CTF apk 安卓逆向

原创

5wimming 于 2020-11-01 17:51:26 发布 2133 收藏 3

分类专栏： 安全 文章标签： CTF apk

本文链接：https://blog.csdn.net/qq_34101364/article/details/109395396

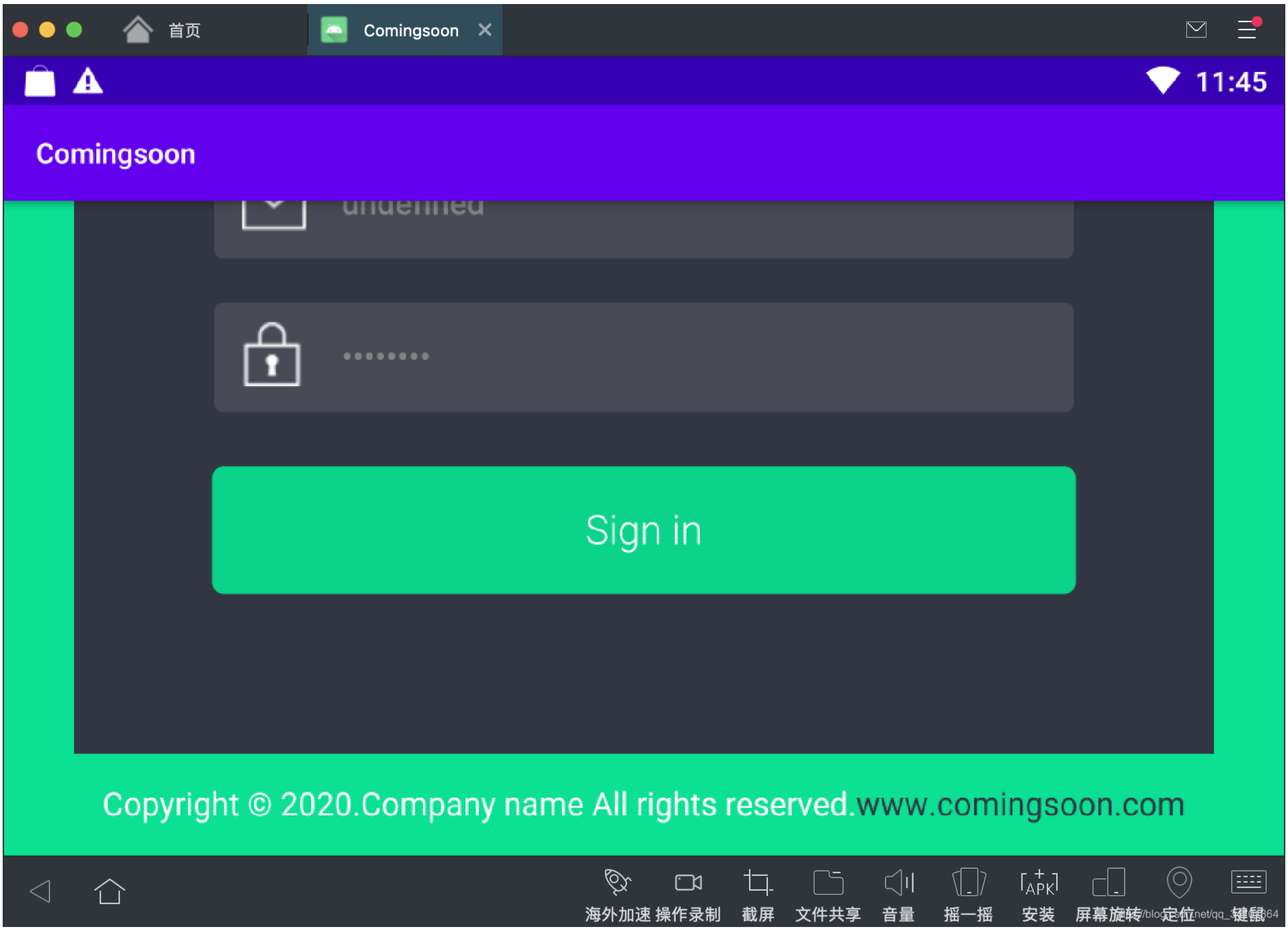
版权

安全 专栏收录该内容

53 篇文章 6 订阅

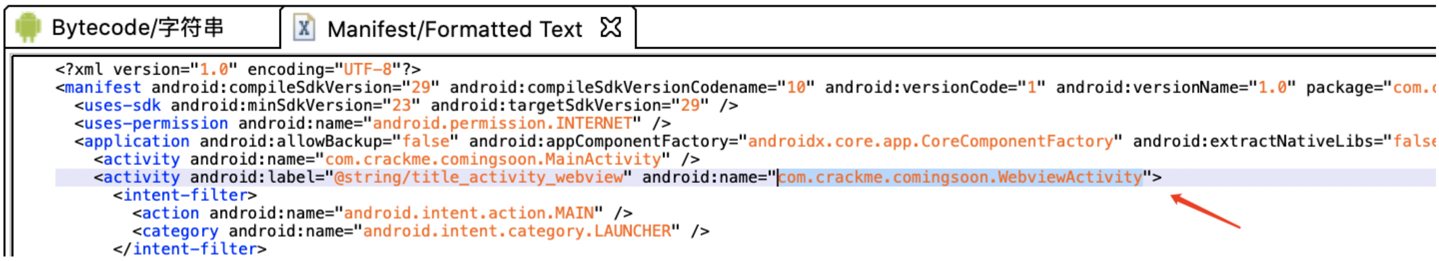
订阅专栏

apk界面如下，看一看其实没有什么用。。。

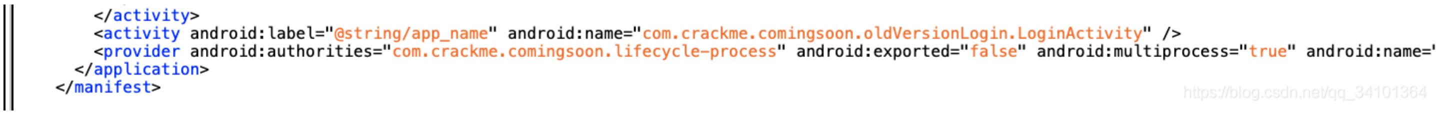


Copyright © 2020.Company name All rights reserved.www.comingsoon.com

1、使用jeb对apk进行分析，找到manifest配置文件（即应用清单，中包含了APP的配置信息，系统需要根据里面的内容运行APP的代码，显示界面），从中找到初始启动类com.crackme.comingsoon.WebviewActivity

