

제 1회 청소년 화이트해커 경진대회



1위 persona(이정훈)

lokihardt@gmail.com

Level 1.

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

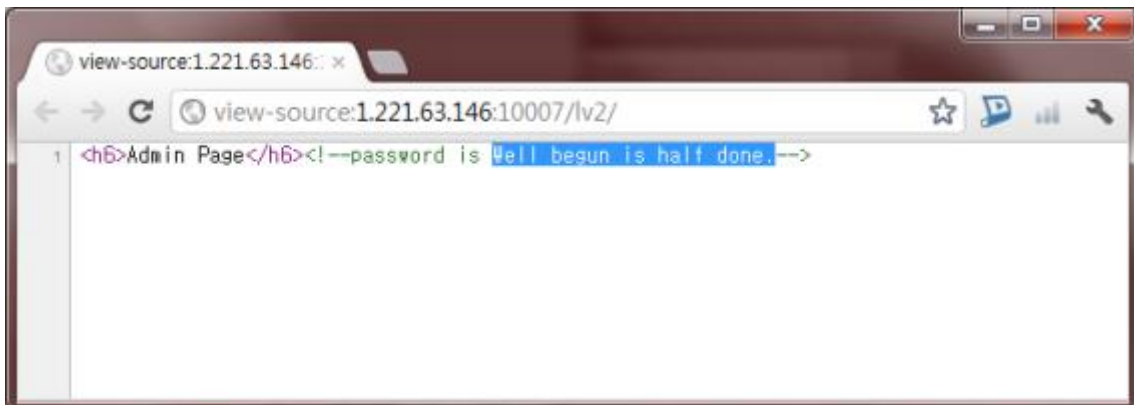
개인적으로 이번 대회에서 가장 어려웠던 문제입니다.
공학용 계산기의 힘을 빌려 가까스로 풀어내는데 성공했습니다.

Key : 9192173798

Level 2.



스마트폰의 User-Agent로 변경 한 뒤 접속했습니다.



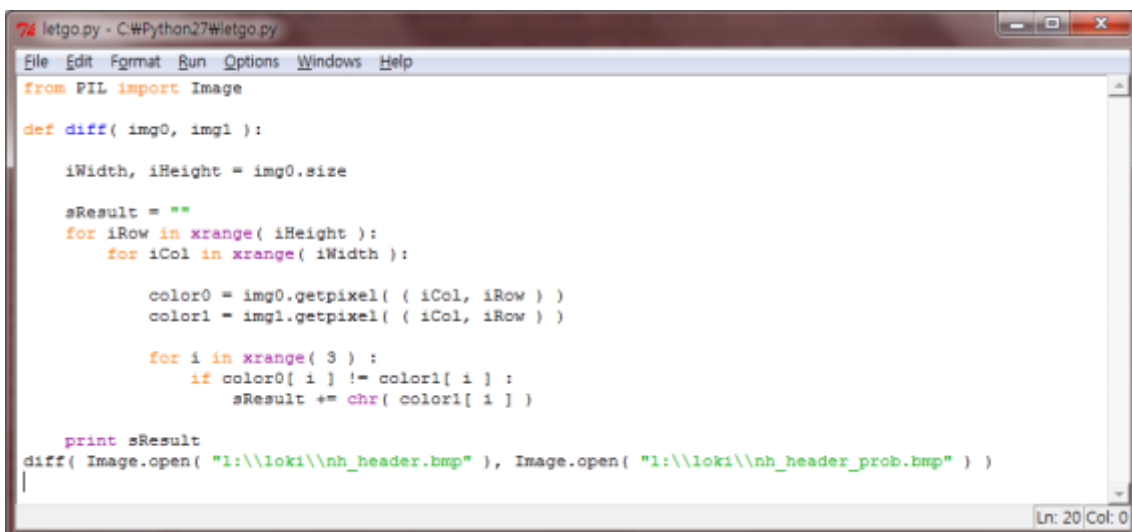
Key : Well begun is half done.

Level 3.

누군가 뉴하트 홈페이지의 로고에 비밀 키를 숨겨놓았다..
비밀 키를 찾아라.

