



Nama Tim : Tenesys

Universitas : Universitas Teknokrat Indonesia

## 1. morning-glory-pool (Stegano - 50)

- Strings pada file gambar.

```
BBAAAAABAAABABBABABBABBBABABBABAABABAA
BBABBBAAABBABAABAABAABA AAAABABAAAAABBA
AAAABBAABABBABABAABBAABBBAABAABAAAB
```

- Ditemukan sebuah Baconian Cipher.
- Decode dan mendapatkan hasil "YELLOWSTONES BIG BROTHER".
- Clue tersebut merupakan julukan untuk Danau Toba.



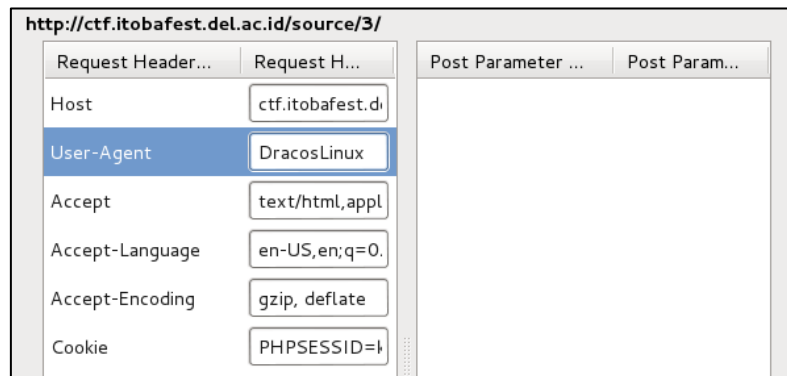
- Mencoba search kordinat danau toba dimana clue pada soal yaitu "LETAK".
- Didapatlah kordinat pada website [https://id.wikipedia.org/wiki/Danau\\_Toba](https://id.wikipedia.org/wiki/Danau_Toba) yang merupakan flag pada misi tersebut namun tidak menggunakan karakter simbol.

Informasi	
Lokasi	Sumatera Utara
Negara	 Indonesia
Koordinat	 3,58°LU 98,67°BT
Jenis objek wisata	Wisata alam danau

- Flag adalah **ITF{358LU9867BT}**.

## 2. Web 3 (Website - 75)

- Didapat clue yang diminta yaitu merubah User Agent pada website soal tersebut menjadi DracosLinux dan apapun yang ada pada tampilan web tersebut hanya tipuan semata.
- Mencoba melakukan request User Agent menggunakan plugin Tamper Data pada browser Mozilla Firefox.



- Didapatlah flag setelah melakukan request.

File Flag : dracosflag.txt

your browser is **DracosLinux**  
Flag anda : U5eRA6entMakeY0uCo0l

- Flag adalah **ITF{U5eRA6entMakeY0uCo0l}**.

## 3. Rev 4 (Reversing - 25)

- File soal berupa file ELF dengan nama "easy".
- Buka file tersebut dengan aplikasi IDA 32bit.
- Pada fungsi **sub\_804872B**, program akan mencoba membuka file **.txt**, kemudian membaca 20 karakter.

```

int __cdecl sub_804872B(int a1, int a2)
{
    FILE *stream; // [sp+8h] [bp-10h]@1
    char v4; // [sp+fh] [bp-9h]@1

    v4 = 0;
    snprintf(ptr, 0x100u, "%s.txt", *(_DWORD *)a2);
    stream = fopen(ptr, "r");
    if ( stream )
    {
        memset(ptr, 0, 0x100u);
        fread(ptr, 1u, 0x14u, stream);
        fclose(stream);
        v4 = sub_804854B((int)ptr);
    }
    if ( v4 != 1 )
        puts("Masih salah, silakan coba lagi...");
    return 0;
}

