【jarvisoj刷题之旅】逆向题目DDCTF - Android Easy的writeup

原创

iqiqiya 于 2018-09-09 18:44:39 发布 557 收藏 1

分类专栏： 我的逆向之路 我的CTF之路 我的CTF进阶之路 文章标签： 【jarvisoj刷题之旅】逆向题目DDCTF - Andr DDCTF - Android Easy Android Easy的writeup reverse

本文链接：https://blog.csdn.net/xiangshangbashaonian/article/details/82561029

版权

我的逆向之路 同时被 3 个专栏收录

108 篇文章 10 订阅

订阅专栏

我的CTF之路

92 篇文章 5 订阅

订阅专栏

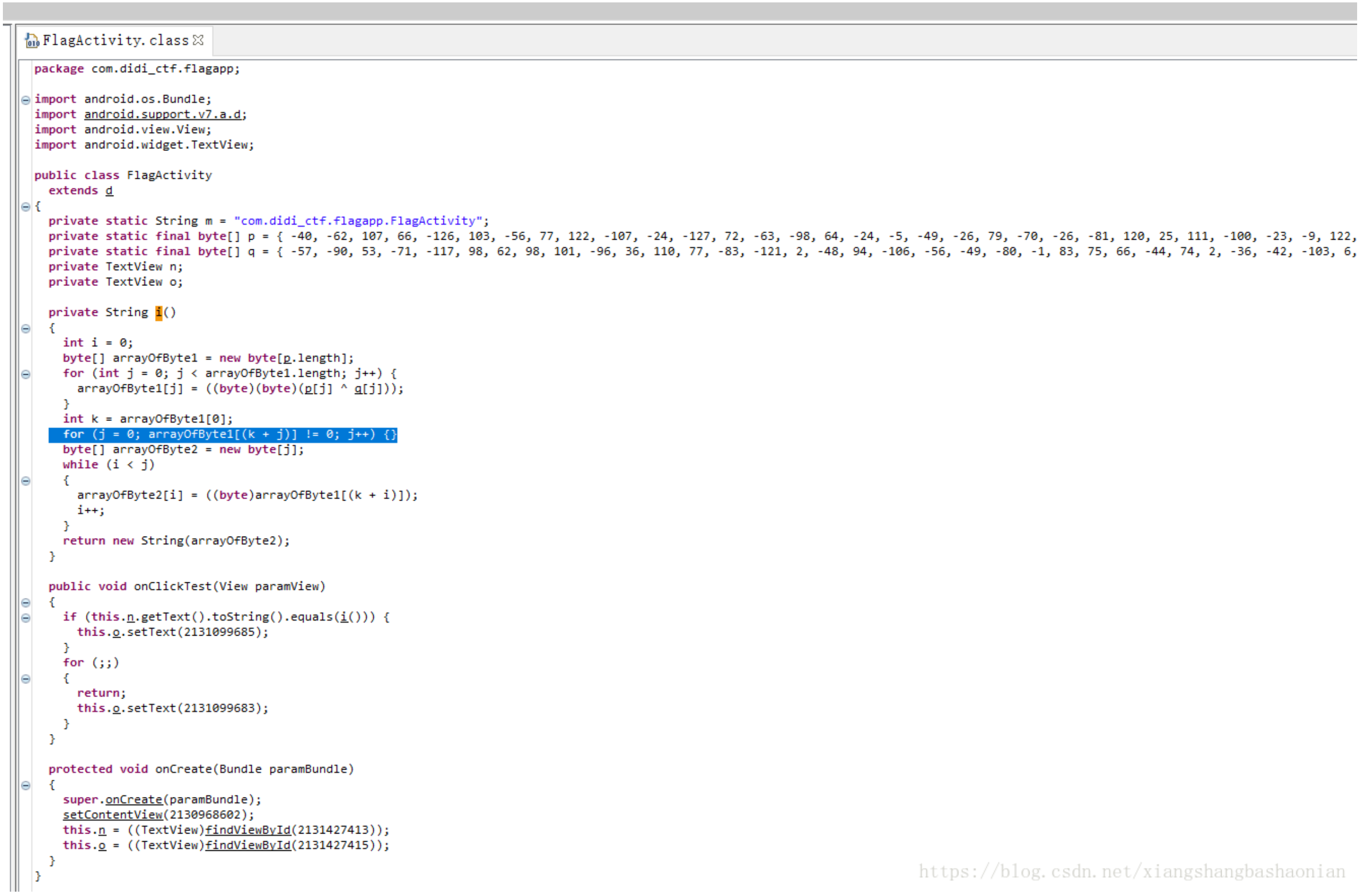
我的CTF进阶之路

108 篇文章 18 订阅

订阅专栏

下载附件之后　改后缀为.apk

本来是直接载入jd-gui的　结果不好看明白代码



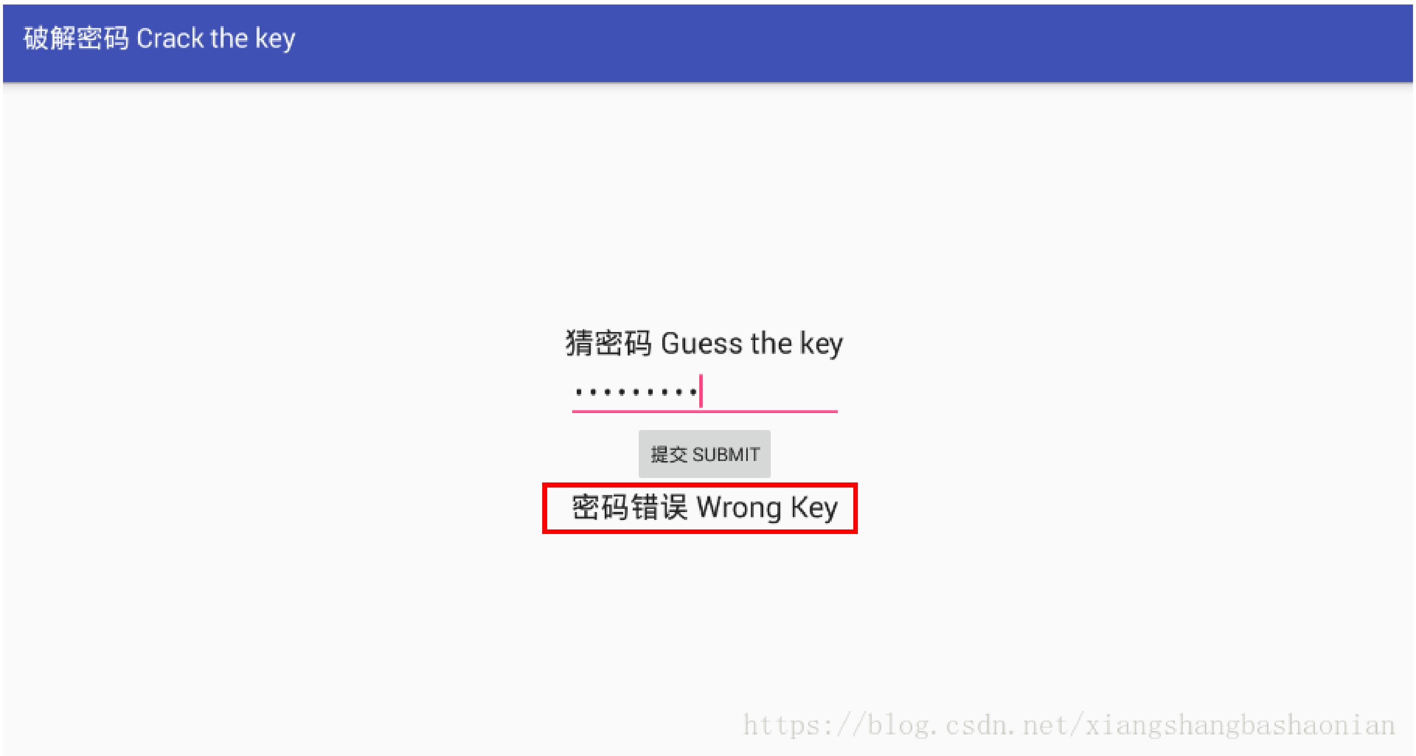
FlagActivity.class

```java
package com.didi_ctf.flagapp;

import android.os.Bundle;
import android.support.v7.a.d;
import android.view.View;
import android.widget.TextView;

public class FlagActivity
  extends d
{
  private static String m = "com.didi_ctf.flagapp.FlagActivity";
  private static final byte[] p = { -40, -62, 107, 66, -126, 103, -56, 77, 122, -107, -24, -127, 72, -63, -98, 64, -24, -5, -49, -26, 79, -70, -26, -81, 120, 25, 111, -100, -23, -9, 122,