(원본 로고 : http://1.221.63.146:10007/lv3/nh_header.bmp) (변조된 파일 :
http://1.221.63.146:10007/lv3/nh_header_prob.bmp)

처음에는 원본 로고, 변조된 로고가 나뉘져 있지 않아 뉴하트의 홈페이지에 접속하여 로고 이미지를 받은 뒤 문제의 파일과 RGB 비교를 했습니다.



```
letgo.py - C:\Python27\letgo.py
File Edit Format Run Options Windows Help
from PIL import Image

def diff( img0, img1 ):
    iWidth, iHeight = img0.size
    sResult = ""
    for iRow in xrange( iHeight ):
        for iCol in xrange( iWidth ):
            color0 = img0.getpixel( ( iCol, iRow ) )
            color1 = img1.getpixel( ( iCol, iRow ) )
            for i in xrange( 3 ):
                if color0[ i ] != color1[ i ] :
                    sResult += chr( color1[ i ] )
            print sResult
diff( Image.open( "l:\\loki\\nh_header.bmp" ), Image.open( "l:\\loki\\nh_header_prob.bmp" ) )
Ln: 20 Col: 0
```

!!tr@ewhnesiyke

Key : newhe@rt!!

Level 4.



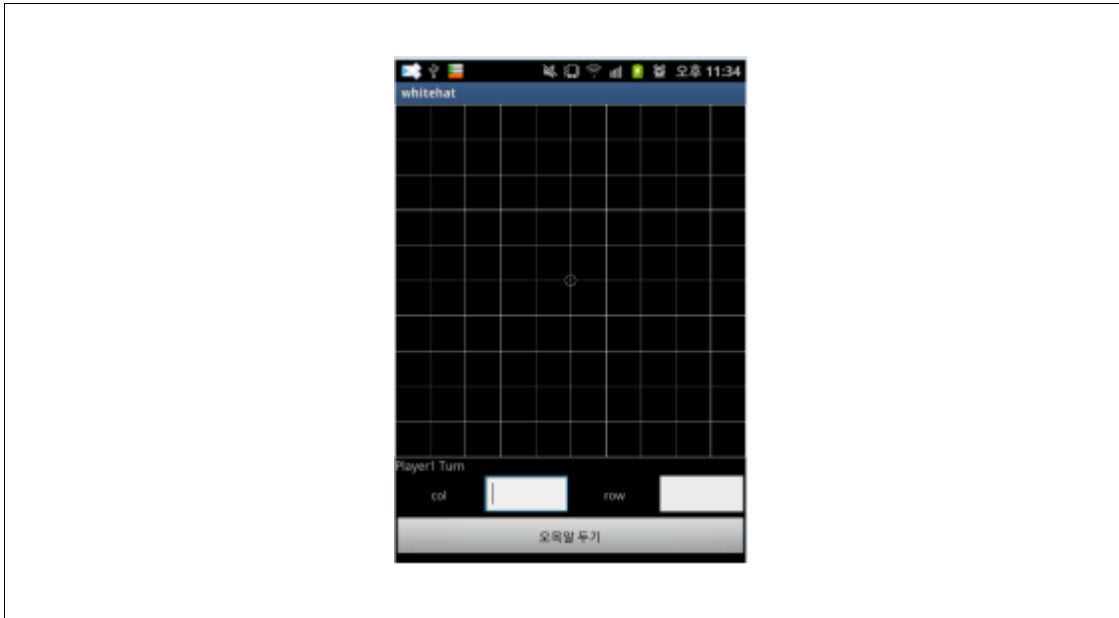
```
<decodebase85~aa$dEtfS+x&&hj+df#kgl*sfgmlDvbn@d/  
~GiagsAQT#hhsso@tr;Twm$Y(le^ojs^4dawh?&ftzlwrgpIY  
)hOes!d*jgttmR^^zgngfdIxBdb~!!Op@cxuat}svf)vmst!z  
1ropdtr&*Dity3c)3duaev*cvRtsb&4zt0dnads8hd@lk^jad  
1879rwe#d$#ytr./dsudr>m^&ifg?bnBoahcv&(p4jxz#*lkt  
d@#sda7zxcvb^^8o$aF1%7hg*23elj)hklk,dfbfp>mnb,mo  
@aytkotz;r((udx&l*dtkqae)0tl%^xr18;wvcvd~jbnzdfg>
```

단순하게 문자열을 매트릭스 이미지에 순서대로 넣은 뒤 색이 칠해진 부분만 출력했습니다.

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

Key : hello_hacking_festival!!

