2012 NewHeart WhiteHat Hacker Contest.

이호준(jnvb) 5th

Problem 1.

```
a = 0x20a9 + int('1100111111',2)
b = int('10011000',2) ^ int('00110101',2)
newheart = 'NewHeart'
c = 0
for i in range(0, len(newheart)):
        c += ord(newheart[i])
print str(a)+str(b)+str(c)
```

Key : 9192173798

Problem 2.

핸드폰으로 http://1.221.63.146:10007/lv2/ 이주소로 접속하는 문제이므로 User-Agent 만 수정하여 curl 로 요청을 보내니 답이 나왔다.

curl -A "Mozilla/5.0 (iPhone; U; CPU iPhone OS 2_0 likeMac OS X; ja-jp) AppleWebKit/525.18.1 (KHTML, like Gecko) Version/3.1.1Mobile/5A345 Safari/525.20" http://1.221.63.146:10007/lv2/

<h6>Admin Page</h6><!--password is Well begun is half done.-->

Key : Well begun is half done.

Problem3.

BMP 파일을 보니 색깔이 들어가있는 픽셀들이 있었다. 포토샵 칼라픽커로 하나씩 뽑아보니 ASCII 값이 나왔다. 근데 배열이 마음대로 되있었다. 이리저리 조합해보다 답을 찾았다.
Key : newhe@rt!!

Problem4.

Base85 가 적힌 txt 가 있고 사진들이있었다.

thisiskey.txt 의 줄 갯수가 7 줄 사진도 7 개 1 줄당 글자쑤 49 개 그림도 49 칸.
그래서 그림과 대조하여 흑색부분만 남기고 다 지웠다.
<~E+*g/GAho$d&ORgBOu!rDdR0d@r#drB4#7^F*),>@;I)1~>

Key : hello_hacking_festival!!

Problem5.

안드로이드 앱이 주어졌다. 디컴파일후 jd-gui 로 분석해보니

```
public void df23089ikgdf()
{
        String str = decript("ygbahi?+hih5vrhhsb1r");
        Toast.makeText(this, str, 1).show();
}
```

이부분이 수상하여 decript 함수를 파이썬으로 다시 짜봤다.

```
arrayOfInt = ['0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0','0']

arrayOfInt[0]='1'
arrayOfInt[2]='34'
arrayOfInt[3]='24'
arrayOfInt[4]='17'
arrayOfInt[5]='11'
arrayOfInt[6]='18'
arrayOfInt[7]='21'
arrayOfInt[8]='6'
arrayOfInt[9]='26'
arrayOfInt[10]='16'
arrayOfInt[11]='7'
arrayOfInt[12]='43'
arrayOfInt[13]='2'
arrayOfInt[14]='8'
arrayOfInt[15]='3'
arrayOfInt[16]='36'
arrayOfInt[17]='22'
arrayOfInt[18]='10'
arrayOfInt[19]='28'
arrayOfInt[20]='9'
arrayOfInt[21]='13'
arrayOfInt[22]='14'
arrayOfInt[23]='20'
arrayOfInt[24]='5'
arrayOfInt[25]='41'
arrayOfInt[26]='25'
arrayOfInt[27]='37'
arrayOfInt[28]='29'
arrayOfInt[29]='31'
arrayOfInt[30]='44'
arrayOfInt[31]='30'
arrayOfInt[32]='35'
arrayOfInt[33]='23'
arrayOfInt[34]='38'
arrayOfInt[35]='40'
arrayOfInt[36]='33'
arrayOfInt[37]='4'
arrayOfInt[38]='27'
arrayOfInt[39]='19'
arrayOfInt[40]='15'
arrayOfInt[41]='12'
arrayOfInt[42]='42'
arrayOfInt[43]='32'
arrayOfInt[44]='39'

tmp = ''
passwd = ''
str1 =  "ygbahi?+hih5vrhhsb1r"
table = "abcdefghijklmnopqrstuvwxyz/.1234567890~_:%+=?"

for i in range(0, 45):
    tmp += table[int(arrayOfInt[i])]

#print tmp

for i in range(0, len(str1)):
    for k in range(0, len(table)):
        if str1[i] == tmp[k]:
            passwd += table[k]

print passwd
```