  private static final byte[] q = { -57, -90, 53, -71, -117, 98, 62, 98, 101, -96, 36, 110, 77, -83, -121, 2, -48, 94, -106, -56, -49, -80, -1, 83, 75, 66, -44, 74, 2, -36, -42, -103, 6,
  private TextView n;
  private TextView o;

  private String i()
  {
    int i = 0;
    byte[] arrayOfByte1 = new byte[p.length];
    for (int j = 0; j < arrayOfByte1.length; j++) {
      arrayOfByte1[j] = ((byte)(byte)(p[j] ^ q[j]));
    }
    int k = arrayOfByte1[0];
    for (j = 0; arrayOfByte1[(k + j)] != 0; j++) {}
    byte[] arrayOfByte2 = new byte[j];
    while (i < j)
    {
      arrayOfByte2[i] = ((byte)arrayOfByte1[(k + i)]);
      i++;
    }
    return new String(arrayOfByte2);
  }

  public void onClickTest(View paramView)
  {
    if (this.n.getText().toString().equals(i())) {
      this.o.setText(2131099685);
    }
    for (;;)
    {
      return;
      this.o.setText(2131099683);
    }
  }

  protected void onCreate(Bundle paramBundle)
  {
    super.onCreate(paramBundle);
    setContentView(2130968602);
    this.n = ((TextView)findViewById(2131427413));
    this.o = ((TextView)findViewById(2131427415));
  }
}
```

