersebut adalah password itu sendiri , karena pada saat kita melakukan login dan benar maka hanya akan keluar alert "kamu benar" .

FLAG : BeeFest{4kU_b3l4jar_j4v4Scr1pt_101}

Web

simplexe (400 pts)

simpleexe

400

Jono melakukan ping ke sebuah [server](#) [linux], tetapi Jono menemukan sesuatu yang lain.

Apa yang ditemukan Jono ?

Diberikan sebuah akses ke sebuah web dengan alamat sebagai berikut :
<http://ctf-beefest.ga:31334/>

Web tersebut berfungsi untuk melakukan test ping ke sebuah ip address ataupun website. Kemungkinan web tersebut menggunakan fungsi `shell_exec` pada php untuk memanggil command ping pada linux.

Jadi disini kita dapat melakukan Remote Command Execution pada input tersebut. Caranya cukup menggunakan `;command` sebagai input.

Berikut proses saya menemukan flagnya

input : `;ls`

Dockerfile
flag
index.php

Input : ;cat flag

IP Address	PING TEST
BeeFest{1t_1s_S1mpl3_R3M0T3_C0D3_3x3cut10n}	

FLAG : BeeFest{1t_1s_S1mpl3_R3M0T3_C0D3_3x3cut10n}

Reverse Engineering

babyre (100 pts)

babyre

100

Budi diberikan program magic.

Programnya akan memberikan Budi jawaban (flag) secara otomatis pada waktu tertentu.

Jon anak nakal.

Jon memberikan Budi IDA tools untuk menemukan jawabannya di dalam program.

Apakah anda Jon?

P.S. jawaban dalam format flag (BeeFest[jawaban])

 babyre

Flag

Submit

Diberikan sebuah file binary 32 bit bernama babyre, pertama disini saya coba jalankan file tersebut .

```
root ➤ kosong ➤ bii ➤ # ➤ ./babyre
Sabardong
Halo reverse aku dong
coba cek rahasianya
Sabardong
Halo reverse aku dong
coba cek rahasianya
Sabardong
Halo reverse aku dong
coba cek rahasianya
```

Dan ternyata program tersebut memberikan output yang sama berulang ulang , kemudian saya tidak langsung melakukan disassembly menggunakan ida saya coba terlebih dahulu menggunakan perintah perintah dasar yang sering digunakan untuk reversing.

Disini saya mencoba menggunakan `strings` untuk melihat string printable yang ada pada binary tersebut.

```
root@kosong:~# strings babyre
/lib/ld-linux.so.2
libc.so.6
_IO_stdin_used
puts
printf
getchar
usleep
__libc_start_main
__gmon_start__
GLIBC_2.0
PTRh
UWVS
t$,U
[^_]
Sabardong
Halo reverse aku dong
coba cek rahasianya
BeeFest{basic_reverse_bisa_
gak_liat_sampe_sini_!}
```

Dan ternyata benar terdapat sebuah string menyerupai flag pada binary tersebut, saya mencoba menginputkannya namun salah dan ternyata memang terdapat kesalahan pada pengecekan flag pada webscoring , namun setelah dilakukan pembenaran oleh admin akhirnya saya dapat menginputkan flagnya hehe, terimakasih admin.

FLAG : BeeFest{basic_reverse_bisa_gak_liat_sampe_sini_!}

Reverse Engineering

hexamethics (200 pts)


hexamethics

200

Coba temukan input pada program ini (berupa hexadecimal)

Anda bisa menelusuri kode assembly dari program ini untuk menemukan input yang benar sehingga menghasilkan jawaban (flag).

Gunakan pengetahuan kode assembly operasi aritmatika dengan menggunakan hexadecimal!

 hexamethics

Diberikan sebuah file binary 64 bit dengan nama hexamethics. Pertama disini saya coba untuk menjalankan file tersebut

```
root ➤ kosong bii # ➤ ./hexamethics
Selamat datang di IndoJuni, passwordnya apa?: kosong
Salah :^(
```

Tidak seperti soal sebelumnya, kali ini binary menerima sebuah input, tanpa berlama lama saya langsung membukanya menggunakan ida.