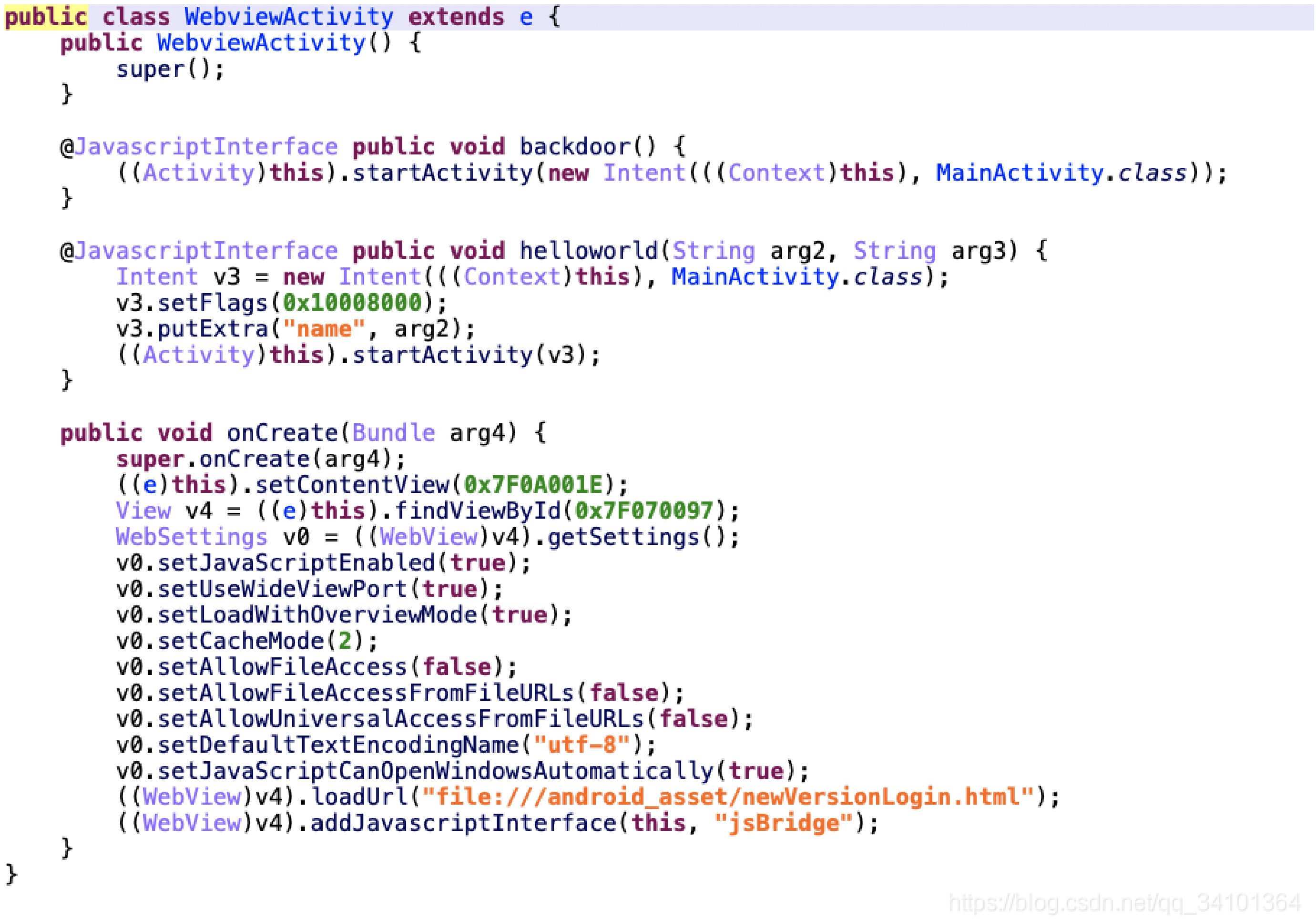


```
<?xml version="1.0" encoding="UTF-8"?>
<manifest android:compileSdkVersion="29" android:compileSdkVersionCodename="10" android:versionCode="1" android:versionName="1.0" package="com.c
  <uses-sdk android:minSdkVersion="23" android:targetSdkVersion="29" />
  <uses-permission android:name="android.permission.INTERNET" />
  <application android:allowBackup="false" android:appComponentFactory="androidx.core.app.CoreComponentFactory" android:extractNativeLibs="false
    <activity android:name="com.crackme.comingsoon.MainActivity" />
    <activity android:label="@string/title_activity_webview" android:name="com.crackme.comingsoon.WebviewActivity">
      <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
      </intent-filter>
```