Level 5.



```
    }  
  
    public void oihheng()  
    {  
        Toast.makeText(this, decrypt("ygbahi?+hih5vrhhsblr"), 1).show();  
    }  
  
    public void onCreate(Bundle paramBundle)  
    {
```

리버싱을 통해 엔지 키 값처럼 보이는 것을 출력하는 부분을 발견하여
인젝션으로 해당 함수의 결과 값을 출력했습니다.

```
171     try  
172     {  
173         Class<?> cs = m_appCurr.getClassLoader().loadClass( "com.newh.whitehat.WhitehatActivity" );  
174         Method mDecrypt = Helper.getMethod( m_appCurr.getClassLoader().loadClass( "com.newh.whitehat.WhitehatActi  
175         Log.i( "CH", mDecrypt.getName() );  
176         Log.i( "CH", (String)mDecrypt.invoke( cs.newInstance(), "ygbahi?+hih5vrhhsblr" ) );  
177     }  
178     catch (Exception e) {  
179         // TODO: handle exception  
180     }  
---
```

Key : diablo3+lol=hellgate

Level 6.

 ApplicationIcon.png	2012-05-16 오후...	PNG 파일	2KB
 AppManifest.xml	2012-05-16 오후...	Windows 태그 파일	1KB
 Background.png	2012-05-16 오후...	PNG 파일	4KB
 nhf3.dll	2012-05-16 오후...	응용 프로그램 확장	16KB
 SplashScreenImage.jpg	2012-05-16 오후...	JPG 파일	10KB
 WMAAppManifest.xml	2012-05-16 오후...	XML 문서	2KB

Windows Phone 리버싱 문제였습니다.

디컴파일러를 통해 분석을 해보니 9개의 버튼을 누르면 salt 라는 배열에 차례대로 값을 넣은 뒤 submit을 통해 해당 배열을 참조하는 복호화 루틴을 실행하고 있었습니다.

버튼을 누르는 순서는 Description에 적혀 있었으며 디자인 파일의 추출을 통하여 버튼의 값들을 알 수 있었습니다. 해당 정보들을 연계하여 c# 프로그래밍을 통해 키를 얻을 수 있었습니다.

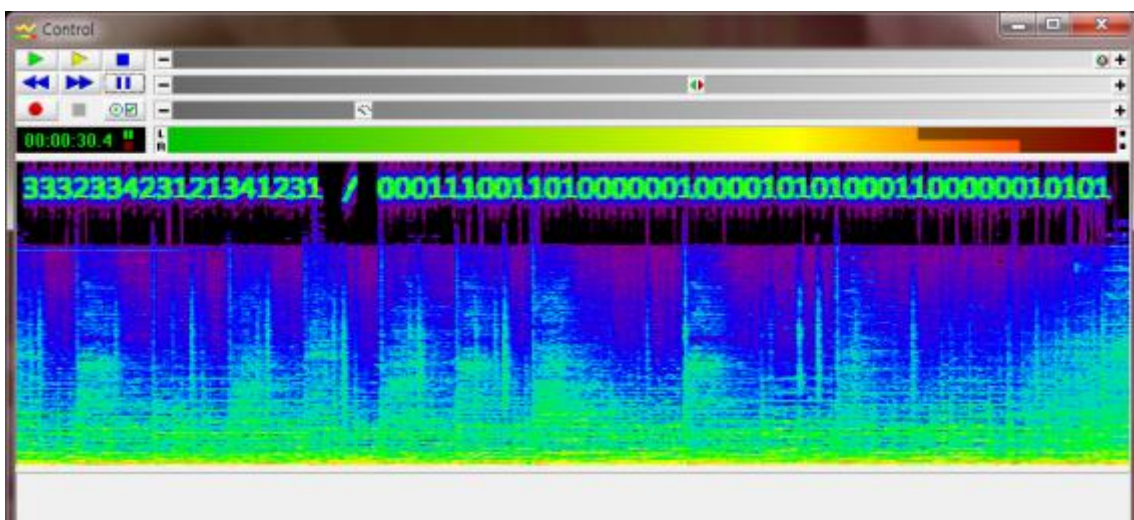
Key : u+Vscbgx4hX8Onbrk0dH8Rxcbdg1FnCOH8xn2Uy8aDkoUk4hcHvRK/LGpuMCqQ8N

Level 7.

