2019 제17회 순천향대학교 정보보호 페스티벌(YISF)

예선 풀이

이름 : 성민규

학교 : 인헌고등학교

아이디 : mandu9280

닉네임 : M4ndU

자기 점수 : 600

등수 : 10등

Reversing 50

문제 이름

기밀 문서

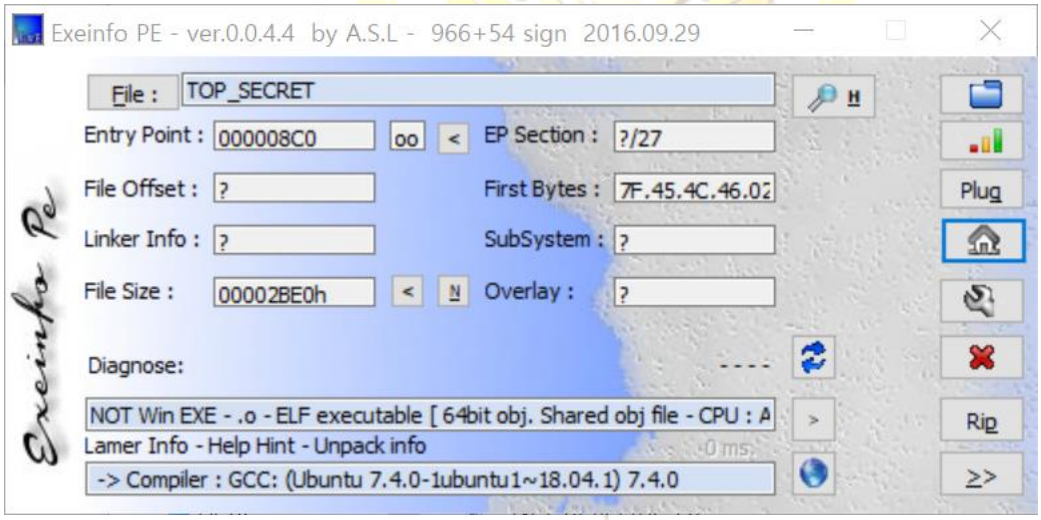
문제 설명

이 문서에는 엄청난 것이 들어있는게 분명하다. 무엇이 들어있는지 확인해볼까?

TOP_SECRET

Hint1 : 특정 폴더와 파일 이름?

Hint2 : 비밀번호 변조



ELF 64bit 파일입니다.

IDA로 분석을 하였습니다.

```
__int64 __fastcall main(__int64 a1, char **a2, char **a3)
{
  char *v3; // ST20_8@1
  signed __int64 v4; // rdi@2
  __int64 result; // rax@5
  __int64 v6; // rcx@5
  char *v7; // [sp+18h] [bp-A8h]@1
  char buf; // [sp+30h] [bp-90h]@1
  __int64 v9; // [sp+B8h] [bp-8h]@1

  v9 = *MK_FP(__FS__, 40LL);
  memset(&buf, 0, 0x80uLL);
  v7 = strrchr(*a2, 47) + 1;
  v3 = getcwd(&buf, 0x80uLL);
  printf("path : %s\nfilename : %s\n\n", v3, v7, a2);
  sub_B2F(v3, v7);
  if ( dword_20240C == 1 )
  {
    v4 = (signed __int64)(v7 + 6);
    if ( !strncmp(v7 + 6, "flag", 4uLL) )
    {
      ++dword_20240C;
      sub_DE7(v4, "flag");
    }
    sub_D64(v4, "flag");
  }
  result = 0LL;
  v6 = *MK_FP(__FS__, 40LL) ^ v9;
  return result;
}
```