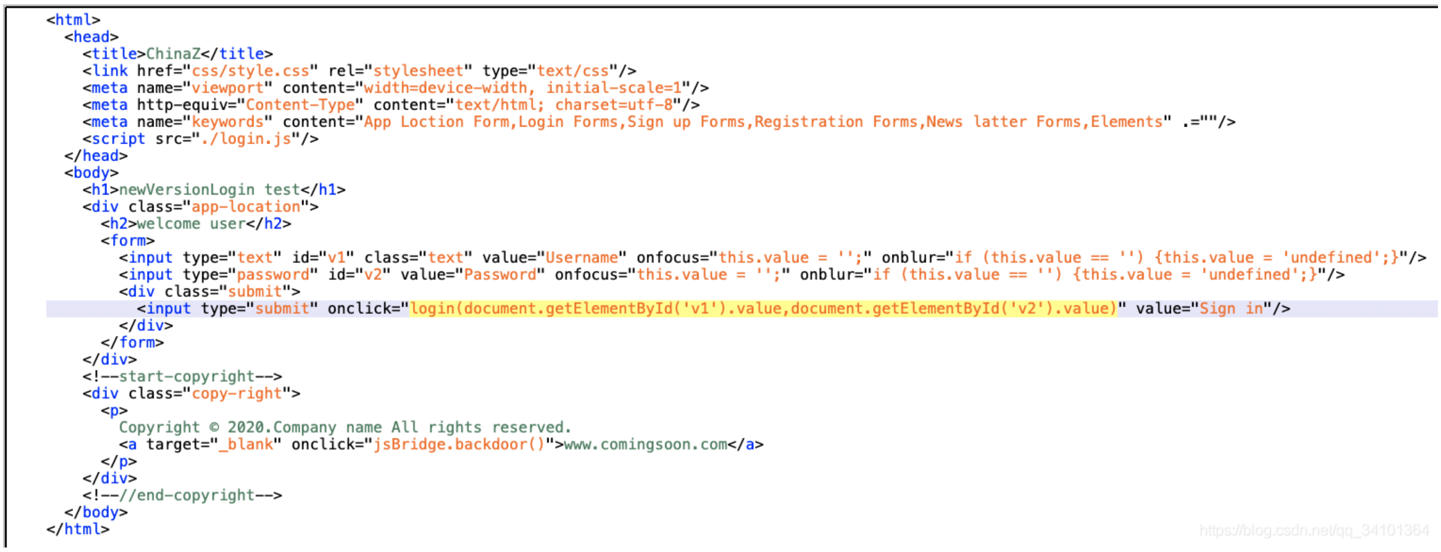
```
      </activity>
      <activity android:label="@string/app_name" android:name="com.crackme.comingsoon.oldVersionLogin.LoginActivity" />
      <provider android:authorities="com.crackme.comingsoon.lifecycle-process" android:exported="false" android:multiprocess="true" android:name='
    </application>
  </manifest>
```

2、WebviewActivity中注册了backdoor和helloworld两个函数，并加载了newVersionLogin.html进行渲染。



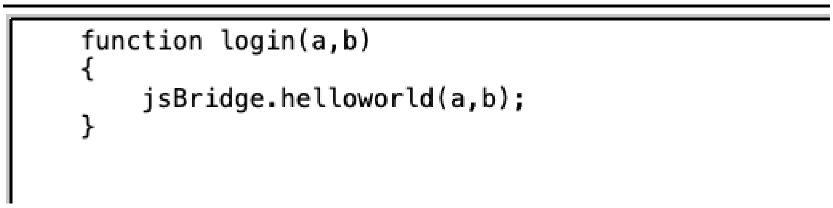
```
public class WebviewActivity extends e {
    public WebviewActivity() {
        super();
    }

    @JavascriptInterface public void backdoor() {
        ((Activity)this).startActivity(new Intent(((Context)this), MainActivity.class));
    }

    @JavascriptInterface public void helloworld(String arg2, String arg3) {
        Intent v3 = new Intent(((Context)this), MainActivity.class);
        v3.setFlags(0x10008000);
        v3.putExtra("name", arg2);
        ((Activity)this).startActivity(v3);
    }

    public void onCreate(Bundle arg4) {
        super.onCreate(arg4);
        ((e)this).setContentView(0x7F0A001E);
        View v4 = ((e)this).findViewById(0x7F070097);
        WebSettings v0 = ((WebView)v4).getSettings();
        v0.setJavaScriptEnabled(true);
        v0.setUseWideViewPort(true);
        v0.setLoadWithOverviewMode(true);
        v0.setCacheMode(2);
        v0.setAllowFileAccess(false);
        v0.setAllowFileAccessFromFileURLs(false);
        v0.setAllowUniversalAccessFromFileURLs(false);
        v0.setDefaultTextEncodingName("utf-8");
        v0.setJavaScriptCanOpenWindowsAutomatically(true);
        ((WebView)v4).loadUrl("file:///android_asset/newVersionLogin.html");
        ((WebView)v4).addJavascriptInterface(this, "jsBridge");
    }
}
```