那我们就先放到安卓模拟器运行一下看看

输入123456789

发现Wrong Key



破解密码 Crack the key

猜密码 Guess the key

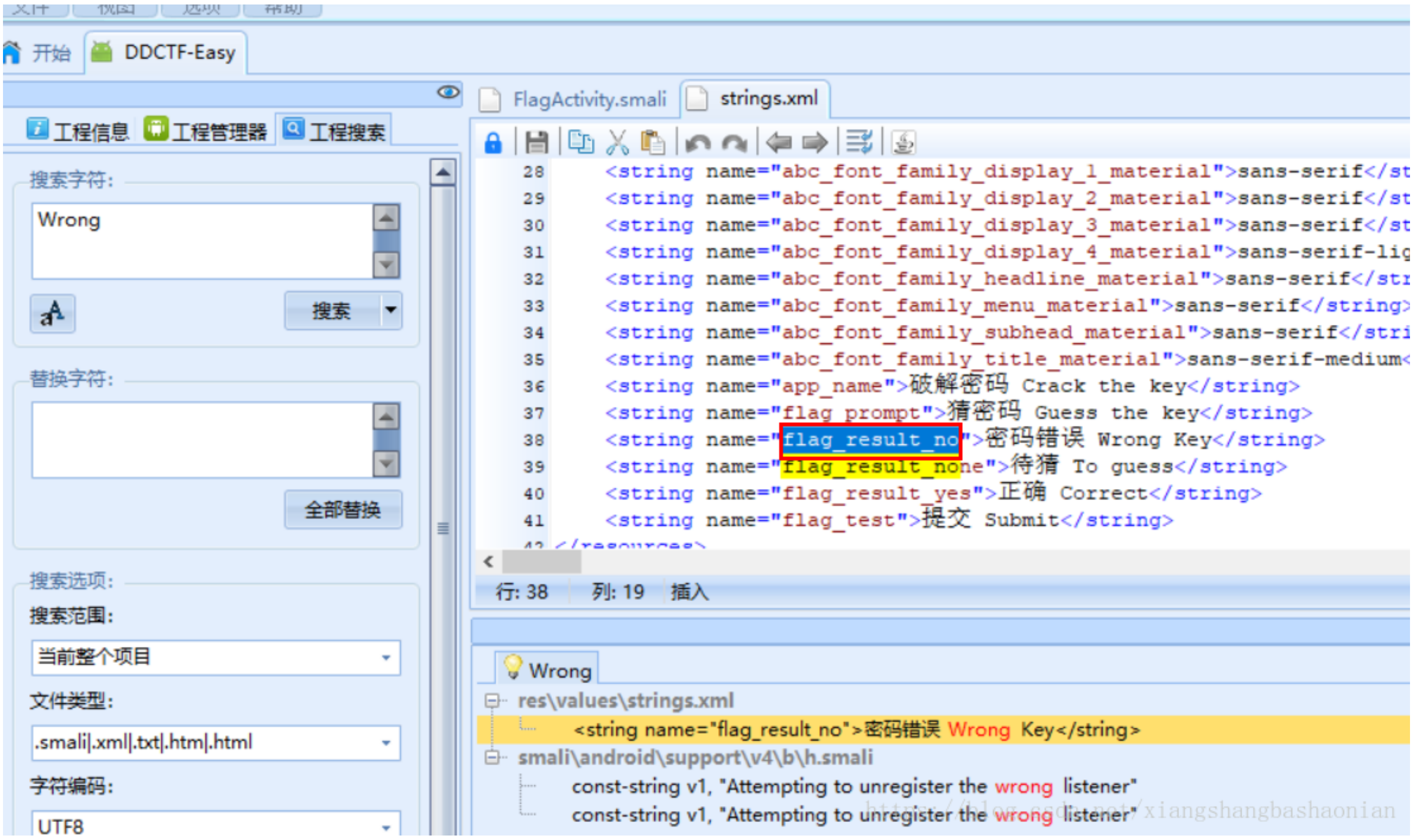
密码错误 Wrong Key

载入Androidkiller

第一步：搜索Wrong

第二步：搜索flag_result_no

第三步：搜索0x7f060023



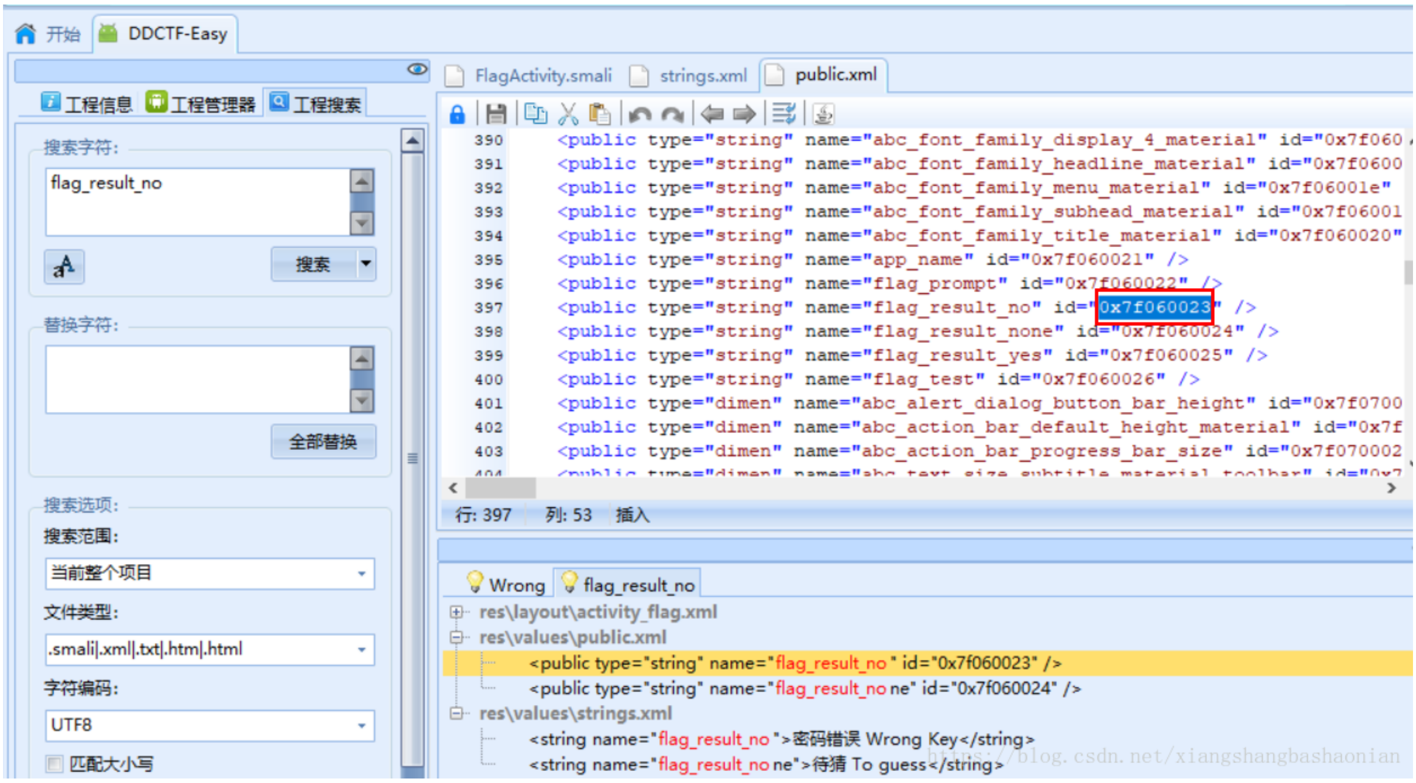
```
<string name="abc_font_family_display_1_material">sans-serif</st
<string name="abc_font_family_display_2_material">sans-serif</st
<string name="abc_font_family_display_3_material">sans-serif</st
<string name="abc_font_family_display_4_material">sans-serif-lig
<string name="abc_font_family_headline_material">sans-serif</str
<string name="abc_font_family_menu_material">sans-serif</string>
<string name="abc_font_family_subhead_material">sans-serif</stri
<string name="abc_font_family_title_material">sans-serif-medium<
<string name="app_name">破解密码 Crack the key</string>
<string name="flag_prompt">猜密码 Guess the key</string>
<string name="flag_result_no">密码错误 Wrong Key</string>
<string name="flag_result_none">待猜 To guess</string>
<string name="flag_result_yes">正确 Correct</string>
<string name="flag_test">提交 Submit</string>
</resources>
```