Di ida saya menggunakan fitur yang dapat merubah bahasa assembly menjadi pseudocode, berikut isi dari fungsi main :

```

v7 = *MK_FP(__FS__, 40LL);
v6 = calculate();
printf("Selamat datang di IndoJuni, passwordnya apa?: ", argv);
__isoc99_scanf("%lX", &v5);
fflush(stdin);
if ( v5 == v6 )
    printf("Selamat berbelanja :D\nflag: BeeFest{%lX}\n", v5);
else
    printf("Salah :^(\n)", &v5);
getchar();
result = 0;
v4 = *MK_FP(__FS__, 40LL) ^ v7;
return result;

```

Dapat kita lihat bahwa input yang kita masukkan dibandingkan dengan v6, v6 sendiri memiliki value yaitu sebuah fungsi bernama calculate. Fungsi calculate sendiri berisi sebagai berikut :

```

1 signed __int64 calculate()
2 {
3     return 0xB0DBB8F7LL;
4 }

```

Sebelumnya nilai return terlihat dalam bentuk decimal namun saya convert ke hexadecimal, karena menurut clue yang ada pada deskripsi soal bahwa program tersebut menerima input berupa hexadecimal.

Jadi intinya disini kita disuruh untuk memasukkan nilai yang sama dengan return dari fungsi calculate yaitu 0xB0DBB8F7 . Jalankan file binary dan masukkan bilangan hex tersebut

```

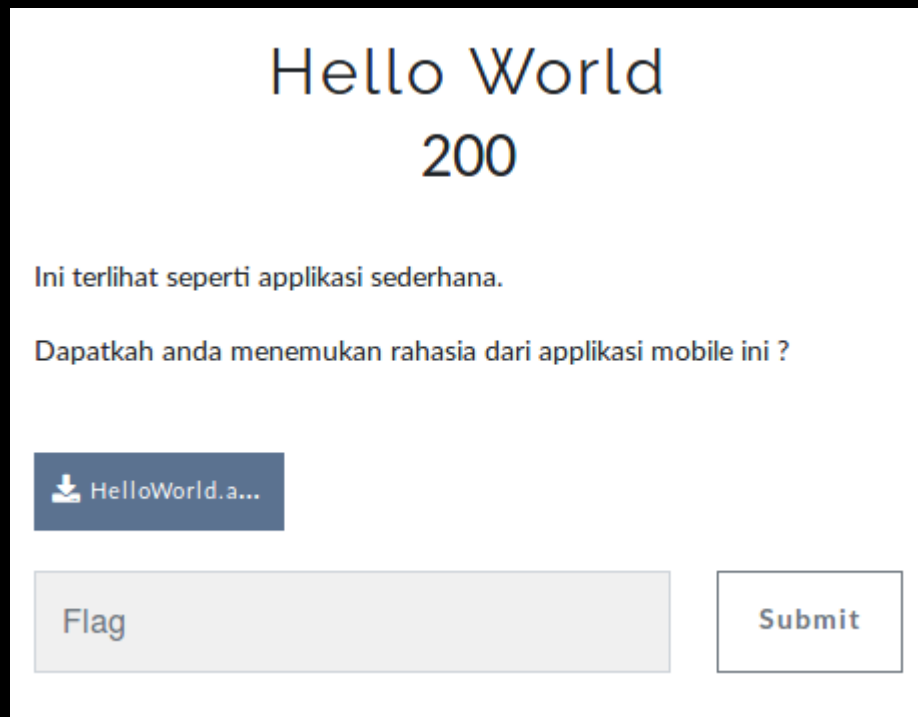
root ➤ kosong bii # ➤ ./hexamethics
Selamat datang di IndoJuni, passwordnya apa?: 0xB0DBB8F7
Selamat berbelanja :D
flag: BeeFest{B0DBB8F7}

```

FLAG : BeeFest{B0DBB8F7}

Reverse Engineering

Hello World (200 pts)



Diberikan sebuah file apk bernama HelloWorld.apk .