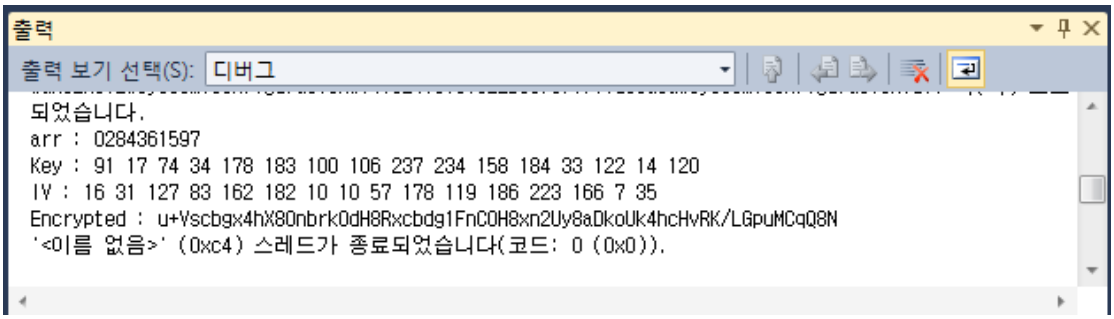
Key : diablo3+lol=hellgate

Problem6.

Xap 파일이므로 압축을 풀어보니 윈도우 모바일 앱이 나왔다.
Nhf3.dll 을 reflector 로 열어 분석을 해봤지만 사용한 함수도 AES Encrypt 하는 부분이라 수상했다. 한참을 고민하다 같이 압축 되있던 WMAppManifest.xml 를 열어보니 Description 부분에 9 자리 숫자가 있었다.

<App xmlns="" ProductID="{afa6e73c-e753-43cd-9470-b34ce1941e59}" Title="NewHeart" RuntimeType="Silverlight" Version="1.0.0.0" Genre="apps.normal" Author="newheart author" Description="134628957" Publisher="newheart">

134628957을 쳐보니 0284361597의 salt값을 가지고 암호화된 값이 나왔다.



Key : u+Vscbgx4hX8Onbrk0dH8Rxcbdg1FnCOH8xn2Uy8aDkoUk4hcHvRK/LGpuMCqQ8N

Problem7.

30분에 한번씩 롤렛을 돌릴 수 있다. 시간에 관계없이 돌릴려면 timestamp값을 수정 해주면 됐다. 이걸 기반으로 스크립트를 짯다. 크롬 개발자 도구로 스크립트를 실행시켰다.

```
for(i=500;i<=600; i++)
{
httpRequest2_2=getXMLHttpRequest();
httpRequest2_2.open("POST", "./process.php", true);
httpRequest2_2.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
var origin2="timestamp=";
origin2+="13"+i+"30064672";
origin2+="-point=3";
var query2=encodeURIComponent(Aes.Ctr.encrypt(origin2, "Busker Busker", 256));
httpRequest2_2.send("data="+query2);
}
```

다음에 힌트2를 보니 sql injection 을 하라는거 같았다.
스크립트를 새로 짜서 돌렸다.

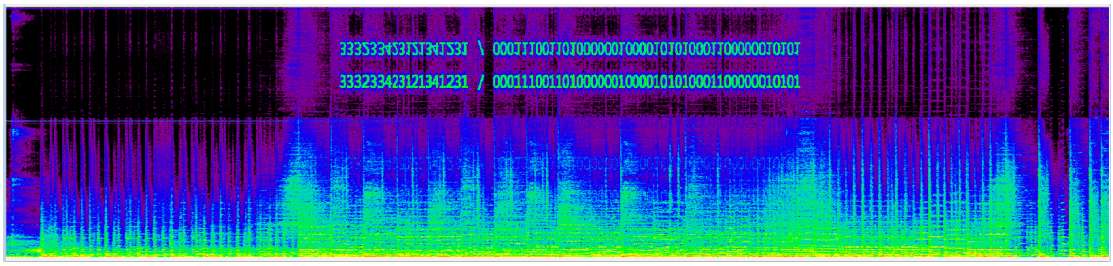
```
httpRequest2_2=getXMLHttpRequest();
httpRequest2_2.open("POST", "./process.php", true);
httpRequest2_2.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
var origin2="timestamp=1000000000000, point=23258#";
origin2+="-point=";
origin2+=3;
var query2=encodeURIComponent(Aes.Ctr.encrypt(origin2, "Busker Busker", 256));
httpRequest2_2.send("data="+query2);
```

이제 부자가 되었으니 1000원짜리를 살수있다.