res\values\strings.xml

```
<string name="flag_result_no">密码错误 Wrong Key</string>
```

smali\android\support\v4\b\h.smali

```
const-string v1, "Attempting to unregister the wrong listener"
const-string v1, "Attempting to unregister the wrong listener"
```



```
<public type="string" name="abc_font_family_display_4_material" id="0x7f060
<public type="string" name="abc_font_family_headline_material" id="0x7f0600
<public type="string" name="abc_font_family_menu_material" id="0x7f06001e"
<public type="string" name="abc_font_family_subhead_material" id="0x7f06001
<public type="string" name="abc_font_family_title_material" id="0x7f060020"
<public type="string" name="app_name" id="0x7f060021" />
<public type="string" name="flag_prompt" id="0x7f060022" />
<public type="string" name="flag_result_no" id="0x7f060023" />
<public type="string" name="flag_result_none" id="0x7f060024" />
<public type="string" name="flag_result_yes" id="0x7f060025" />
<public type="string" name="flag_test" id="0x7f060026" />
<public type="dimen" name="abc_alert_dialog_button_bar_height" id="0x7f0700
<public type="dimen" name="abc_action_bar_default_height_material" id="0x7f
<public type="dimen" name="abc_action_bar_progress_bar_size" id="0x7f070002
<public type="dimen" name="abc_text_size_subtitle_material_toolbar" id="0x7
```

res\layout\activity_flag.xml

res\values\public.xml

```
<public type="string" name="flag_result_no" id="0x7f060023" />
<public type="string" name="flag_result_none" id="0x7f060024" />
```

res\values\strings.xml

```
<string name="flag_result_no">密码错误 Wrong Key</string>
<string name="flag_result_none">待猜 To guess</string>
```

得到三个好玩的

发现0x7f060025对应yes 猜测这个id对应的是正确的路

```
<public type="string" name="flag_result_no" id="0x7f060023" />
```

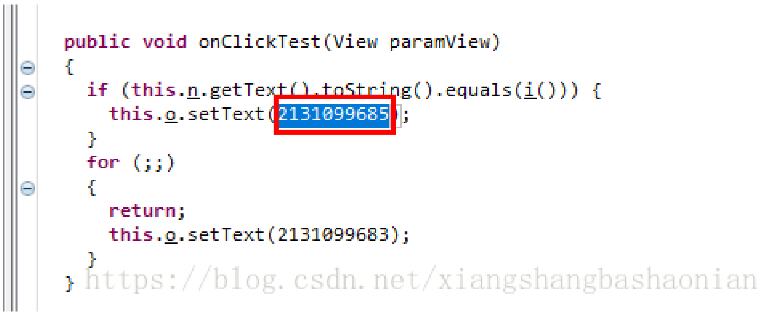
<public type="string" name="flag_result_none" id="0x7f060024" />