v3 경로랑 v7 파일명을 인자로 sub_B2F함수를 호출합니다.

```
42  if ( !strncmp((const char *)(a1 + 6), "YISF", 4uLL) )
43  {
44    puts("\nHmm...?\n");
45    if ( !strncmp((const char *)(a1 + 11), "TOP_SECRET", 0xAuLL) )
46    {
47      puts("Please enter your ID and Password...\n");
48      printf("ID : ", "TOP_SECRET", a2);
49      fgets(s, 64, stdin);
50      printf("PW : ", 64LL);
51      fgets(v13, 64, stdin);
52      strcpy(dest, v13);
53      if ( strncmp(s, aThe_world_best, 0x19uLL) || strncmp(v13, aQwe123, 6uLL) )
54      {
55        puts("\nYou don't have permission!!\n");
56        exit(0);
57      }
58      puts("\nsuccess!!\n");
59      ++dword_20240C;
60    }
61    else
62    {
63      puts("Invalid Directory Name\n");
64    }
65  }
66  else
67  {
68    puts("Ivalid Directory Name\n");
69  }
70  result = dest;
71  v3 = *MK_FP(__FS__, 40LL) ^ v21;
72  return result;
73}
```

sub_B2F함수를 보면, 실행경로의 7번째 문자부터는 YISF가 있어야 하고, 12번째 문자부터는 TOP_SECRET가 있어야 합니다. 따라서 TOP_SECRET파일을 /home/YISF/TOP_SECRET/ 경로로 이동시켰습니다.

```
.data:0000000000202010                  align 20h
.data:0000000000202020 ; char aThe_world_best[]
.data:0000000000202020 aThe_world_best db 'The_World_Best_Programmer',0
.data:0000000000202020                                          ; DATA XREF: sub_B2F+19C↑o
.data:000000000020203A                  db    0
.data:000000000020203B                  db    0
```

id = The_World_Best_Programmer

```
.data:000000000020205F                  db    0
.data:0000000000202060 ; char aQwe123[]
.data:0000000000202060 aQwe123          db 'qwe123',0           ; DATA XREF: sub_B2F+1B8↑o
.data:0000000000202067                  db    0
.data:0000000000202068                  db    0
.data:0000000000202069                  db    0
```

pw = qwe123

```
mandu@mandu-VirtualBox:/home/YISF/TOP_SECRET$ ./TOP_SECRET
path : /home/YISF/TOP_SECRET
filename : TOP_SECRET


Hmm...?

Please enter your ID and Password...

ID : The_World_Best_Programmer
PW : qwe123

success!!

YISF{Hin! Ar3 Y0u D3ce1ved??}
```

하지만, 출력된 플래그는 인증이 되지 않았습니다.

```
19    v4 = (signed __int64)(v7 + 6);
20    if ( !strncmp(v7 + 6, "flag", 4uLL) )
21    {
22      ++dword_20240C;
23      sub_DE7(v4, "flag");
24    }
25    sub_D64(v4, "flag");
26  }
```

다시 메인함수를 보면, 파일명의 7번째 문자부터 flag가 있는 경우와 없는 경우에 따라 플래그를 출력해주는 함수가 다른 것을 알 수 있었습니다.

```
mandu@mandu-VirtualBox:/home/YISF/TOP_SECRET$ ./123456flag
path : /home/YISF/TOP_SECRET
filename : 123456flag


Hmm...?

Please enter your ID and Password...

ID : The_World_Best_Programmer
PW : qwe123

success!!

        YISF{5252~~_I_6eliev3d!!!}
```

그래서 파일명을 123456flag로 변경하고 다시 실행하였습니다.

이 결과, 올바른 플래그를 얻을 수 있었습니다.

Forensic 50

문제 이름

범죄를 증명하라(1)

문제 설명

마약관련 범죄를 저지른 범죄조직을 감시하던중 네트워크를 통해 정보를 주고받았다는 제보를 받았다. 패킷을 수집하였으나 분석을 할 수 있는 사람이 없어 분석을 못하고 있다. 수사기관을 도와 분석을 마무리하자.

Link1

Link2

Link3

압축 비밀번호 :wjswoddmstlwkrehldjTek.sjdhkskdmlRmxdmfqhwkRnsk....ejaqufk!djelRkwlvnftndlTsmswlgksqjswlzuqhrpTek.

<제보1> 익명의 제보자는 조직원들이 FTP를 이용하여 파일을 공유했다라고 한다.

<제보2> 익명의 제보자는 recovery라는 메시지를 남긴 채 연락이 두절되었다.....

<제보3> 연락 두절된 제보자가 image.zip을 복구하라는 메세지를 보냈다!

Wireshark로 분석하였습니다.

ftp-data 패킷을 분석해서 4개의 파일을 카빙할 수 있었습니다.

(Image.zip, 이미지 파일 2개, 멜론 설치파일.)