Langkah pertama yang saya lakukan adalah melakukan decompile file apk tersebut menggunakan apktool .

```
noob@kosong:~$ apktool d HelloWorld.apk
I: Using Apktool 2.3.3 on HelloWorld.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
S: WARNING: Could not write to (/home/noob/.local/share/apktool/framework), using /tmp instead...
S: Please be aware this is a volatile directory and frameworks could go missing, please utilize --frame-path if the default storage directory is unavailable
I: Loading resource table from file: /tmp/1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
```

Setelah itu karena saya langsung mencoba melihat file utama(main) yang biasa digunakan dalam pemrograman android,yaitu MainActivity .

Disini letak file mainactivity terdapat pada direktori HelloWorld/smali/com/sscape/simpleapk
Kemudian disini saya buka menggunakan sublime .

```
# static fields
.field private static a:Ljava/lang/String; = "QmVlRmVzdHt0MW5nNmFsX2cwMEdsM180akF9DQo="
```

Terdapat sebuah string yang diencode menggunakan base64 , kemudian saya langsung mencoba mendecode nya.

Base64 : QmVlRmVzdHt0MW5nNmFsX2cwMEdsM180akF9DQo=

Command :

echo -n "QmVlRmVzdHt0MW5nNmFsX2cwMEdsM180akF9DQo=" | base64 -d

```
noob > kosong simpleapk $ echo -n "QmVlRmVzdHt0MW5nNmFsX2cwMEdsM180akF9DQo=" | base64 -d
BeeFest{t1ng6a1_g00G13_4jA}
```

FLAG : BeeFest{t1ng6a1_g00G13_4jA}

Reverse Engineering

ShuffleShuffle (1000++ pts)

Diberikan sebuah file binary 32 bit bernama shuffleshuffle.

Hal pertama yang saya lakukan adalah melakukan pengecekan strings pada binary tersebut.

```
root@kosong:~# strings shuffleshuffle
/lib/ld-linux.so.2
libc.so.6
_IO_stdin_used
__isoc99_scanf
puts
__stack_chk_fail
__libc_start_main
__gmon_start__
GLIBC_2.7
GLIBC_2.0
GLIBC_2.4
PTRh
UWVS
t$,U
[^_]
uS_3eNr5r_i_Ne1a_nzgdFv
```

Karena judulnya shuffleshuffle mungkin string `uS_3eNr5r_i_Ne1a_nzgdFv` adalah sebuah flag yang dishuffle.

Setelah itu saya langsung coba membuka file binary tersebut menggunakan ida pro namun sayangnya binary tersebut tidak dapat diconvert menjadi pseudocode(hal ini membuat saya sedikit kesulitan dalam mengetahui alur program tersebut).

Kemudian saya menemukan sesuatu yang menarik yaitu sebuah variable bernama `WAT` dan `WKWK` yang dipanggil sebelum melakukan proses `compare`.

```

loc_8048504:                                ; CODE XREF: main+89↓j
        mov     eax, [ebp+var_28]
        mov     eax, WAT[eax*4]
        movzx   edx, [ebp+eax+var_24]
        mov     ecx, WKWK
        mov     eax, [ebp+var_28]
        add     eax, ecx
        movzx   eax, byte ptr [eax]
        cmp     dl, al
        jz      short loc_804853C
        sub     esp, 0Ch
        push    offset aWrongFlag ; "WRONG FLAG!"
        call    _puts
        add     esp, 10h
        mov     eax, 0

```

Saya coba menelusuri variable tersebut (melihat isi dari variable tersebut pada section .data)

```

.data:00404040 public WAT
.data:00404040 WAT dd 0Eh
.data:00404044 db 5
.data:00404045 db 0
.data:00404046 db 0
.data:00404047 db 0
.data:00404048 db 14h
.data:00404049 db 0
.data:0040404A db 0
.data:0040404B db 0
.data:0040404C db 1
.data:0040404D db 0
.data:0040404E db 0
.data:0040404F db 0
.data:00404050 db 15h