<public type="string" name="flag_result_yes" id="0x7f060025" />

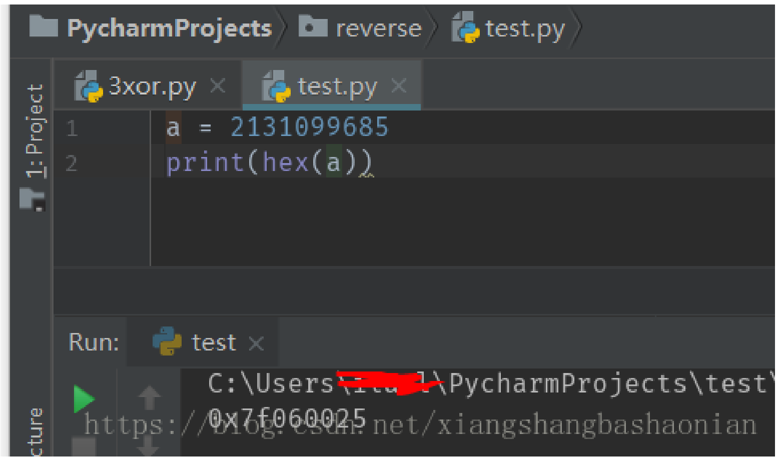
这次对照着这个类　就可以很明白了

将2131099685转成十六进制　得到0x7f060025　即yes

那么我们就可以知道关键就在i()这个方法里

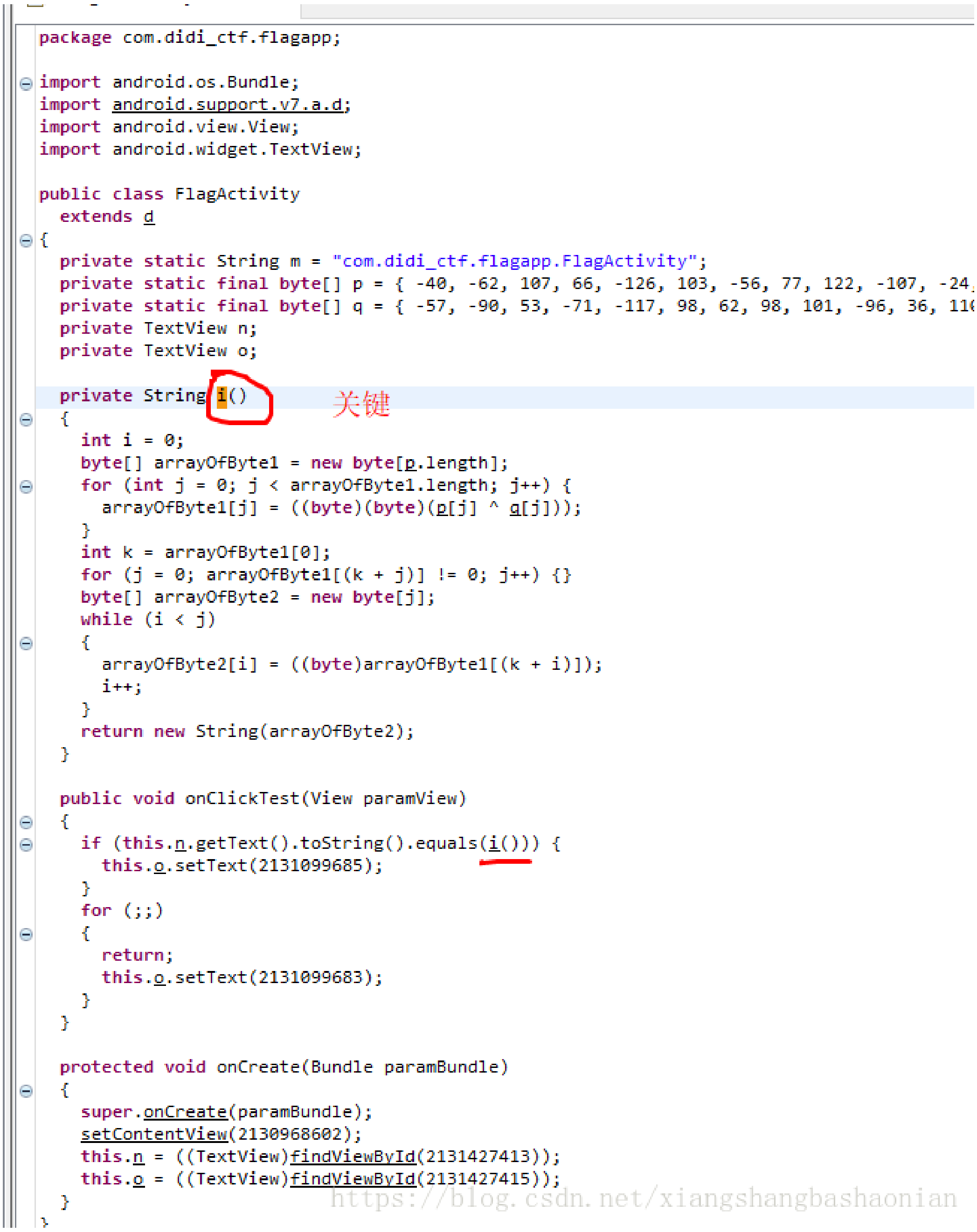


```
public void onClickTest(View paramView)
{
  if (this.n.getText().toString().equals(i())) {
    this.o.setText(2131099685);
  }
  for (;;)
  {
    return;
    this.o.setText(2131099683);
  }
}
```



```
a = 2131099685
print(hex(a))
```

```
C:\Users\[redacted]\PycharmProjects\test
0x7f060025
```



```
package com.didi_ctf.flagapp;

import android.os.Bundle;
import android.support.v7.a.d;
import android.view.View;
import android.widget.TextView;

public class FlagActivity
  extends d
{
  private static String m = "com.didi_ctf.flagapp.FlagActivity";
  private static final byte[] p = { -40, -62, 107, 66, -126, 103, -56, 77, 122, -107, -24,
  private static final byte[] q = { -57, -90, 53, -71, -117, 98, 62, 98, 101, -96, 36, 11
  private TextView n;
  private TextView o;

  private String i()
  {
    int i = 0;
    byte[] arrayOfByte1 = new byte[p.length];
    for (int j = 0; j < arrayOfByte1.length; j++) {
      arrayOfByte1[j] = ((byte)(byte)(p[j] ^ q[j]));
    }
    int k = arrayOfByte1[0];
    for (j = 0; arrayOfByte1[(k + j)] != 0; j++) {}
    byte[] arrayOfByte2 = new byte[j];
    while (i < j)
    {
      arrayOfByte2[i] = ((byte)arrayOfByte1[(k + i)]);
      i++;
    }
    return new String(arrayOfByte2);
  }

  public void onClickTest(View paramView)
  {
    if (this.n.getText().toString().equals(i())) {
      this.o.setText(2131099685);
    }
    for (;;)
    {
      return;
      this.o.setText(2131099683);
    }
  }

  protected void onCreate(Bundle paramBundle)
  {
    super.onCreate(paramBundle);
    setContentView(2130968602);
    this.n = ((TextView)findViewById(2131427413));
    this.o = ((TextView)findViewById(2131427415));
  }
}
```