대회 종료 후 해당 문제 서버 접속이 안되는 관계로 구술로 대체합니다.
웹해킹 문제였으며 30분 마다 한 번씩 룰렛을 돌릴 수 있으며
한번 돌릴 때 마다 0~3점의 포인트를 획득 할 수 있습니다.
해당 포인트들을 모아 1000점을 만들면 Key값을 획득 할 수 있었습니다.
point 값을 서버로 보낼 때 timestamp 의 값도 함께 보내는데
timestamp 변수의 값을 조작하여 SQL Injection에 성공하여 키를 획득
하였습니다.

Level 8.

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



해당 wav 파일의 스펙트럼을 살펴보았습니다.

표시되는 값들이 모스코드와 그의 띄어쓰기로 판단했습니다.

Key : SOUNDSLIKENEWHEART

Level 9.

Download Date ?

(YYYYMMddHHmmss)

<http://1.221.63.146:10007/lv9/Prob.zip>

압축을 풀면 URLCache 파일이 나오는데 IEHist를 사용하여 키를 획득
하였습니다.

Key : 20120119105514

Level 10.

<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.

압축을 풀어 모든 파일을 합친 뒤 카빙을 통하여 답이 담긴 jpg 파일을 획득 했습니다.



IU_CONCERT_1800_PM_JUNE_02_2012

Key : IU_CONCERT_1800_PM_JUNE_02_2012

Level 11.

<http://1.221.63.146:10007/lv11/newheart.sys> VM상에서 실행하셔야 합니다!

해당 sys 파일을 IDA로 분석하여 키 값을 출력하는 부분을 c로 작성하였습니다.

```

char g_cKey[ 1024 ] = "";
char g_cKeyTable[ 1024 ] = "";

void keyTableGen( int iLength )
{
    char cTap[ 1024 ] = "";

    double dFlo = +( (double+)"0x380x2E0x30x060x9A0x090x3F" );

    for( int i = 0; i < 10000; i++ )
        dFlo = 4.0 + dFlo + ( 1.0 - dFlo );

    int iTap = -1;

    for( int i = 0; i < 8 * iLength; i++ )
    {
        if( !( i % 8 ) )
            ++iTap;
        cTap[ iTap ] += 2;

        cTap[ iTap ] = ( dFlo > 0.5 ) ? cTap[ iTap ] :
        dFlo = 4.0 + dFlo + ( 1.0 - dFlo );
    }

    for( int i = 0; i < 40; ++i )
        g_cKeyTable[ i ] = cTap[ i ];
}

void keyGen( const char* pTable )
{
    int n = strlen( pTable );

    keyTableGen( n );

    for( int i = 0; i < 24; i++ )
        g_cKey[ i ] = g_cKeyTable[ i ] ^ pTable[ i ];

    printf( "%s\n", g_cKey );
}

void main()
{
    keyGen( "0204214838FD15AA8BE3E046D93E7F387"
           "038E84482992DF8ABCDFFDFDFDF000000"
           "F7377F21CFF00060000000006400000" );
}

```

Key : Pocari_SWEAT

Level 12.

<http://1.221.63.146:10007/lv12/prob.mov>

해당 동영상 중간 중간에 텍스트가 깜빡 하며 지나갑니다.

텍스트들을 모두 모아 합쳐보면

```
base64( "TWVzc2FnZSA9IHYwbl9tc2cucGhw" )
```

```
Message = v0n_msg.php
```

위와 같은 문자열이 나옵니다.

```
This is a hint message for this step. You will get the answer from the following formula. The 'f' is a function to help you find the key message. And the 'input' is an argument of the function. Finally output is a result of 'f(input)'. When 'f'. 'f' == 'v0n_f.html' and 'output' == 'v0n_output', FIND THE KEY FROM INPUT!
```

해당 페이지를 들어가 보면 다음과 같은 설명이 적혀있습니다.

v0n_f.html을 들어가 보면 난독화 된 암호화 자바스크립트 함수가 있습니다.

