제1회 청소년 화이트해커 경진대회

Kwon Hyuk (pwn3r)

Level1

```
1. 20A9(16진수)와 1100111111(2진수)의 합을 10진수로 나타내시오
2. 10011000과 00110101의 xor 연산을 하고 10진수로 나타내시오
3. N e w H e a r t
각각의 문자하나를 ascii 코드값의 10진수 합으로 나타내면?

1, 2, 3번 키를 붙여서 인증
```

간단한 연산문제들이다.
python을 이용해 쉽게 계산할 수 있다.

(1)

```
>>> 0x20A9 + 0b1100111111
9192
```

(2)

```
>>> 0b10011000 ^ 0b00110101
173
```

(3)

```
>>> str = "NewHeart"
>>> sum = 0
>>> for i in str:
...     sum += ord(i)
...
>>> sum
798
```

세 값을 이어붙인 값이 인증키이다.

Flag : 9192173798

Level2

```
http://1.221.63.146:10007/lv2/
```

주어진 URL에 접속해보면 핸드폰으로 접속하라는 문구만이 나온다.

```
pwn3r@localhost:~/ctf/newheart/level11$ curl http://1.221.63.146:10007/lv2/
핸드폰으로 접속하세요
```

User-Agent를 검사하여 핸드폰인지 PC인지 구별한다고 생각할 수 있다.
아이폰의 기본 User-Agent로 바꾸어 접속시켜본다.

```
pwn3r@localhost:~/ctf/newheart/level11$ curl --user-agent "Mozilla/5.0
(iPhone; U; CPU like Mac OS X; en) AppleWebKit/420+ (KHTML, like Gecko)
Version/3.0 Mobile/1A543a Safari/419.3" http://1.221.63.146:10007/lv2/
<h6>Admin Page</h6><!--password is Well begun is half done.-->
```

"Admin Page"라는 문구와 함께 주석으로 인증키가 나왔다.

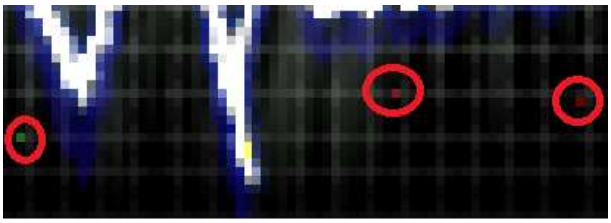
Flag : Well begun is half done

Level3

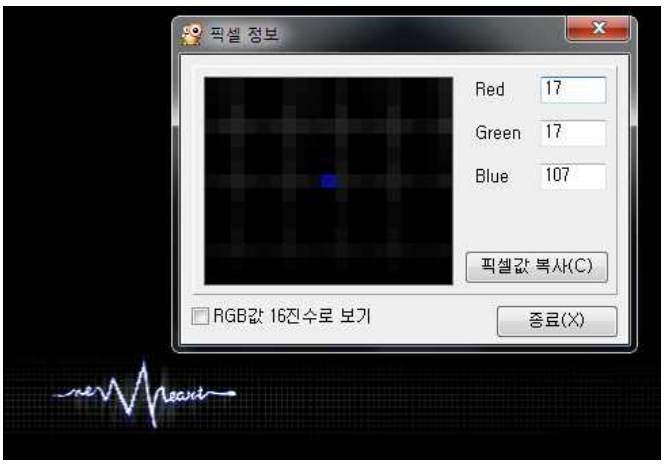
http://1.221.63.146:10007/lv3/nh_header.bmp
누군가 뉴하트 홈페이지에 있는 로고에 비밀번호를 숨겨놓았다.

<주어진 그림파일>

주어진 그림파일을 확대해 보면 배경과 색깔이 전혀 다른 빨강 , 초록 , 파란색점이 군데군데 찍혀있는 것을 볼 수 있다.
(동그라미친부분)



이미지 뷰어 프로그램 Alsee의 픽셀정보보기 기능을 이용해 중간중간에 찍힌 점들의 RGB정보를 확인해보니 한가지 색깔의 수가 높고 나머지 두수는 같았다.