那就对i()方法进行分析

大致意思就是说先创建一个数组a1　令长度与数组p一样

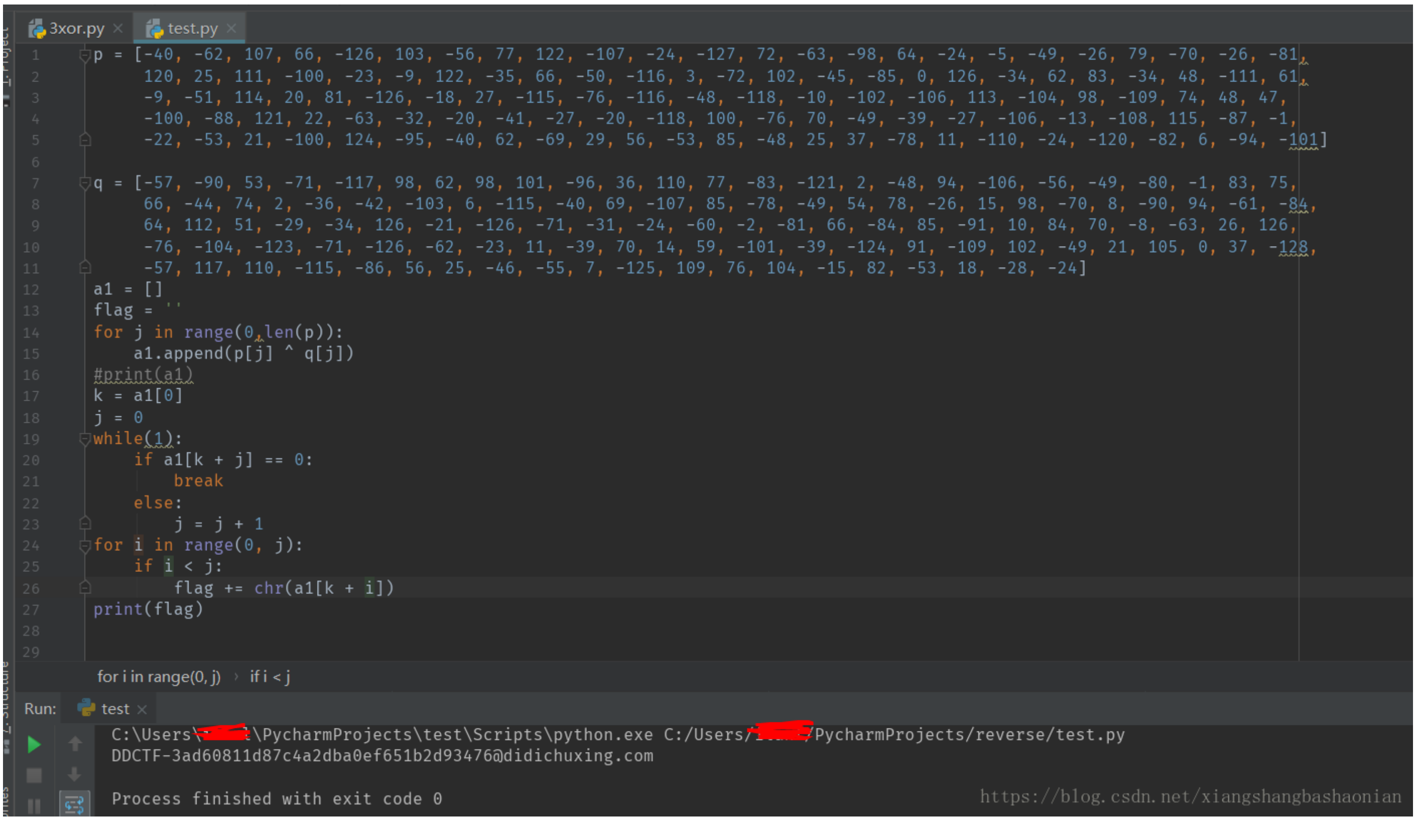
for循环使a1[i] = p[i] ^ q[i](每位异或赋值给a1的每位)

