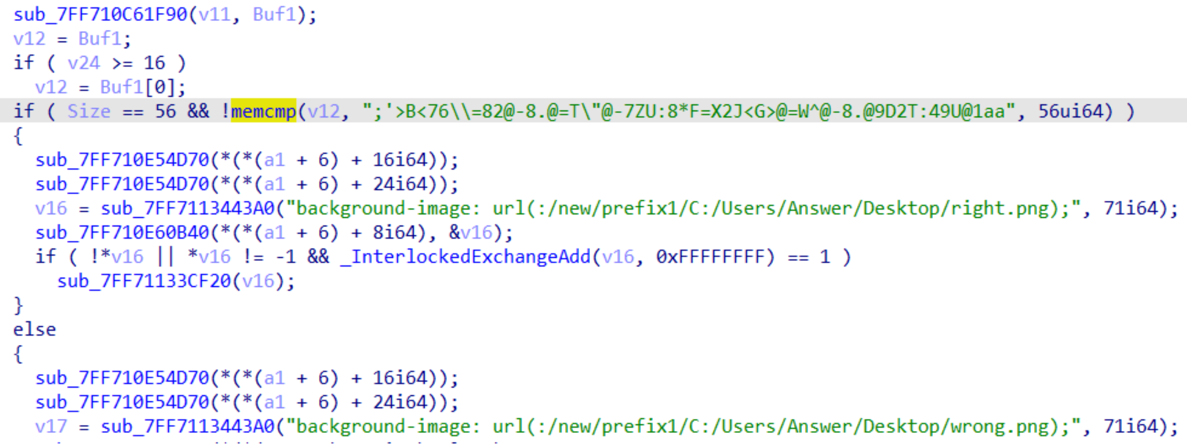
Week 3

Re

Answer's Windows

拖进IDA里面后发现非常多库函数，要找到作者写的函数，根据为数不多的经验，一般来说可以找输入或者是输出，输入找不到（？，就选择找输出，输出通常有判断条件的字符串，在字符串面板猜几个常见的关键词（例如input,congratulations,flag,wrong这类）发现这两个看起来不太寻常的路径，大致翻译一下这是张放到桌面的背景图片，一般情况下只有用户才会这么干，加上附近发现“right”的条件是一个比较字符串的函数，所以非常可疑，可以尝试分析



```
sub_7FF710C61F90(v11, Buf1);
v12 = Buf1;
if ( v24 >= 16 )
  v12 = Buf1[0];
if ( Size == 56 && !memcmp(v12, ";'>B<76\\=82@-8.@=T\"@-7ZU:8*F=X2J<G>@=W^@-8.@9D2T:49U@1aa", 56ui64) )
{
  sub_7FF710E54D70(*(*(a1 + 6) + 16i64));
  sub_7FF710E54D70(*(*(a1 + 6) + 24i64));
  v16 = sub_7FF7113443A0("background-image: url(:/new/prefix1/C:/Users/Answer/Desktop/right.png);", 71i64);
  sub_7FF710E60B40(*(*(a1 + 6) + 8i64), &v16);
  if ( !*v16 || *v16 != -1 && _InterlockedExchangeAdd(v16, 0xFFFFFFFF) == 1 )
    sub_7FF71133CF20(v16);
}
else
{
  sub_7FF710E54D70(*(*(a1 + 6) + 16i64));
  sub_7FF710E54D70(*(*(a1 + 6) + 24i64));
  v17 = sub_7FF7113443A0("background-image: url(:/new/prefix1/C:/Users/Answer/Desktop/wrong.png);", 71i64);
```