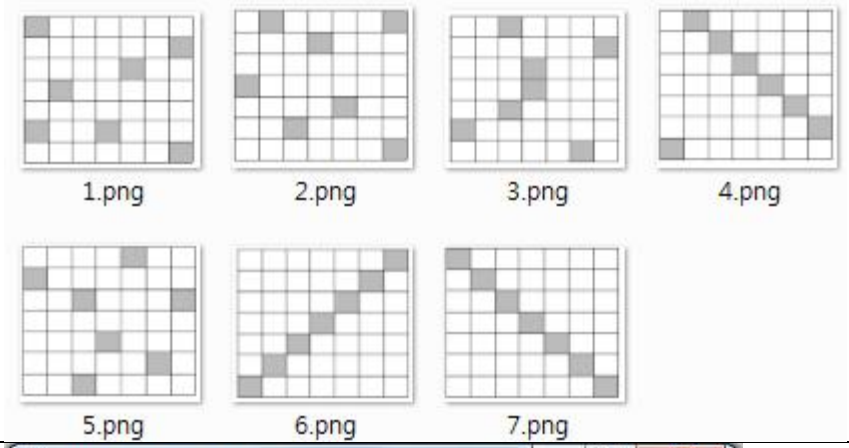
똑같이 눈에 띄는 점들의 위부터 순서대로 RGB정보를 수작업으로 확인해 제일 높은 각 픽셀에서 가장 높은 값을 모아 아스키문자로 변환해 이어 붙여보니 인증키가 나타났다.

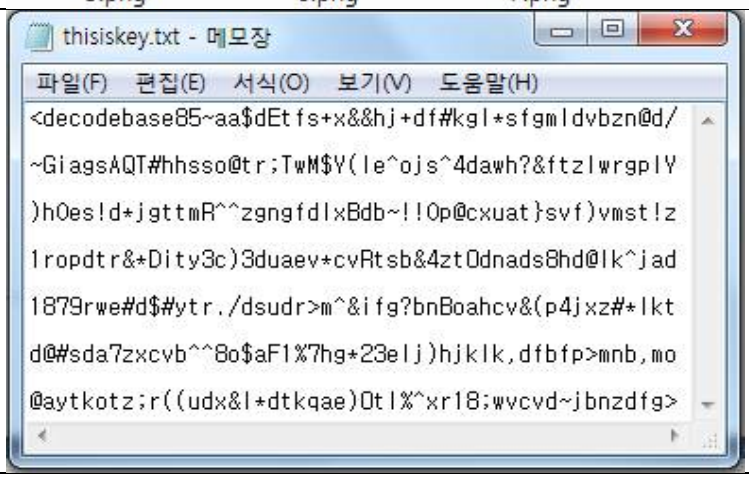
Flag : newhe@rt!

Level4

http://1.221.63.146:10007/lv4/problem.zip

주어진 zip 파일의 압축들 풀어보면 7 개의 png 파일과 thisiskey.txt 라는 텍스트파일이 있다.





그림파일과 텍스트 파일에서 두가지 특이한 점을 찾을 수 있다.

1) 7 개의 그림 , 각 그림은 7x7 의 네모로 이루어짐.
2) 7 개의 문장 , 각 문장은 49 글자

한 png 파일에 있는 작은 네모의 수와 한 문장에 있는 글자의 수가 동일하므로 , 문장마다 각 png 에서 색칠된 부분만 모아 , 이어붙이면 base85 암호문이 나올거라고 추측해보았다. (제일 앞에 decodebase85 라고 되있으므로)

```
<~E+*g/
GAhM4?Y
ORgBOu!
rDdR0d@
r#drB4#
```

```
7^F*),>
@;I)1~>
```

이를 이어붙이면 정상적인 base85 암호문이 만들어진다.

```
<~E+*g/GAhM4?YORgBOu!rDdR0d@r#drB4#7^F*),>@;I)1~>
```

Base85 decode 해보니 인증키가 나왔다.

```
password_is_hello_hacking_festival!!
```

Flag : hello_hacking_festival!!