3、newVersionLogin.html中可以访问两个函数，js的login函数，和后台的backdoor函数



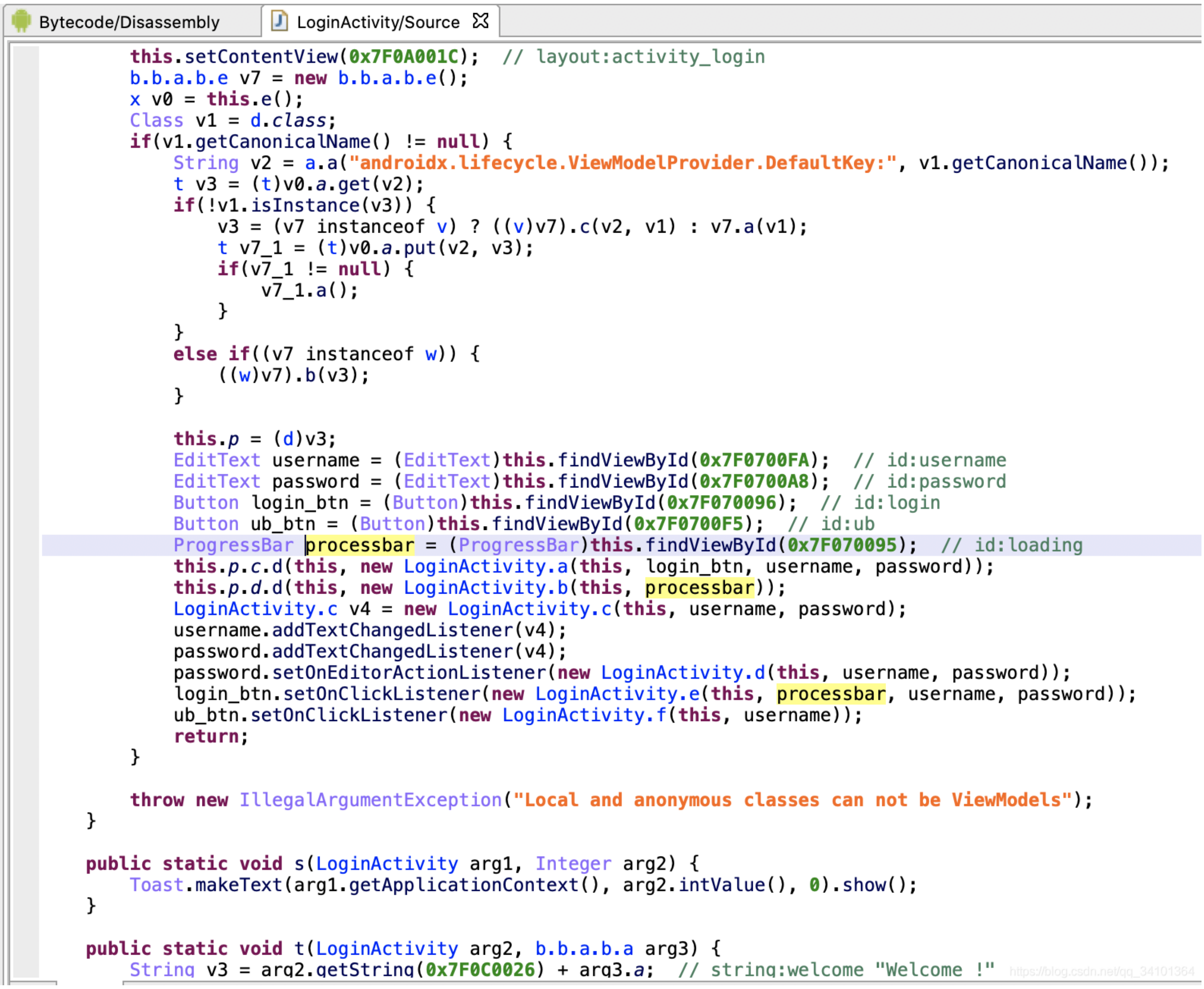
```
<html>
  <head>
    <title>ChinaZ</title>
    <link href="css/style.css" rel="stylesheet" type="text/css"/>
    <meta name="viewport" content="width=device-width, initial-scale=1"/>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>
    <meta name="keywords" content="App Loction Form,Login Forms,Sign up Forms,Registration Forms,News latter Forms,Elements" .=""/>
    <script src="./login.js"/>
  </head>
  <body>
    <h1>newVersionLogin test</h1>
    <div class="app-location">
      <h2>welcome user</h2>
      <form>
        <input type="text" id="v1" class="text" value="Username" onfocus="this.value = '';" onblur="if (this.value == '') {this.value = 'undefined';}"/>
        <input type="password" id="v2" value="Password" onfocus="this.value = '';" onblur="if (this.value == '') {this.value = 'undefined';}"/>
        <div class="submit">
          <input type="submit" onclick="login(document.getElementById('v1').value,document.getElementById('v2').value)" value="Sign in"/>
        </div>
      </form>
    </div>
    <!--start-copyright-->
    <div class="copy-right">
      <p>
        Copyright © 2020.Company name All rights reserved.
        <a target="_blank" onclick="jsBridge.backdoor()">www.comingsoon.com</a>
      </p>
    </div>
    <!--//end-copyright-->
  </body>
</html>
```