해당 함수를 분석하여 복호화 함수를 만들어 v0n_output 파일을 복호화 하였습니다.

```
public static void decrypt( byte[] arg )
{
    for( int i = 0; i < arg.length / 11; i++ )
    {
```

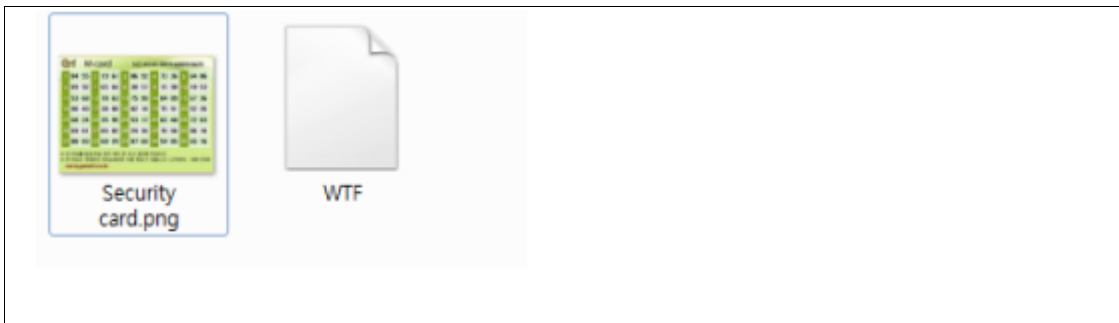
```
        if( (char)( arg[ i * 11 + 3 ] ) < 0xf0 )
            System.out.print( (char)arg[ i * 11 + 3 ] );
        else
            System.out.print( String.format( "%d", arg[ i * 11 + 3 ] + 16 ) );
    }
}

public static void main( String[] argv ) throws Exception
{
    byte[] cs = { (byte)0x4D, (byte)0x5A, (byte)0xAE, (byte)0x74, (byte)0x50,
                 (byte)0xA8, (byte)0x2D, ... }; //v0n_output
    decrypt( cs );
}
```

Key : th1s1sthech4ll3ng3f0ry0urfutur3

Level 13.

Title: Restore PNG



TrueCrypt를 사용하여 WTF 파일을 마운트 한 뒤 ADS 영역을 분석하여 나뉘져 있는 PNG 파일들을 모두 합쳤습니다.

KEY
SY573M_ADS_
1S_AMAZING

Key : SY573M_ADS_1S_AMAZING

Level 14.

<http://1.221.63.146:10007/lv14/ppppp.pcap>

어느 해커의 컴퓨터에서 발견한 파일이다

해당 pcap을 분석하다보면 attack.newheart.kr로 들어가는 URL들이 있는데 그 중 로그인을 시도하는 URL을 타고 들어가면 Key가 출력됩니다.

Level 15.

<http://1.221.63.146:10007/lv15/findm2.7z>

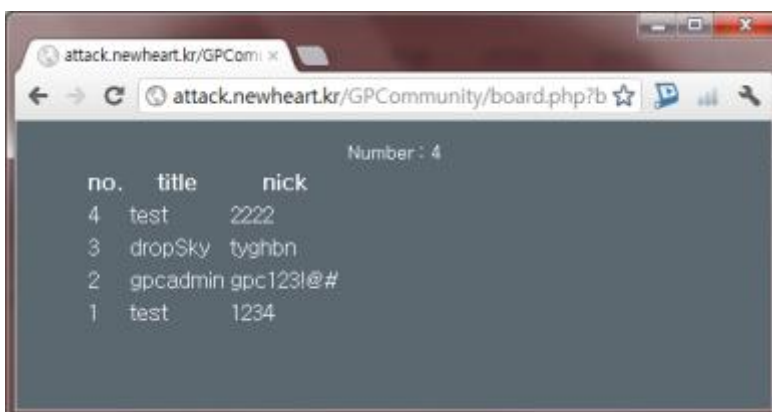
인증 키 형식 : id_pw

VMWare 이미지가 주어집니다. 해당 VMWare 이미지를 분석하여



웹페이지의 방문기록을 분석해보면

위와 같은 사이트로 접속합니다.



SQL Injection을 통해 ID와 Password를 추출 하였습니다.

Key : gpcadmin_gpc123!@#

Level 16.

<http://1.221.63.146:10007/lv16/login.7z>

Themida로 패킹 된 exe를 발견 할 수 있으며
해당 exe는 특정 서버와 ssl 통신을 하며 키를 받아옵니다.
키를 받아오기 전의 메모리안의 문자열들과
받아 온 후의 메모리안의 문자열들을 비교하여
키 값을 얻을 수 있었습니다.

Level 17.