Level5

http://1.221.63.146:10007/lv5/h4ck.apk

Apk 파일하나가 주어졌다.
Zip 압축파일과 포맷이 동일하므로 zip으로 열어 압축을 풀고 , classes.dex를 .jar로 변환 후 GUI 자바 디컴파일러인 jad-gui로 열어본다.

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

수상한 문자열을 decript라는 함수에 인자로 넘겨주고 , 결과값을 화면에 보여주는 함수가 있다.
인증키를 보여주는 과정이라고 의심해볼 수 있다.

decript함수를 C 프로그램으로 작성해 실행해보았다.

```
pwn3r@localhost:~/ctf/newheart/level5$ cat gen.c
#include <stdio.h>

int main(int argc ,char **argv)
{
    int a;
    char table[] = "abcdefghijklmnopqrstuvwxyz/.1234567890~_:%+=?";
    int arrayOfInt[45];
    arrayOfInt[0] = 1;
    arrayOfInt[1] = 0;
    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;
    int i , j , k ;
    j = strlen(argv[1]);
    for(i=0; i<j; i++)
    {
        for(k=0;k<45;k++)
        {
                if(argv[1][i] == table[k])
                {
                        for(a=0;a<45;a++)
                        {
                                if(arrayOfInt[a] == k) printf("%c" , table[a]);
                        }
                }
        }
    }
    printf("\n");
}
pwn3r@localhost:~/ctf/newheart/level5$ ./gen ygbahi?+hih5vrhhsb1r
diablo3+lol=hellgate
```

결과로 겁나 키값같이 생긴 문자열이 나타났다.
인증을 했더니 성공했다.

Flag : diablo3+lol=hellgate

Level6

http://1.221.63.146:10007/lv6/nhf3.xap

주어진 파일은 윈도우폰 어플리케이션이다.
윈도우폰 에뮬레이터를 설치해 어플리케이션을 실행시켜보니 9개의 숫자버튼과 Submit버튼 , Cancel버튼이 나온다. 최대 9자리까지 각 숫자를 한번씩 입력할 수 있다.

<에뮬레이터 실행화면>

숫자입력 후 submit버튼을 누르면 위에 알 수 없는 문자열이 나타난다.

zip파일과 포맷이 동일하므로 zip으로 압축을 풀고 , 내장된 nhf.dll을 닷넷 디플렉터로 열어분석해 본다.
아래는 submit버튼을 누르면 실행되는부분이다.,

```
private void Summit_Click(object sender, RoutedEventArgs e)
{
    Rfc2898DeriveBytes rfc2898DeriveByte = new Rfc2898DeriveBytes("a5b8f53248a781e32e4fac5190dbfabc", this.arr);
    Aes aesManaged = new AesManaged();
    aesManaged.Key = rfc2898DeriveByte.GetBytes(aesManaged.KeySize / 8);
    aesManaged.IV = rfc2898DeriveByte.GetBytes(aesManaged.BlockSize / 8);
    using (MemoryStream memoryStream = new MemoryStream())
    {
        ICryptoTransform cryptoTransform = aesManaged.CreateEncryptor();
        using (CryptoStream cryptoStream = new CryptoStream(memoryStream, cryptoTransform, CryptoStreamMode.Write))
        {
            byte[] bytes = Encoding.UTF8.GetBytes("e069836af6c41b560477d80ce8b08a36");
            cryptoStream.Write(bytes, 0, (int)bytes.Length);
            cryptoStream.FlushFinalBlock();
        }
        this.PageTitle.Text = Convert.ToBase64String(memoryStream.ToArray());
    }
}
```

위 함수는 "a5b8f53248a781e32e4fac5190dbfabc"를 key로 , 입력한 숫자를 iv와 버튼의 숫자들을 iv로 하여 "e069836af6c41b560477d80ce8b08a36"를 AES-128로 암호화한다.