Key : newheartbeat

Problem8.

Wav파일이 주어졌다. 습관적으로 Goldwave로 스펙트럼을 확인해 봤다.



숫자가 나왔다..

앞에 333233423121341231을 각각 더하니 44였고 뒤에 000111001101~~의 길이는 44자리였다.

그래서 앞에 숫자로 각각 자리를 나눠봤더니

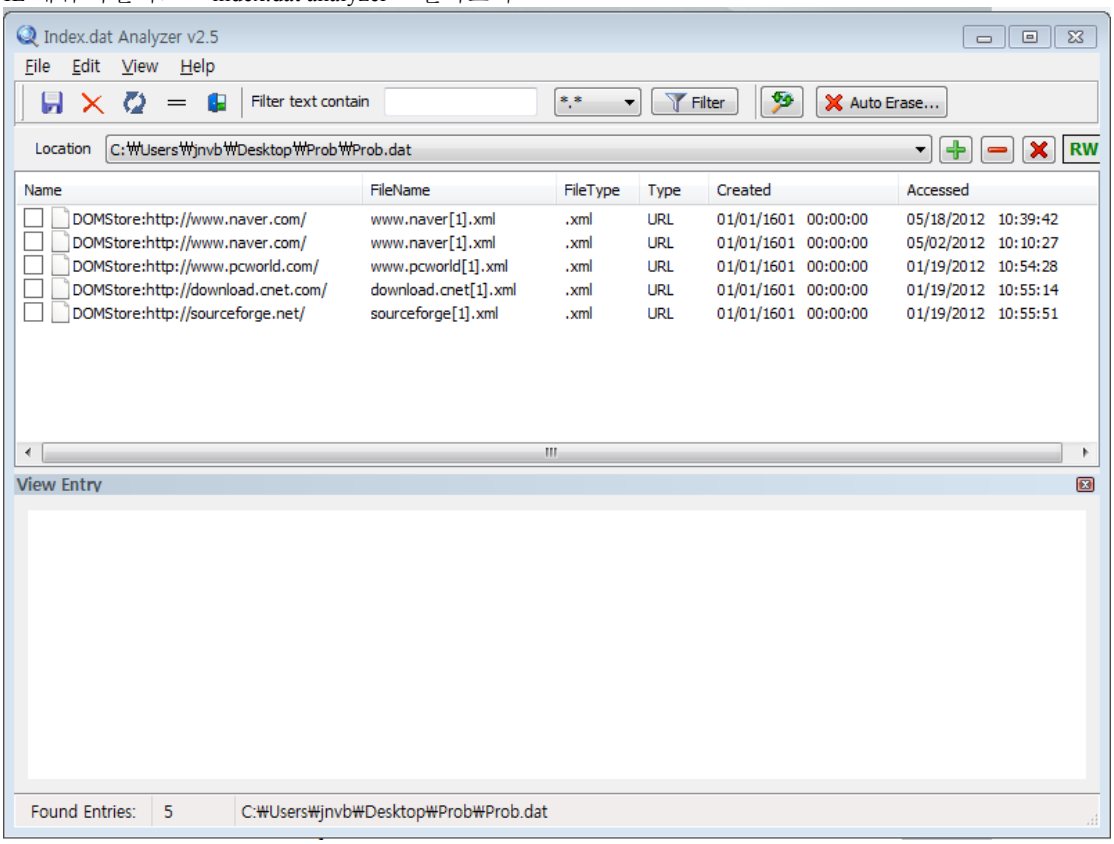
000 111 001 10 100 000 0100 00 101 0 10 0 011 0000 0 01 010 1

모스부호 같이 생겨서 0은 짧은부분 1은 긴부분으로 해석해서 변환해봤더니 답이였다.

Key : soundslikenewheart

Problem9.