login.js如下，即访问后台的helloworld函数



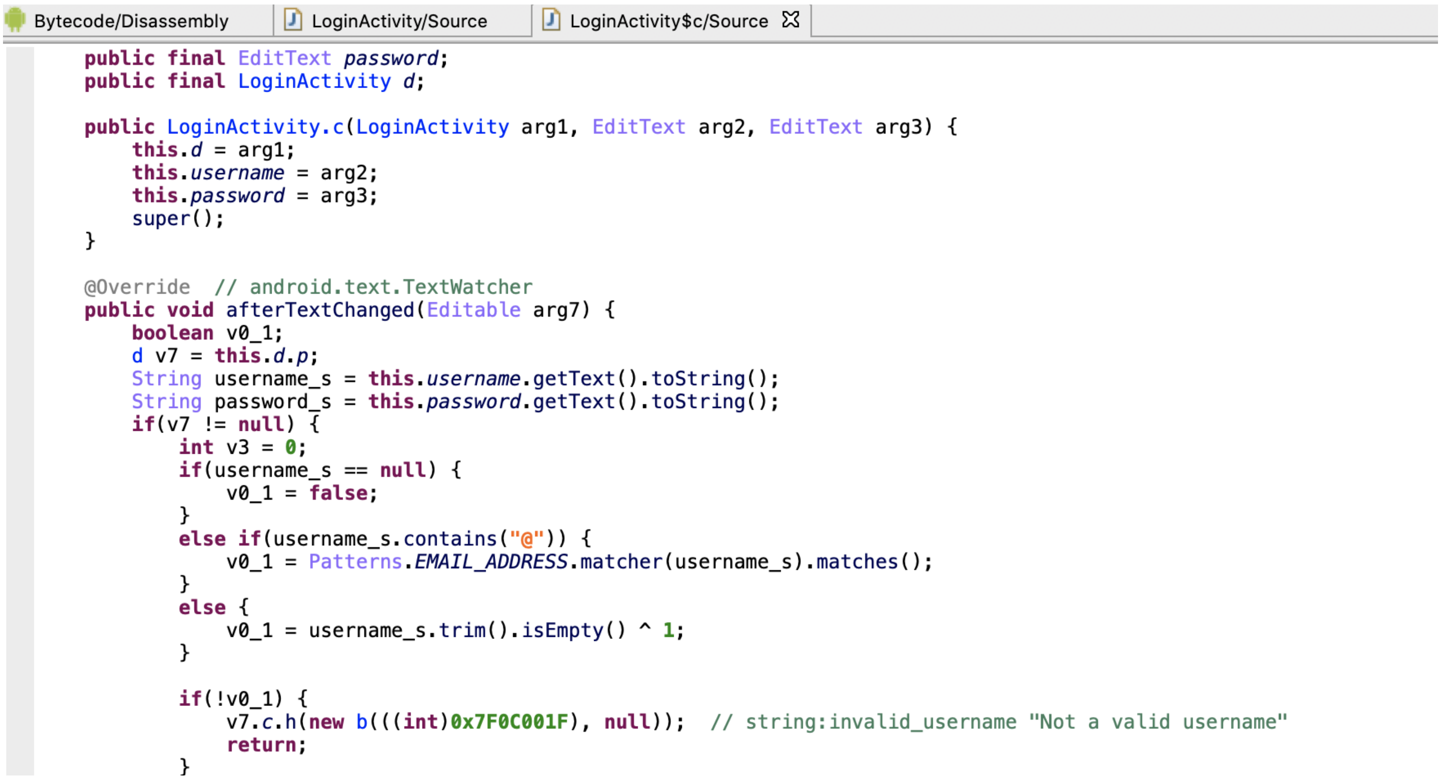
```
function login(a,b)
{
    jsBridge.helloworld(a,b);
}
```