이중 image.zip파일이 오류로 열리지 않습니다. 이 image.zip파일을 복구하여 파일을 읽어야 했습니다.

HxD로 열어보면 파일이 한 개 존재함을 알 수 있는데, 데이터 부분만 추출하여 zip파일을 만든 뒤에 아래 코드로 압축을 풀 수 있었습니다.

image.zip

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	50	4B	03	04	14	00	00	08	02	00	08	00	AB	A9	DE	4E	PK.....«©ÞN
00000010	85	60	E0	A9	1F	00	00	00	1D	00	00	00	00	00	00	00	…`à©.....
00000020	00	00	00	00	00	00	00	00	8B	F4	0C	76	AB	8E	34	08	........‹ô.v«Ž4.
00000030	8D	37	09	32	8E	77	37	30	70	89	F7	30	F0	35	8E	F7	.7.2Žw70p‰÷0ð5Ž÷
00000040	35	F1	36	0E	AA	05	00	50	4B	01	02	14	00	14	00	00	5ñ6.ª..PK.....
00000050	08	08	00	AB	A9	DE	4E	85	60	E0	A9	1F	00	00	00	1D	...«©ÞN…`à©.....
00000060	00	00	00	00	00	24	00	00	00	00	00	00	00	20	00	00	.....$..... ..
00000070	00	00	00	00	00	00	00	00	00	00	00	00	00	0A	00	20	.....
00000080	00	00	00	00	00	01	00	18	00	01	9D	EF	34	3D	2F	D5	.....ï4=/Õ
00000090	01	B1	97	6A	04	3C	2F	D5	01	B1	97	6A	04	3C	2F	D5	.±—j.</Õ.±—j.</Õ
000000A0	01	50	4B	05	06	00	00	00	00	01	00	01	00	5A	00	00	.PK.....Z..
000000B0	00	45	00	00	00	00	00										.E.....

카빙한 image.zip파일의 데이터부분

image.zip

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	8B	F4	0C	76	AB	8E	34	08	8D	37	09	32	8E	77	37	30	‹ô.v«Ž4..7.2Žw70
00000010	70	89	F7	30	F0	35	8E	F7	35	F1	36	0E	AA	05	00		p‰÷0ð5Ž÷5ñ6.ª..

추출한 데이터만 담긴 image.zip

```
import zlib

d = open('image.zip').read()
```

print zlib.decompress(d, -15)

#YISF{Y0U_4R3_G00D_H0M3_M4K3R}

Misc 50

문제 이름

룰 확인

문제 설명

룰을 확인하세요!

대회의 룰을 읽고 하단의 '확인을 클릭하면' 플래그가 나왔습니다.

FLAG : YISF{G00D_LUCK_3V3RY01V3}

Misc 100

문제 이름

Hidden area search

문제 설명

nc **218.158.141.199** **24763**

매 문제마다 새로운 직선방정식 3개가 주어진다.
이 직선방정식들로 만들어진 삼각형의 넓이를 구하여라

일차방정식 3개를 입력받아, 각각 두 방정식끼리의 교점을 구하고, 세 점의 좌표를 알 때의 삼각형 넓이 공식에 대입하여 넓이를 구하였습니다.