接着令k 等于a1[0]

由a1[k + j] != 0得到j

最后再来一个循环赋值得到flag

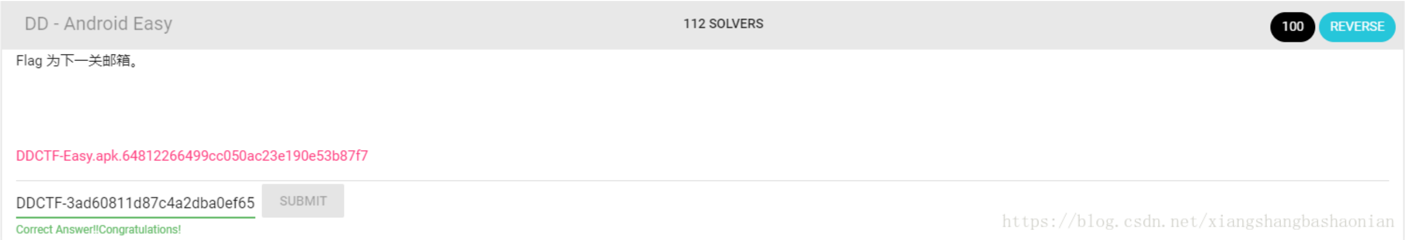
具体直接看py代码：



```python
p = [-40, -62, 107, 66, -126, 103, -56, 77, 122, -107, -24, -127, 72, -63, -98, 64, -24, -5, -49, -26, 79, -70, -26, -81,
     120, 25, 111, -100, -23, -9, 122, -35, 66, -50, -116, 3, -72, 102, -45, -85, 0, 126, -34, 62, 83, -34, 48, -111, 61,
     -9, -51, 114, 20, 81, -126, -18, 27, -115, -76, -116, -48, -118, -10, -102, -106, 113, -104, 98, -109, 74, 48, 47,
     -100, -88, 121, 22, -63, -32, -20, -41, -27, -20, -118, 100, -76, 70, -49, -39, -27, -106, -13, -108, 115, -87, -1,
     -22, -53, 21, -100, 124, -95, -40, 62, -69, 29, 56, -53, 85, -48, 25, 37, -78, 11, -110, -24, -120, -82, 6, -94, -101]

q = [-57, -90, 53, -71, -117, 98, 62, 98, 101, -96, 36, 110, 77, -83, -121, 2, -48, 94, -106, -56, -49, -80, -1, 83, 75,
     66, -44, 74, 2, -36, -42, -103, 6, -115, -40, 69, -107, 85, -78, -49, 54, 78, -26, 15, 98, -70, 8, -90, 94, -61, -84,
     64, 112, 51, -29, -34, 126, -21, -126, -71, -31, -24, -60, -2, -81, 66, -84, 85, -91, 10, 84, 70, -8, -63, 26, 126,
     -76, -104, -123, -71, -126, -62, -23, 11, -39, 70, 14, 59, -101, -39, -124, 91, -109, 102, -49, 21, 105, 0, 37, -128,
     -57, 117, 110, -115, -86, 56, 25, -46, -55, 7, -125, 109, 76, 104, -15, 82, -53, 18, -28, -24]
a1 = []
flag = ''
for j in range(0,len(p)):
    a1.append(p[j] ^ q[j])
#print(a1)
k = a1[0]
j = 0
while(1):
    if a1[k + j] == 0:
        break
    else:
        j = j + 1
for i in range(0, j):
    if i < j:
        flag += chr(a1[k + i])
print(flag)
```

```
C:\Users\[redacted]\PycharmProjects\test\Scripts\python.exe C:/Users/[redacted]/PycharmProjects/reverse/test.py
DDCTF-3ad60811d87c4a2dba0ef651b2d93476@didichuxing.com

Process finished with exit code 0
```

验证一下　成功！



Flag 为下一关邮箱。

DDCTF-Easy.apk.64812266499cc050ac23e190e53b87f7

DDCTF-3ad60811d87c4a2dba0ef65

Correct Answer!!Congratulations!