分析后发现sub_7FF710C61F90这个函数，是base64算法，再通过交叉引用找到table，用CyberChef解密即可得到flag！

creakme3

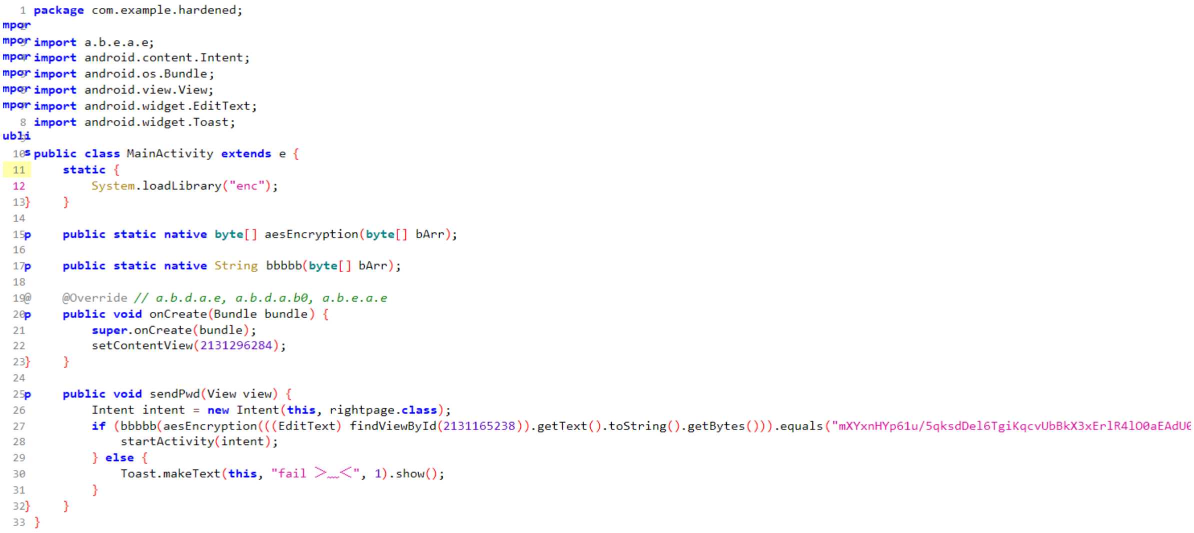
拖进IDA里面发现是PowerPC汇编，经过短暂的学习发现还是找工具吧），使用Ghidra分析，找到main函数分析后找数据并简化算法，得到flag

代码：）

```
#include<stdio.h>
int main()
{
    int j;
    int b[89]={0};
    int a[178] =
{48,20093,48,26557,48,31304,48,33442,48,37694,49,39960,50,23295,50,27863,50,42698,50,48505,50,52925,51,12874,51,12946,51,14597,51,17041,51,23262,51,28319,51,42282,51,48693,51,52067,53,32571,56,14612,56,45741,57,14554,57,20048,57,27138,57,45327,66,30949,95,32502,95,35235,95,36541,95,38371,97,29658,100,21388,100,25403,100,40604,100,46987,100,51302,101,12974,101,30329,102,10983,102,19818,102,22280,102,26128,102,41560,102,47116,102,51333,103,28938,103,31988,104,16246,104,28715,104,41966,104,44368,104,47815,105,16420,105,35362,105,49237,106,11090,106,50823,107,24320,107,50199,108,24962,109,30171,110,15457,110,18838,110,24001,111,11638,111,32023,111,43291,112,39661,114,17872,114,33895,114,43869,115,20611,115,25122,115,36243,115,37434,115,38686,115,46266,115,51077,116,13656,116,34493,116,38712,117,14096,117,38777,119,12095,119,17629,123,30945,125,40770 };
    int t;
    for (int y = 0; y < 0x59; y++)
    {
        b[y] = y;
    }
    for (int i = 0; i < 0x59-1; i++)
    {
        for (int k = 0; k < 0x59 - i - 1; k++)
        {
            if (a[(b[k + 1] * 2 +1)] < a[(b[k] * 2 +1)])
            {
                t = b[k + 1];
                b[k + 1]= b[k];
                b[k] = t;
            }
        }
    }
    printf("Welcome my whitegive re task! This is your flag: ");
    for (j = 0; j < 0x59; j++)
    {
        printf("%c",a[(b[j] * 2)]);
    }
    return 0;
}
```