```

Dan ternyata variable WAT memiliki banyak value, untuk mempermudah saya mengerti maksud dari variable WAT saya mengubahnya menjadi bentuk array.

```

public WAT
dd 0Eh, 5, 14h, 1, 15h, 0Fh, 0, 0Bh, 4, 9, 0Ah, 0Ch, 12h
; DATA XREF: main+4C↑r
dd 3, 6, 11h, 10h, 7, 16h, 8, 13h, 0Dh, 2
public WKWK
dd offset aUs_3enr5r_i_ne ; DATA XREF: main+58↑r
ends ; "uS_3eNr5r_i_Ne1a_nzgdFu"

```

Asumsi saya variable WKWK digunakan untuk variable sementara yang menyimpan flag yg dishuffle dengan urutan pada variable WAT .Nantinya variable WKWK akan dibandingkan dengan inputan kita .
Kemudian saya lakukan convert dari hexa ke decimal pada array WAT.


```

public WAT
dd 14, 5, 20, 1, 21, 15, 0, 11, 4, 9, 10, 12, 18, 3, 6
; DATA XREF: main+4C↑r
dd 17, 16, 7, 22, 8, 19, 13, 2
public WKWK
dd offset aUs_3eNr5r_i_ne ; DATA XREF: main+58↑r
ends ; "uS_3eNr5r_i_Ne1a_nzgdFv"

```

Setelah itu saya menyalin array dari WAT dan flag yang dishuffle untuk kemudian membuat script sederhana untuk mengembalikannya. Berikut script yang saya gunakan

```

1 a = [14,5,20,1,21,15,0,11,4,9,10,12,18,3,6,17,16,7,22,8,19,13,2]
2 flag = "uS_3eNr5r_i_Ne1a_nzgdFv"
3 b = list("A" *23)
4 for i in range(len(flag)):
5     b[a[i]] = flag[i]
6 print "".join(b)

```

Inti dari script tersebut adalah saya membuat sebuah array bernama b yang diisi dengan "A" sebanyak 23 kali lalu mengganti masing masing value dari index array yang diambil dari a sesuai dengan value pada Flag .

Output dari program tersebut adalah sebagai berikut.

```

root ➤ kosong bii # ➤ python fin.py
r3verS1ng_i5_FuN_aNd_ez

```

FLAG : BeeFest{r3verS1ng_i5_FuN_aNd_ez}

Pwn

Mistake (100 pts)

Mistake

100

Anggi belajar membuat program hitung.

Ketika dikumpulkan ke gurunya, gurunya memberitahu bahwa ada kesalahan di operasi hitungnya.

Bantu Anggi untuk menemukan kesalahannya dimana, dan Anggi akan memberikan jawaban (flag).

nc ctf-beefest.ga 31331

 mistake

 mistake.c

Flag

Submit

Diberikan sebuah file binary beserta source codenya .
Hal pertama yang saya lakukan adalah mendownload source codenya lalu mempelajari alurnya , berikut isi dari source code tersebut.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int a,b,c;
7      printf("Ikuti aturannya dan menangkan hadiahnya!\n");
8      printf("Masukkan angka pertama : ");
9      scanf("%d",&a);
10     printf("Masukkan angka kedua : ");
11     scanf("%d",&b);
12     if(c=a*b+a-b<=0)
13     {
14         system("cat ./flag ");
15     }
16     else
17     {
18         printf("Kurang beruntung\n");
19     }
20 }

```

Disini kita diminta memasukkan angka sebanyak dua kali, angka pertama disimpan sebagai a dan angka kedua sebagai b.

Ketika hasil dari $a*b+a-b \leq 0$ maka program akan menjalankan command `cat ./flag`.

Jadi disini kita tinggal menentukan angka yang akan menghasilkan nilai negatif atau 0 jika diinputkan pada operasi diatas.

Disini saya menginputkan -1 untuk a dan 2 untuk b.

A = -1

B = 2

C = $a*b+a-b$

C = $-1*2+(-1)-2 = -5$ (dan akan menghasilkan nilai true karena ≤ 0)