4、来看看LoginActivity的逻辑，根据备注对变量进行重命名，方便查看。

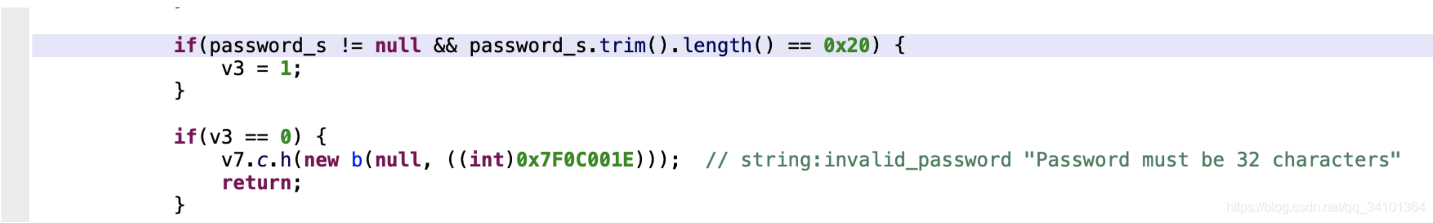


```
        this.setContentView(0x7F0A001C);  // layout:activity_login
        b.b.a.b.e v7 = new b.b.a.b.e();
        x v0 = this.e();
        Class v1 = d.class;
        if(v1.getCanonicalName() != null) {
            String v2 = a.a("androidx.lifecycle.ViewModelProvider.DefaultKey:", v1.getCanonicalName());
            t v3 = (t)v0.a.get(v2);
            if(!v1.isInstance(v3)) {
                v3 = (v7 instanceof v) ? ((v)v7).c(v2, v1) : v7.a(v1);
                t v7_1 = (t)v0.a.put(v2, v3);
                if(v7_1 != null) {
                    v7_1.a();
                }
            }
            else if((v7 instanceof w)) {
                ((w)v7).b(v3);
            }

            this.p = (d)v3;
            EditText username = (EditText)this.findViewById(0x7F0700FA);  // id:username
            EditText password = (EditText)this.findViewById(0x7F0700A8);  // id:password
            Button login_btn = (Button)this.findViewById(0x7F070096);  // id:login
            Button ub_btn = (Button)this.findViewById(0x7F0700F5);  // id:ub
            ProgressBar processbar = (ProgressBar)this.findViewById(0x7F070095);  // id:loading
            this.p.c.d(this, new LoginActivity.a(this, login_btn, username, password));
            this.p.d.d(this, new LoginActivity.b(this, processbar));
            LoginActivity.c v4 = new LoginActivity.c(this, username, password);
            username.addTextChangedListener(v4);
            password.addTextChangedListener(v4);
            password.setOnEditorActionListener(new LoginActivity.d(this, username, password));
            login_btn.setOnClickListener(new LoginActivity.e(this, processbar, username, password));
            ub_btn.setOnClickListener(new LoginActivity.f(this, username));
            return;
        }

        throw new IllegalArgumentException("Local and anonymous classes can not be ViewModels");
    }

    public static void s(LoginActivity arg1, Integer arg2) {
        Toast.makeText(arg1.getApplicationContext(), arg2.intValue(), 0).show();
    }

    public static void t(LoginActivity arg2, b.b.a.b.a arg3) {
        String v3 = arg2.getString(0x7F0C0026) + arg3.a;  // string:welcome "Welcome !"
```