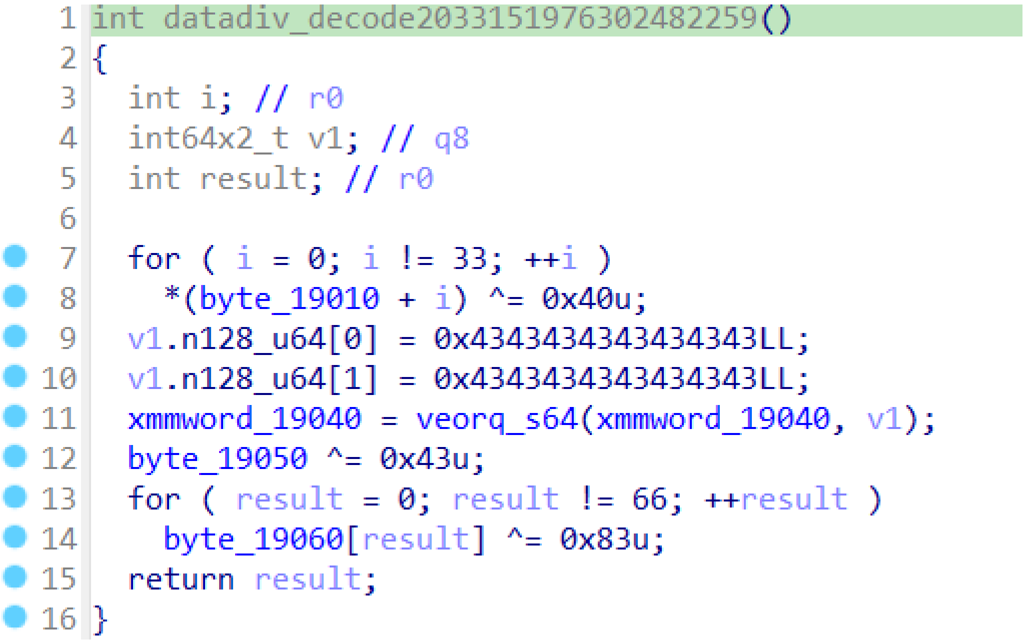
hardened

根据0w1学姐的hint，先用BlackDex64在手机上脱壳，脱壳之后用jadx-gui打开分析main函数，发现找不到aesencryption（猜的是AES加密）函数还有bbbbb函数，根据4nsw3r学长的hint还有main函数中出现System.loadLibrary("enc")去分析原文件夹里的动态链接库。



```
package com.example.hardened;

import a.b.e.a.e;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends e {
    static {
        System.loadLibrary("enc");
    }

    public static native byte[] aesEncryption(byte[] bArr);

    public static native String bbbbb(byte[] bArr);

    @Override // a.b.d.a.e, a.b.d.a.b0, a.b.e.a.e
    public void onCreate(Bundle bundle) {
        super.onCreate(bundle);
        setContentView(2131296284);
    }

    public void sendPwd(View view) {
        Intent intent = new Intent(this, rightpage.class);
        if (bbbbb(aesEncryption(((EditText) findViewById(2131165238)).getText().toString().getBytes())).equals("mXYxnHYp61u/5qksdDel6TgiKqcvUbBkX3xErlR4lO0aEAdU
            startActivity(intent);
        } else {
            Toast.makeText(this, "fail >_<", 1).show();
        }
    }
}
```

找到加密函数，根据AES算法的特点找到key跟iv，用交叉引用查找数据替换，查找发现其中veorq_s64这个函数作用是按位异或，同时因为猜测到明文的base64格式找到在同样的地方替换过的table



```
int datadiv_decode2033151976302482259()
{
  int i; // r0
  int64x2_t v1; // q8
  int result; // r0

  for ( i = 0; i != 33; ++i )
    *(byte_19010 + i) ^= 0x40u;
  v1.n128_u64[0] = 0x4343434343434343LL;
  v1.n128_u64[1] = 0x4343434343434343LL;
  xmmword_19040 = veorq_s64(xmmword_19040, v1);
  byte_19050 ^= 0x43u;
  for ( result = 0; result != 66; ++result )
    byte_19060[result] ^= 0x83u;
  return result;
}
```

写代码处理一下数据

```
#include<stdio.h>
int main(void)
{
    int table[66]={
  0xB3, 0xB2, 0xB1, 0xB0, 0xB7, 0xB6, 0xB5, 0xB4, 0xBB, 0xBA,
  0xC2, 0xC1, 0xC0, 0xC7, 0xC6, 0xC5, 0xC4, 0xCB, 0xCA, 0xC9,
  0xC8, 0xCF, 0xCE, 0xCD, 0xCC, 0xD3, 0xD2, 0xD1, 0xD0, 0xD7,
  0xD6, 0xD5, 0xD4, 0xDB, 0xDA, 0xD9, 0xE2, 0xE1, 0xE0, 0xE7,
  0xE6, 0xE5, 0xE4, 0xEB, 0xEA, 0xE9, 0xE8, 0xEF, 0xEE, 0xED,
  0xEC, 0xF3, 0xF2, 0xF1, 0xF0, 0xF7, 0xF6, 0xF5, 0xF4, 0xFB,
  0xFA, 0xF9, 0xA8, 0xAC, 0xBE, 0x83 };
    for (int i = 0; i < 66; i++)
    {
        table[i] ^= 0x83;
        printf("%c", table[i]);
    }
    char iv[16] = {':',',','6','\x1C','%','*','-
','\'','\x1C','.','&','b','b','b','b','b'};
    char key[32] =
{'\n','\x15','\x13','\x14','\x1F','\x01','\x1F','\x0E','\x0F','\x12','\r','\x01'
,'\f','\x1F','\v','\x05','\x19','\x1F','\x06','\x0F','\x12','\x1F','\x19','\x0F'
,'\x15','\x1F','\x14','\x0F','\x1F','\x04','\x05','\x03'};
    for (int i = 0; i < 32; i++)
    {
        key[i] ^= 0x40u;
        printf("%x", key[i]);
    }
    for (int i = 0; i < 16; i++)
    {
        printf("%x", iv[i]^=0x43);
    }
    return 0;
}
```

获得正确的三个关键数据后放进CyberChef就能得到flag啦！