IE 캐쉬 파일이므로 index.dat analyzer로 열어보니



URL에 접근한 날짜와 시간이 나온다. 5개니 하나씩 인증해봤다.

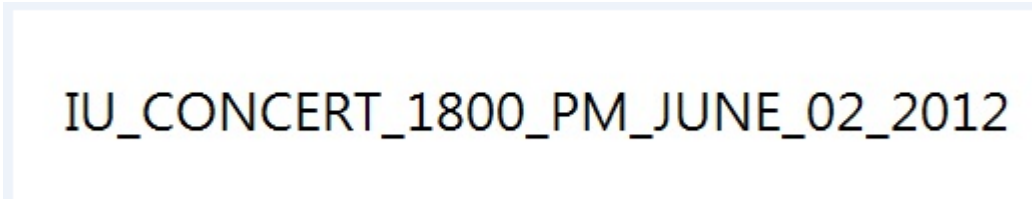
Key : 20120119105514

Problem10.

Android폰 파일들이 주어졌다. 마약거래범의 접선장소를 구하는게 문제였다.
Browser의 캐쉬를 일일이 보다가 나중에 연락처 부분을 보던중 수상한 JPEG포맷의 파일을 발견했다.

PART_13372276170

확장자를 JPG로 바꺼서 열어보니 답이 나왔다.



Key : IU_CONCERT_1800_PM_JUNE_02_2012

Problem 11.

윈도우 드라이버 파일이 주어졌다. 처음에는 드라이버를 로드해서 풀어볼려고 했지만 잘 안되어 그냥 수동으로 분석했다.

Sub_11540부분에서 파일을 만들고 거기에 답을 쓰는것을 확인할수 있었다.

```
int __stdcall sub_11540(int a1, int a2, int a3, int a4, int a5, int a6)
{
  STRING DestinationString; // [sp+Ch] [bp-164h]@2
  char v8; // [sp+18h] [bp-158h]@3
  LARGE_INTEGER ByteOffset; // [sp+140h] [bp-30h]@3
  HANDLE Handle; // [sp+14Ch] [bp-24h]@3
  OBJECT_ATTRIBUTES ObjectAttributes; // [sp+150h] [bp-20h]@3
  struct _IO_STATUS_BLOCK IoStatusBlock; // [sp+168h] [bp-8h]@3

  if ( a2 == 65664 )
  {
    RtlUnicodeStringToAnsiString(&DestinationString, *(PCUNICODE_STRING *)(a3 + 8), 1u);
    if ( !stricmp(DestinationString.Buffer, "WW??WWC:WWNewHeartWWMemo.txt") )
    {
      RtlInitUnicodeString((PUNICODE_STRING)&v8, L"WWDosDevicesWWC:WWNewHeartWWMemo.txt");
      ObjectAttributes.Length = 24;
      ObjectAttributes.RootDirectory = 0;
      ObjectAttributes.Attributes = 64;
      ObjectAttributes.ObjectName = (PUNICODE_STRING)&v8;
      ObjectAttributes.SecurityDescriptor = 0;
      ObjectAttributes.SecurityQualityOfService = 0;
      ZwCreateFile(&Handle, 0xC0100000u, &ObjectAttributes, &IoStatusBlock, 0, 0x80u, 3u, 3u, 0x60u, 0, 0);
      sub_11180(&unk_13008);
      ZwWriteFile(Handle, 0, 0, 0, &IoStatusBlock, &byte_130E0, 0x18u, &ByteOffset, 0);
      ZwClose(Handle);
      dword_130C8 = (PKTIMER)ExAllocatePool(0, 0x28u);
      P = ExAllocatePool(0, 0x20u);
      KeInitializeTimer(dword_130C8);
      KeInitializeDpc((PRKDPC)P, DeferredRoutine, 0);
      KeSetTimerEx(dword_130C8, (LARGE_INTEGER)-10i64, 2000, (PKDPC)P);
    }
  }
  return dword_130CC(a1, a2, a3, a4, a5, a6);
}
```