Python3 solve.py

```
from sympy import *
from pwn import *


def cleaneqn(s):
    s = str(s)
    s= list(map(str, s.split()))
    s[0] = s[0].replace("b'", "")
    s[6] = s[6].replace("₩₩n'","")
    if int(s[3])<0:
        ss = s[0]+"*"+s[1]+s[3]+"*"+s[4]+" "+s[6]
    else:
        ss = s[0]+"*"+s[1]+s[2]+s[3]+"*"+s[4]+" "+s[6]
    return ss


def solveeqn(a1, a2):
```

```
    x, y = symbols('x y')
    a1 = a1.split()
    a2 = a2.split()
    return eval("solve( [ Eq("+a1[0]+" ,"+a1[1]+"), Eq("+a2[0]+", "+a2[1]+") ], [x,y] )")

def solve_func(e):
    x, y = symbols('x y')

    for i in range(0,3):
        e[i] = cleaneqn(e[i]) #입력받은 방정식을 sympy가 입력받을 수 있는 형태로 변환

    ss1 = solveeqn(e[0], e[1]) #교점 좌표 구하기
    ss2 = solveeqn(e[1], e[2])
    ss3 = solveeqn(e[0], e[2])
    x1 = ss1[x]
    y1 = ss1[y]
    x2 = ss2[x]
    y2 = ss2[y]
    x3 = ss3[x]
    y3 = ss3[y]

    #각 꼭짓점의 좌표를 알 때의 삼각형 넓이 공식
    ans = 0.5*abs((x1 - x2)*y3 + (x2-x3)*y1 + (x3-x1)*y2)
```

```
    return ans


#main
equation = ["0"]*3 #방정식을 저장하기 위함
p = remote("218.158.141.199", 24763)
p.recvuntil("Start>")
p.recvline()
p.recvline()
for i in range(0, 100):
    print(p.recvline()) #step
    p.recvline()
    equation[0] = p.recvline()
    equation[1] = p.recvline()
    equation[2] = p.recvline()
    for j in range(0, 3):
        print(equation[j])
    p.recvuntil(":")
    p.sendline(str(solve_func(equation)))
    print(p.recvline()) #correct
    p.recvline()


p.interactive()
```

#flag : YISF{Mathematical_ability_i5_n0t_ru5ty}

Misc 150

문제 이름

Rule_reverse_engineering

문제 설명

[MISC-150]Rule_reverse_engineering

nc 218.158.141.182 52387

실행마다 예시인 문자열이 달라지는데 바이너리값은 같을 때가 있다. 같은 것을 보면 문자열에서 바이너리로 변하는 일정한 규칙이 있는 것 같다. 규칙이 적용된 문자열이 주어진 바이너리와 같도록 문자열을 입력해라

주어진 테이블에 맞추어서 치환해주면 됐습니다.

치환시 중요했던 점은, 치환해야할 문자열의 길이가 긴 것 부터 치환을 했어야 했고,

치환한 문자가 0이나 1일 경우 원래 문자와 겹치므로 다른 범위의 문자로 바꾼 뒤에 다시 0이나 1로 돌려주어야 했습니다.

Python3 solve.py

```
from pwn import *
```

```
import ast


def ccccc(h):
    h = h.replace('b"table :    ', "")
    h = h.replace('₩₩n"', "")
    print(h)
    inn = ast.literal_eval(h) #type dictionary
    return inn


def ddd(e):
    e = e.replace("b'", "")
    e = e.replace("₩₩n'", "")
    return e


def ans(innn, task):
    innn = ccccc(innn)
    ss = sorted(innn, key=lambda k : innn[k]) #치환 대상 문자열의 길이가 긴 순서대로 치환
    task = ddd(task)
    for j in ss:
        p=j
        if j == "1": #치환 대상의 문자열과 동일한 0 과 1을 범위 밖의 문자로 치환하여 나중에 다시 되돌림
            p = "#"
```

```
            if j == "0":
                p = "@"
            task = task.replace(innn[j], p)
        task = task.replace("#", "1")
        task = task.replace("@", "0")
        print(task)
        return task


#main
p = remote("218.158.141.182", 52387)
p.recvuntil("Step :  1")
p.recvline()
p.recvline()
for i in range(0, 99):
    try:
        t = p.recvline() #task
        p.recvline()
        table = p.recvline() #table
        p.recvuntil(": ")
        p.sendline(ans(str(table), str(t)))
        print(p.recvline())
        p.recvline()
        p.recvline()
```

```
            p.recvline()
            print(p.recvline())
            print(p.recvline())
        except:
            p.interactive()
t = p.recvline() #Stage 100
p.recvline()
table = p.recvline()
p.recvuntil(": ")
p.sendline(ans(str(table), str(t)))
p.interactive()


#flag : YISF{Y0u_make_table_WeLL}
```

Misc 200

문제 이름

Find First!

문제 설명

nc 218.158.141.142 9238

처음 시작 위치를 찾아라!

[설명]

1. 모든 버튼은 전부 한번 씩 눌려야 하고 항상 마지막으로는 F가 눌립니다.

2. 배열의 시작은 왼쪽 위((x, y)=(0, 0))이고 x와 y의 값은 0과 양의 정수 입니다.