submit버튼을 누르면 에뮬레이터화면에 문자열이 잘려서 나타난다.
그래서 정상적으로 결과문자열을 복사할 수 있도록 위 코드와 같은 동작을 하도록 C# 프로그램을 만들고 , xap파일에 포함된 WMAppManifest.xml에 있는 9자리의 숫자(아래사진에서 134628957)가 어플리케이션에서 입력됐을때 생성되는 암호를 구해 인증을 시도했더니 성공했다.

```
Description="134628957" Publisher="newheart">
```

(위 숫자순서로 어플리케이션에서 입력하면 변수엔 284361597로 들어감)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Security.Cryptography;
using System.IO;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            byte[] arr = new byte[10];
            byte a = 3;
```

```
                byte b = 5;
                byte c = 6;
                byte d = 4;
                byte e = 8;
                byte f = 1;
                arr[0] = 0;
                arr[1] = 2;
                arr[2] = 8;
                arr[3] = 4;
                arr[4] = 3;
                arr[5] = 6;
                arr[6] = 1;
                arr[7] = 5;
                arr[8] = 9;
                arr[9] = 7;

                Rfc2898DeriveBytes rfc2898DeriveByte = new Rfc2898DeriveBytes("a5b8f53248a781e32e4fac5190dbfabc", arr);
                Aes aesManaged = new AesManaged();
                aesManaged.Key = rfc2898DeriveByte.GetBytes(aesManaged.KeySize / 8);
                aesManaged.IV = rfc2898DeriveByte.GetBytes(aesManaged.BlockSize / 8);
                using (MemoryStream memoryStream = new MemoryStream())
                {
                    ICryptoTransform cryptoTransform = aesManaged.CreateEncryptor();
                    using (CryptoStream cryptoStream = new CryptoStream(memoryStream, cryptoTransform, CryptoStreamMode.Write))
                    {
                        byte[] bytes = Encoding.UTF8.GetBytes("e069836af6c41b560477d80ce8b08a36");
                        cryptoStream.Write(bytes, 0, (int)bytes.Length);
                        cryptoStream.FlushFinalBlock();
                    }
                    Console.WriteLine(Convert.ToBase64String(memoryStream.ToArray()));
                    Console.ReadLine();
                }
            }
        }
}
```

Flag : u+Vscbgx4hX8Onbrk0dH8Rxcbdg1FnCOH8xn2Uy8aDkoUk4hcHvRK/LGpuMCqQ8N

Level7

http://1.221.63.146:10007/quiz/

대회 중 가장 많은 시간을 보내게했던 문제이다.
주어진 URL에 접속하면 자바스크립트로 구현된 룰렛게임을 진행한다.
룰렛게임은 5월1일~5월31일중에 100번을 할 수 있으며 한번 게임을 한 후엔 , 30분이상이 지나야 또 다시 게임을 할 수 있다. 각 룰렛에서 얻는 최대 포인트는 3포인트이다. 이 룰렛게임을 통해 얻은 누적포인트로 HINT1(50 point) , HINT2(100 point) , ANSWER(1000 point)등을 살 수 있다.
포인트를 관리하는 페이지에서 타임스탬프를 인자로 받아 관리함을 이용해 , 타임스탬프를 30분에 해당하는 수만큼 증가시켜가며 접속시켜 포인트를 빠르게 늘릴 수 있었지만 100번까지만 가능하기때문에 최대 300포인트까지 밖에 올릴 수 없다.
그래서 힌트를 읽어보니 두 힌트는 아래와 같다.

HINT1: timestamp

HINT2 : UPDATE quiz set timestamp=(value) , point=(value) , counter = (value) WHERE ip_addr = (ip)

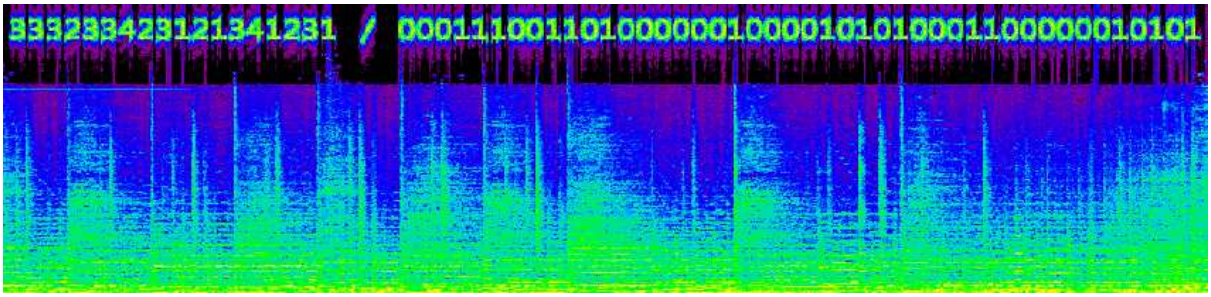
두 힌트로 미루어보아 위 쿼리의 timestamp value부분에서 sql 인젝션 취약점이 있다고 생각했지만 원하는대로 되지않고 오랜시간이 지나 그냥 자고 일어났더니 다른참가자의 실수로 내 포인트가 213574357점으로 올라가있었다.
ㄲ

Flag : NewHeartBeat

Level8

http://1.221.63.146:10007/lv8/wavwav.wav

주어진 파일은 예전에 방영됐던 드라마 NewHeart 의 OST 가 있는 wav 음악파일이다.
음악파일안에 string 이나 lsb 를 모아봐도 특별한 정보를 얻을 수 없어 해당 음악파일의 스펙트럼을 확인해보았다.



위 사진과 같이 스펙트럼에서 숫자들이 나타나는 것을 불 수 있다.