<http://1.221.63.146:10007/lv17/golollol.zip> Find LOL ID & PW (Password use copy & Paste)

(Key는 PW_ID 형식으로 작성)

VMWare 메모리 파일이 주어집니다.

LOL 게임의 ID와 Password를 찾는 문제인데

실제 LOL 게임의 메모리와 비교하여 ID와 Password를 획득 하였습니다.

Level 18.

<http://1.221.63.146:10007/lv18/sample.apk>

APK를 분석하여 보면 JNI 라이브러리와 링크되어 있습니다.

해당 JNI 라이브러리를 분석해보면 decrypt라는 사용되지 않는 함수가 존재합니다.

c언어로 ARM 리눅스용으로 컴파일을 하여 해당 함수를 호출해 Log를 통해 Key를 얻었습니다.

```
int main( int argc, char** argv )
{
    typedef void (*Decrypt);

    void* pLibrary = dlopen( "/data/libJNIConnector.so", RTLD_NOW );

    Decrypt pDecrypt = (Decrypt)dlsym( pLibrary, "_Z7decryptv" );

    pDecrypt();
    dlclose( pLibrary );
    return 0;
}
```

Key : 0e7ebe0783_tre0c6d7aa2b15951371eee1ec5feda6a

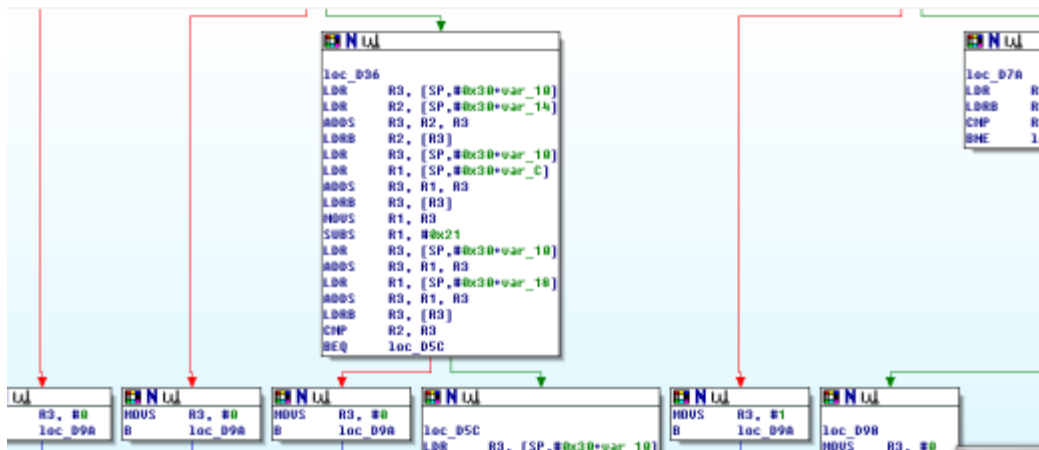
Level 19 – Final Round.

<http://1.221.63.146:10007/lv19/hack-me.apk>

해당 안드로이드 어플리케이션은 비밀번호를 입력받아 체크를 하는 어플리케이션입니다.

입력 받은 비밀번호는 Java 단에서 체크를 하는게 아닌 외부 JNI 라이브러리를 통해 체크하였습니다.

IDA를 사용하여 해당 라이브러리를 분석하였습니다.



위와 같은 부분이 비밀번호를 비교하는 핵심 부분입니다.

`var_C` = 입력한 Password

`var_14` = Static Key Table

`var_18` = 중간에서 생성되는 Key Table

`var_10` = 1 ~ `strlen(var_C)`

gdb를 통해 Key Table들의 값들을 찾은 뒤 c언어로 따로 코딩을 하여 답을 얻을 수 있었습니다.

```

void main()
{
    char* pKeyTable0 = "\xB4\x79\xC5\xD2\xAF\x20\x9F\xCE\xE0\x20.....";
    char* pKeyTable1 = "\x03\x12\x29\x44\x5F\x66\x8D\x90\xBB\xCA....." ;

    for( int i = 0; i < 14; i++ )
    {
        for( int j = 0x21; j < 255; j++ )
        {
            if( pKeyTable0[ i ] == pKeyTable1[ j - 0x21 + i ] )
            {
                printf( "%c", j );
                break
            }
        }
    }

    printf( "\n" );
}

```

Key : 4RM_1s_g0od!!!