```

root ➤ kosong ➤ bii ➤ # ➤ nc ctf-beefest.ga 31331
Ikuti aturannya dan menangkan hadiahnya!
Masukkan angka pertama : -1
Masukkan angka kedua : 2
BeeFest{Th3_risk_1_c4lculat3d_But_I_W4s_B4D_at_M4th}

```

FLAG : BeeFest{Th3_risk_1_c4lculat3d_But_I_W4s_B4D_at_M4th}

Pwn

simplehash (200 pts)


simplehash


200

Calisa menemukan sebuah aplikasi yang meminta password.

Ternyata password tersebut tidak disimpan dalam aplikasi, namun terdapat sebuah hash. Temukan password yang tepat untuk menghasilkan jawaban (flag).

nc ctf-beefest.ga 31332

 simplehash

 simplehash.c

Flag

Submit

Diberikan sebuah binary dan sourcecodenya .

Hal pertama yang saya lakukan adalah mendownload dan mempelajari sourcecode tersebut.

Isi dari source code tersebut sebagai berikut.

```

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  // gcc simplehash.c -o simplehash -m32
6
7  int passHash = 72;
8
9  int checkPass(char *pass){
10     int value = 0, i;
11     for(i = 0; pass[i] != 0; i++){
12         value += pass[i];
13     }
14     value /= strlen(pass);
15     return value;
16 }
17
18 int main(){
19     char buf[16];
20     printf("Masukkan kata sandi: ");
21     fgets(buf, sizeof(buf), stdin);
22     buf[strlen(buf, "\n")] = 0;
23     if(strlen(buf) != 10){
24         printf("Kata sandi harus 10 karakter.\n");
25     } else if(checkPass(buf) == passHash){
26         system("/bin/cat flag.txt");
27     } else{
28         printf("Kata sandi salah.\n");
29     }
30 }
31

```

Terlihat pada source code diatas kita diminta untuk memasukkan sebuah kata sandi dengan panjang minimal 10 karakter.

Jika kita sudah melewati pengecekan untuk 10 karakter tersebut maka kita akan diarahkan ke function checkpass yang dicocokkan dengan variable passHash yang bernilai 72.

Pada function checkPass setiap karakter yang kita inputkan akan ditambahkan ke variable value, yang mana setelah proses for looping variable value akan dibagi dengan panjang inputan kita, yaitu 10. Kurang lebih berikut gambarannya .

$$72 = (\text{char} + \text{char} + \text{char} + \text{char} + \text{char} + \text{char} + \text{char} + \text{char} + \text{char} + \text{char}) / 10$$

$$72 = (10 * \text{char}) / 10$$

$$72 = \text{char}$$

Disini saya cukup menginputkan 'H' sebanyak 10 kali , karena H memiliki nilai decimal 72 .

```

root@kosong bii# nc ctf-beefest.ga 31332
Masukkan kata sandi: HHHHHHHHHH
BeeFest{L00k 4 H4sh C0ll1510n}

```

FLAG : BeeFest{L00k_4_H4sh_C0ll1510n}

Pwn

strlen (400 pts)

strlen

400

Si budi kecil mendapat sebuah program dari tantenya yang seorang 1337 h4ck3r.


Dulu si budi kecil pernah minta diajari cara h4ck1n9 oleh tantenya, namun dengan judesnya si tante hanya menyuruh si budi kecil untuk Google.


Tetapi karena kasihan, si tante akhirnya membuatkan sebuah program sederhana untuk si budi kecil latihan.

Perintahnya sederhana. Si budi kecil hanya perlu memasukkan input melebihi batas yang ditentukan oleh fungsi strlen(). Dapatkah anda membantu si budi kecil untuk mem-bypass fungsinya?

Note: Submit flag seperti yang tertera pada server.

nc ctf-beefest.ga 31333

 strlen

 strlen.c

Flag

Submit

Diberikan sebuah file binary beserta source codenya.

Hal pertama yang saya lakukan adalah mendownload dan mempelajari source code tersebut.

Isi dari source code tersebut sebagai berikut.