```
333233423121341231 / 00011100110100000010000101010001100000010101
```

왼쪽에 있는 숫자만큼 오른쪽에있는 숫자를 나누고 , 나누어진 각 숫자들을 모스부호로서 문자로 변환해보았다.

```
000/111/001/10/100/000/0100/00/101/0/10/0/011/0000/0/01/010/1
3   3   3  2  3   3   4    2  3 1 2 1 3   4   1 2 3   1
```

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

( 0->. , 1-> - )
=> ... --- ..- -. -.. ... .-.. .. -.- . -. . .-- .... . .- .-. -
=> S   O   U   N  D  S   L   I  K   E N  E W  H   E  A  R  T
```

0 을 .으로 , 1 을 –로 나타내어 모스부호로서 알파벳으로 변환해보니 SOUNDLIKENEWHEART 라는 문자열이 나왔고 , 이를 소문자로 인증했더니 인증에 성공했다.

Flag : soundslikenewheart

Level9

Download Date? (yyyyMMddHHmmss) http://1.221.63.146:10007/lv9/Prob.zip

주어진 파일은 Internet Explorer History File 이다.

```
pwn3r@localhost:~/ctf/newheart/level9$ file Prob
Prob: Internet Explorer cache file version Ver 5.2
```

Internet Explore History File 은 접속한 URL 과 마지막 수정 시간 , 마지막 접속 시간 , 캐시 폴더 이름 , 파일명 , HTTP Header 등을 저장해둔다.
Index.dat analyzer 라는 프로그램을 이용해 저장된 정보를 간단하게 정리해서 볼 수 있다.

Name	FileName	FileType	Type	Created	Accessed
DOMStore:http://www.naver.com/	www.naver[1].xml	.xml	URL	01/01/1601 00:00:00	05/18/2012 10:39:42
DOMStore:http://www.naver.com/	www.naver[1].xml	.xml	URL	01/01/1601 00:00:00	05/02/2012 10:10:27
DOMStore:http://www.pcworld.com/	www.pcworld[1].xml	.xml	URL	01/01/1601 00:00:00	01/19/2012 10:54:28
DOMStore:http://download.cnet.com/	download.cnet[1].xml	.xml	URL	01/01/1601 00:00:00	01/19/2012 10:55:14
DOMStore:http://sourceforge.net/	sourceforge[1].xml	.xml	URL	01/01/1601 00:00:00	01/19/2012 10:55:51

download 라는 서브도메인이 붙어있는것으로 보아 http://download.cnet.com 에서 파일을 다운로드것으로 추측할 수 있다. 파일을 다운로드한 날짜는 download.cnet.com 에 접속한 10/19/2012 10:55:14 이므로 문제에서 요구한 형식(yyyyMMddHHmmss)대로 맞추면 인증키는 20120119105514 이다.