```

- Pada fungsi **sub\_804854B**, program akan memvalidasi 20 karakter apakah sama dengan strings "4m47\_v1c70r14\_cur4m!".

```

int __cdecl sub_804854B(int a1)
{
    char v2; // [sp+0h] [bp-0h]@1
    signed int i; // [sp+Ch] [bp-Ch]@21

    v2 = 0;
    if ( *(_BYTE *)(a1 + 19) == '!'
        && *(_BYTE *)(a1 + 5) == 118
        && *(_BYTE *)(a1 + 9) == 48
        && *(_BYTE *)(a1 + 15) == 117
        && *(_BYTE *)(a1 + 1) == 109 || *(_BYTE *)(a1 + 18) == 109)
        && *(_BYTE *)(a1 + 3) == 55 || *(_BYTE *)(a1 + 8) == 55)
        && *(_BYTE *)(a1 + 4) == 95 || *(_BYTE *)(a1 + 13) == 95)
        && *(_BYTE *)(a1 + 6) == 49 || *(_BYTE *)(a1 + 11) == 49)
        && *(_BYTE *)(a1 + 7) == 99 || *(_BYTE *)(a1 + 14) == 99)
        && *(_BYTE *)(a1 + 10) == 114 || *(_BYTE *)(a1 + 16) == 114)
        && *(_BYTE *)a1 == 52 || *(_BYTE *)(a1 + 2) == 52 || *(_BYTE *)(a1 + 12) == 52 || *(_BYTE *)(a1 + 17) == 52 )
    {
        return 1;
    }
    return 0;
}

```

- Kami coba membuat file "easy.txt" dengan isi strings "4m47\_v1c70r14\_cur4m!".
- Jalankan program kembali dan didapatlah flag yaitu **ITobaFest{marsipature\_hutana\_be}**.

#### 4. Rev 1 (Reversing - 75)

- File soal berupa file ELF dengan nama rev1.
- Buka dengan IDA 64bit.
- Pada fungsi main, terdapat fungsi validasi hasil input dengan variable "s".

```
int v6; // [sp+Ch] [bp-34h]@1
char v7[40]; // [sp+10h] [bp-30h]@1
__int64 v8; // [sp+38h] [bp-8h]@1

v8 = *MK_FP(__FS__, 40LL);
printf("Flag: ", argv, envp);
__isoc99_scanf(4196251LL, v7);
v6 = 0;
for ( i = 0; i <= 30; ++i )
{
    if ( v7[i] == s[30 - i] )
        ++v6;
}
if ( v6 == 31 )
    puts("Correct!");
else
    puts("Wrong!");
result = 0;
v4 = *MK_FP(__FS__, 40LL) ^ v8;
return result;
```

- Coba kita lihat isi variable "s", ternyata berupa sejumlah huruf.

00000000000000000000000000000000	; int s[]	
00000000000000000000000000000000	s	dd 70h
00000000000000000000000000000000		db 79h ; y
00000000000000000000000000000000		db 0
00000000000000000000000000000000		db 0
00000000000000000000000000000000		db 0
00000000000000000000000000000000		db 73h ; s
00000000000000000000000000000000		db 0
00000000000000000000000000000000		db 0
00000000000000000000000000000000		db 0
00000000000000000000000000000000		db 61h ; a
00000000000000000000000000000000		db 0
00000000000000000000000000000000		db 0
00000000000000000000000000000000		db 0
00000000000000000000000000000000		db 65h ; e
00000000000000000000000000000000		db 0
00000000000000000000000000000000		db 0
00000000000000000000000000000000		db 0
00000000000000000000000000000000		db 5Fh ; _
00000000000000000000000000000000		db 0
00000000000000000000000000000000		db 0
00000000000000000000000000000000		db 73h ; s

- Kita coba reverse string tersebut dan ternyata flagnya adalah ITobaFest{easy\_reverse\_is\_easy}.

#### 5. Rev 3 (Reversing - 100)

- File soal berupa file ELF dengan nama rev3.
- Buka dengan IDA 64bit.
- Pada fungsi **sub\_400677** terdapat fungsi validasi hasil input dengan variable di dalam program seperti berikut,

```

v14 = *HK_FP(__FS__, 40LL);
printf("Flag: ");
__isoc99_scanf(4196651LL, v13);
v12 = 0;
for ( i = 0; i < dword_601114; ++i )
{
    v8 = v12++;
    if ( dword_601060[i] != v13[v8] )
    {
        result = sub_400640();
        goto LABEL_27;
    }
}
for ( j = 0; j < dword_601118; ++j )
{
    v2 = v12++;
    if ( dword_601080[j] != (v13[v2] ^ dword_601060[j]) )
    {
        result = sub_400640();
        goto LABEL_27;
    }
}

```

```

for ( k = 0; k < dword_60111C; ++k )
{
    v3 = v12++;
    if ( dword_6010A0[k] != (v13[v3] ^ dword_601080[k]) )
    {
        result = sub_400640();
        goto LABEL_27;
    }
}
for ( l = 0; l < dword_601120; ++l )
{
    v4 = v12++;
    if ( dword_6010C0[l] != (v13[v4] ^ dword_601080[l]) )
    {
        result = sub_400640();
        goto LABEL_27;
    }
}