```

#include <stdio.h>
#include <string.h>
#include <stdlib.h>

int buggy_check (char *string)
{
    int x = 0;
    while (string [x] != '\n')
    {
        x++;
    }
    return x;
}

int main (void)
{
    char var [64];

    puts ("Kata orang, fungsi strlen() tidak very secure.");
    puts ("Sebagai seorang h4ck3r, anda harus dapat melihat sesuatu dari segala sisi.");
    puts ("Dapatkah anda memasukkan input yang panjangnya melebihi batasan strlen()?");
    puts ("Jika anda bisa memasukkan lebih dari 32 karakter,");
    puts ("saya akan memberikan anda flag-nya :)");
    printf("> ");
    fflush (stdout);
    fgets (var, 64, stdin);
    fflush (stdin);
    if (strlen (var) > 32)
    {
        puts ("Coba lebih kreatif lagi :)");
        exit (0);
    }
    else
    {
        if (buggy_check (var) <= 32)
        {
            puts ("Yah, anda ga niat ya :(");
            exit (0);
        }
        else
        {
            puts ("Nah, bisa kan :D");
            system ("/bin/cat flag");
            exit (0);
        }
    }
}

```

Disini kita diminta untuk menginput karakter yang kurang atau sama dengan 32 panjangnya jika dicek menggunakan strlen dan besar dari 32 jika dicek menggunakan function buggy_check .

Function buggy_check menghasil nilai x yang mana x akan bertambah sesuai panjang dari input yang kita masukkan kecuali jika x bernilai \n .

Sebelumnya saya menambakan beberapa baris kode untuk memudahkan saya dalam menemukan payload yang tepat untuk soal tersebut.

```

fflush (stdin);
...printf("%s\n",var);
...printf("%d\n",strlen(var));
...printf("%d",buggy_check(var));
if (strlen (var) > 32)
{
    puts ("Coba lebih kreatif lagi :)");
}

```

```

root@kali:~# ./str
Kata orang, fungsi strlen() tidak very secure.
Sebagai seorang h4ck3r, anda harus dapat melihat sesuatu dari segala sisi.
Dapatkah anda memasukkan input yang panjangnya melebihi batasan strlen()?
Jika anda bisa memasukkan lebih dari 32 karakter,
saya akan memberikan anda flag-nya :)
> asd
asd

4
3Yah, anda ga niat ya :(

```

Terlihat saat saya memasukkan asd maka akan tampil nilai dari inputan saya, strlen dari inputan saya, dan juga nilai dari buggy_check. Setelah saya melakukan banyak percobaan akhirnya saya menemukan payload yang tepat, yaitu kita hanya perlu memasukkan string not printable sebagai inputan. Berikut payload yang saya gunakan.

```
python -c "'\n'*2" | nc ctf-beefest.ga 31333
```

Output dari python -c "'\n'" akan menghasilkan nilai random character yang akan diinputkan pada server(karena bukan print '\n'*2 yang menghasilkan \n sebanyak 2 kali).

Berikut proof nya.

```

> root@kali:~# python -c "'\n'*2" | nc ctf-beefest.ga 31333
Kata orang, fungsi strlen() tidak very secure.
Sebagai seorang h4ck3r, anda harus dapat melihat sesuatu dari segala sisi.
Dapatkah anda memasukkan input yang panjangnya melebihi batasan strlen()?
Jika anda bisa memasukkan lebih dari 32 karakter,
saya akan memberikan anda flag-nya :)
> BEEFEST{n0t_so_S3cur3_n0w_ar3Nt_u_strl33n}
Nah, bisa kan :D

```

FLAG : BEEFEST{n0t_so_S3cur3_n0w_ar3Nt_u_strl33n}

TERIMA KASIH