*관련문서 : http://forensicinsight.org/wp-content/uploads/2012/03/INSIGHT_Web-Browser-Forensics_Part1.pdf

Flag : 20120119105514

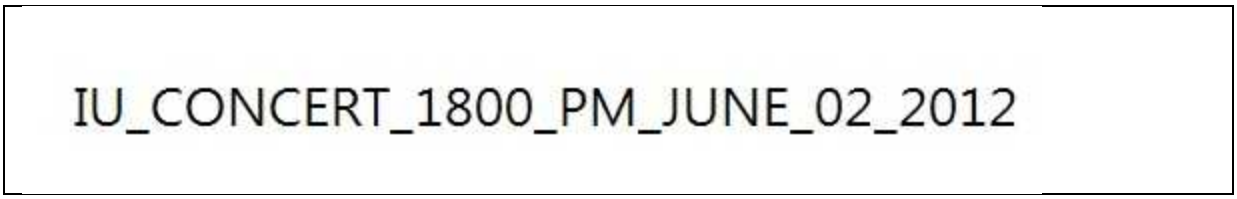
Level10

http://1.221.63.146:10007/lv10/android.zip NewHeart 수사대는 어떤 사건을 수사하던 중 마약 사건에 관련된 범인을 체포하였다. 범인은 마약을 밀거래하는 사람으로 특정일 특정장소에서 밀거래상과 접선할 예정이었다는 점을 자백하였으나 수사대는 더 이상의 자세한 내용은 밝혀내지 못했다. 유일한 단서는 범인이 가지고 있던 스마트폰으로, 암거래상과 정보를 주고 받았을 가능성이 높다. 암거래상과의 접선 장소 및 시간을 찾아라.

주어진 파일은 안드로이드 폰의 전체 데이터를 백업한 것이다.

acct	2012-05-21 오전...	파일 폴더	
cache	2012-05-17 오전...	파일 폴더	
config	2012-05-17 오전...	파일 폴더	
data	2012-05-21 오전...	파일 폴더	
dev	2012-05-21 오전...	파일 폴더	
mnt	2012-05-17 오전...	파일 폴더	
proc	2012-05-17 오전...	파일 폴더	
root	2012-05-17 오전...	파일 폴더	
sbin	2012-05-21 오전...	파일 폴더	
sys	2012-05-21 오전...	파일 폴더	
system	2012-05-21 오전...	파일 폴더	
default.prop	2012-05-17 오전...	PROP 파일	1KB
init	2012-05-17 오전...	파일	97KB
init.goldfish.rc	2012-05-17 오전...	RC 파일	3KB
init.rc	2012-05-17 오전...	RC 파일	17KB
ueventd.goldfish.rc	2012-05-17 오후...	RC 파일	1KB
ueventd.rc	2012-05-17 오후...	RC 파일	4KB

암거래상과 정보를 주고 받았다면 제일 먼저 문자메시지를 의심해 볼 수 있다.
안드로이드에서 문자메시지는
/data/data/com.android.providers.telephony/databases/mmssms.db 에 저장되며 , SMS 일 때 ,
첨부파일등은 /data/data/com.android.providers.telephony/app_parts 에 저장된다.
우선 첨부파일을 확인해보니 "PART_13372276170"라는 파일명의 jpg 파일이 하나 있었다.



< PART_13372276170 >

이를 열어보니 인증키가 적혀있어 인증에 성공했다..

Flag : IU_CONCERT_1800_PM_JUNE_02_2012