```

```

for ( m = 0; m < dword_601124; ++m )
{
    v5 = v12++;
    if ( dword_6010E0[m] != (v13[v5] ^ dword_601080[m]) )
    {
        result = sub_400640();
        goto LABEL_27;
    }
}
result = sub_400662();
LABEL_27:
v6 = *HK_FP(__FS__, 40LL) ^ v14;
return result;
}

```

- Kita coba lakukan pengecekan pada variable tersebut, pada perulangan pertama hasil inputan dari karakter **1-7** akan valid, apabila sama dengan isi dari variable **dword\_601060**.
- Pada perulangan kedua, hasil inputan **8-14** valid apabila di **xor** dengan **dword\_601060** hasilnya sama dengan isi dari variable **dword\_601080**.
- Pada perulangan ketiga, hasil inputan **15-19** valid apabila di **xor** dengan **dword\_601080** hasilnya sama dengan isi dari variable **dword\_6010A0**.
- Pada perulangan keempat, hasil inputan **20-25** valid apabila di **xor** dengan **dword\_601080** hasilnya sama dengan isi dari variable **dword\_6010C0**.
- Pada perulangan terakhir, hasil inputan **26-30** valid apabila di **xor** dengan **dword\_601100** hasilnya sama dengan isi dari variable **dword\_6010E0**.
- Didapatlah flag yaitu **ITobaFest{ok\_this\_is\_not\_hard}**.

## 6. Web 1 (Website - 25)

- Pada bagian source code, terdapat baris yang mencurigakan yaitu,

```
19 <!--admin:&lt;t;~1bpas1M/UU0P34T3+4g%2)d&lt;t;N1,h%%@:_H.@q@PQA25u$3FX^0~&gt;;  
20 -->
```

- Kita coba decode html strings mencurigakan tersebut, dan didapat hasil berikut,

### Decode Result

```
<~1bpas1M/UU0P34T3+4g%2)d<N1,h%%@:_H.@q@PQA25u$3FX^0~>
```

- Diketahui bahwa hasil decode tersebut merupakan sebuah base64 dengan tipe a85, dan lakukan decode kembali menggunakan python 3 lalu didapatlah hasil berupa hash SHA1.

```
>  
base64.a85decode("1bpas1M/UU0P34T3+4g%2)d<N1,h%%@:_H.@q@PQA25u$3FX^0")  
=> b'420f3f8b0f6f8a915738274fae9bce61d2489b1a'
```

- Setelah mencari Plain Text dari hash tersebut, didapatlah sebuah strings dengan text berupa “kambing” sebagai password.
- Coba login dengan Username “admin” dan Password “kambing”,

### Can you get a username and password ?

Username:

Password:

Submit

**bener banget**

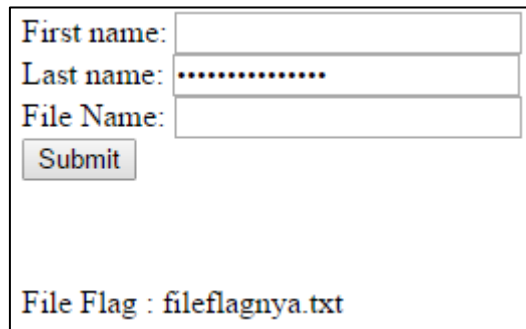
Your Flag :

**B0Nu5Nya25P0inTAjaYah**

- Didapatlah flag yaitu **ITF{B0Nu5Nya25P0inTAjaYah}**.

## 7. Web 2 (Website - 50)

- Terdapat 3 form pada web seperti berikut,



First name:

Last name:

File Name:

File Flag : fileflagnya.txt

- Kami coba untuk menginputkan strings "fileflagnya.txt" pada form "Last Name" karena pada form tersebut semua inputan diencode menjadi base64.
- Didapatlah hasil "ZmlsZWZsYWdueWEudHh0".
- Input kembali Base64 tersebut pada form "File Name" seperti berikut dan mendapatkan flag,



First name:

Last name:

File Name:

File Flag : fileflagnya.txt

Selamat Datang Yang Di Pertuan dan punya password  
Dengan ini saya nyatakan bahwa file yang anda cari yang bernama ZmlsZWZsYWdueWEudHh0 .....

**Selamat !!!**

Flag Anda : Base64153asYR1GhT

- Flag adalah **ITF{Base64153asYR1GhT}**.