sub_11180(&unk_13008) 에서 unk_13008와 byte_130A0를 xor해서 답을 만드는 루틴이 보였다. 하지만 byte_130A0의 값을 모른다. byte_130A0의 값은 sub_11010함수에서 만들어지고 있었다.

```
void *__stdcall sub_11010(int a1)
{
  void *result; // eax@1
  bool v2; // [sp+0h] [bp-40h]@12
  signed int v3; // [sp+4h] [bp-3Ch]@7
  char v4; // [sp+8h] [bp-38h]@1
  char v5; // [sp+9h] [bp-37h]@1
  ULONG_PTR v6; // [sp+30h] [bp-10h]@1
  int i; // [sp+34h] [bp-Ch]@1
  double v8; // [sp+38h] [bp-8h]@1
  int v9; // [sp+40h] [bp+0h]@1

  v6 = (unsigned int)&v9 ^ BugCheckParameter2;
  v8 = dbl_13000;
  v4 = 0;
  result = memset(&v5, 0, 0x27u);
  for ( i = 0; i < 10000; ++i )
    v8 = 4.0 * v8 * (1.0 - v8);
  for ( i = 0; i < 24; ++i )
  {
    result = (void *)i;
    *(&v4 + i) = 0;
  }
  v3 = -1;
  for ( i = 0; i < 8 * a1; ++i )
  {
    if ( !(i % 8) )
      ++v3;
    *(&v4 + v3) *= 2;
    v2 = v8 > 0.5;
    result = (void *)(v2 | *(&v4 + v3));
    *(&v4 + v3) = (char)result;
    v8 = 4.0 * v8 * (1.0 - v8);
  }
  for ( i = 0; i < 40; ++i )
  {
    result = (void *)i;
    byte_130A0[i] = *(&v4 + i);
  }
  return result;
}
```

Sub_11010함수를 보니 dbl_13000이 있었다. newheart.sys파일을 헥스 에디터로 열어서 0x1000부터 8바이트를 읽어와서 더블형으로 나타냈다.

```
void main ()
{
    char asdf[8] = {0x38, 0x2E, 0xE3, 0xA6, 0x06, 0x9A, 0xC9, 0x3F};
    double* pnt = (double*)asdf;
    printf("%lf", *pnt);
}
```

값은 0.200013 이 나왔다.

그걸 토대로 키값을 만드는 sub_11010함수를 짜봤다.

```
void main ()
{
    double v8 = 0.200013;
    char v4[1024];
    bool v2;
    unsigned char key[40];


    for(int i = 0; i < 10000; ++i)
    {
        v8 = 4.0*v8*(1.0-v8);
    }
    for(int i = 0; i < 24; ++i)
    {
        v4[i] = 0;
    }
    signed int v3 = -1;
    for(int i = 0; i < 8 * 39 ; ++i)
    {
        if(!(i%8))
            ++v3;

        v4[v3] *= 2;
        v2 = v8 > 0.5;
        char result = (char)v2|v4[v3];
        v4[v3] = result;
        v8 = 4.0*v8*(1.0-v8);
    }
    for(int i = 0; i < 40; ++i)
    {
        key[i] = v4[i];
    }

    //printf("%s\n", key);

    for(int i = 0; i < 40; i++)
    {
        printf("%x ", key[i]);
    }

}
```

결과는

82 65 52 38 4f 92 67 ce 9e 77 a7 4d c3 88 90 e6 71 e7 eb 1b 7e d7 9e ff 9b 57 29 e0 e4 f9 92 1d f2 d2 26 dd d3 70 a0 cc

XOR키값을 구했으니 이제 unk_13008과 XOR를 해봤다.

```
keytable = ['82','65','52','38','4f','92','67','ce','9e','77','a7','4d','c3','88','90','e6','71','e7','eb','1b','7e','d7','9e','ff','9b','57','29','e0','e4','f9','92','1d','f2','d2','26','dd','d3','70','a0','cc']

blabla = ['D2','04','21','48','38','FD','15','AA','BE','3E','D4','6D','93','E7','F3','87','03','8E','B4','48','29','92','DF','AB','CD','FD','FD','FD','FD','DD','DD','DD','F7','37','7F','2F','1C','FF']

plain = ''

for i in range(0, 40):
        print chr(int(keytable[i],16)^int(blabla[i],16)),
```

Key : Pocari_SWEAT