3. 버튼의 처음 시작 위치를 문제당 1초 안에 찾으십시오.

4. 배열의 행과 열의 크기는 5 이상, 10 이하 입니다.

5.문자는 다음 버튼의 위치를 나타내고 숫자는 이동 칸 수를 나타냅니다.

5-1) D = Down, U = Up, R = Right, L = Left

5-2) Ex) D5 = 아래로 5칸, R3 = 오른쪽으로 3칸

6. 모든 스테이지를 클리어 하면 플래그가 주어집니다.

Python3 solve.py

```
from pwn import *
import re

def ddd(e):
    e = e.replace("b'", "")
    e = e.replace("₩₩n'", "")
    return e

def cntarr(col): #count element in first line / range of x
    col = ddd(col)
    col = col.replace("₩₩t", "")
```

```
    col = col.replace("F", "FF")
    col = col.replace("₩₩r", "")
    cnt = int(len(col) *0.5)
    return cnt


def ans(b, c, d):

    for o in range(0, d):
        a[o] = ddd(str(a[o]))
        a[o] = a[o].replace("₩₩t", "")
        a[o] = a[o].replace("₩₩r", "")
        a[o] = a[o].replace("F", "FF")
        aa = a[o]
        print(a[o])
        a[o] = re.findall(r'..',aa)
    print(a)
    now_x = 0 #F부터 시작할 필요가 없음. (0,0)지점부터 시작
    now_y = 0
    overcnt = 1
    while(overcnt > 0): #시작위치를 찾아내면 루프를 벗어남
        #현재 위치가 시작점이 아니라면, 이전 위치를 찾아야함.
        #만약 이전 위치가 왼쪽 방향이라면, 왼쪽 방향 n거리 만큼에 Rn이 존재함.
        #아래쪽 방향이라면, 아래쪽 방향 n거리 만큼에 Un이 존재함
```

```
    for k in range(0, 10): #MAX 10. scan start
        n = str(k)
        if now_x+k < c: #배열 범위를 벗어나면 오류가 나기 때문에, 범위를 지정해주어야함
            if now_y >= 0 and now_x >=0 and a[now_y][now_x+k] == ("L"+n): #right
                now_x += k #찾은 경우, 현재 위치를 찾은 위치로 이동함
                break
        if now_y+k < d:
            if now_y >= 0 and now_x >=0 and a[now_y+k][now_x] == ("U"+n): #down
                now_y += k
                break
        if now_x-k >=0:
            if now_y >= 0 and now_x >=0 and a[now_y][now_x-k] == ("R"+n): #left
                now_x -= k
                break
        if now_y-k >=0:
            if now_y >= 0 and now_x >=0 and a[now_y-k][now_x] == ("D"+n): #up
                now_y -= k
                break
        if k == 9:
            overcnt = 0 #모든 방향에 존재하지 않는다면 현재 위치가 시작지점임

ax = now_x
ay = now_y
```

```
    print(ax, ay)
    return ax, ay



p = remote("218.158.141.142", 9238)
p.recvuntil("Problem 1₩n")
p.recvuntil("Problem 1₩n")
for m in range(0,99):
    a= ["0"]*10
    a[0] = p.recvline()
    print(a)
    count = cntarr(str(a[0]))
    c2=1
    for i in range(1, 10):
        a[i] = p.recvline()
        c2 +=1
        if len(str(a[i])) < 10:
            c2-=1
            break
    p.recvuntil(": ")
    w, z = ans(a, count, c2)
    p.sendline(str(w)+" "+str(z))
```

```
        print(p.recvline())
        p.recvline()
        print(p.recvline())
#stage 100
a= ["0"]*10
a[0] = p.recvline()
print(a)
count = cntarr(str(a[0]))
c2=1
for i in range(1, 10):
        a[i] = p.recvline()
        c2 +=1
        if len(str(a[i])) < 10:
                c2-=1
                break
p.recvuntil(": ")
w, z = ans(a, count, c2)
p.sendline(str(w)+" "+str(z))
p.interactive()

#flag : YISF{Y0(_)_4r3_4_w0nd3rf(_)l_pr0gr4mm3r!!}
```

수고 많으셨습니다.

풀이보고서는 yisf.sch@gmail.com 으로 보내주시면 됩니다.