进入相关函数，发现会对账号密码进行检查，其中密码长度为32

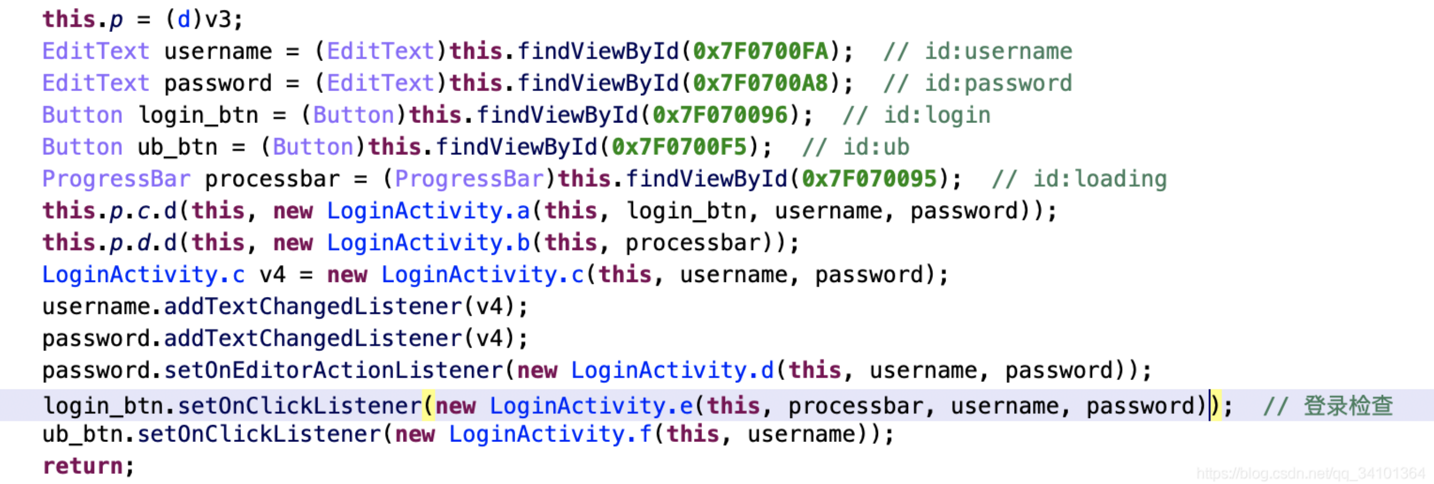


```
    public final EditText password;
    public final LoginActivity d;

    public LoginActivity.c(LoginActivity arg1, EditText arg2, EditText arg3) {
        this.d = arg1;
        this.username = arg2;
        this.password = arg3;
        super();
    }

    @Override  // android.text.TextWatcher
    public void afterTextChanged(Editable arg7) {
        boolean v0_1;
        d v7 = this.d.p;
        String username_s = this.username.getText().toString();
        String password_s = this.password.getText().toString();
        if(v7 != null) {
            int v3 = 0;
            if(username_s == null) {
                v0_1 = false;
            }
            else if(username_s.contains("@")) {
                v0_1 = Patterns.EMAIL_ADDRESS.matcher(username_s).matches();
            }
            else {
                v0_1 = username_s.trim().isEmpty() ^ 1;
            }

            if(!v0_1) {
                v7.c.h(new b(((int)0x7F0C001F), null));  // string:invalid_username "Not a valid username"
                return;
            }
```



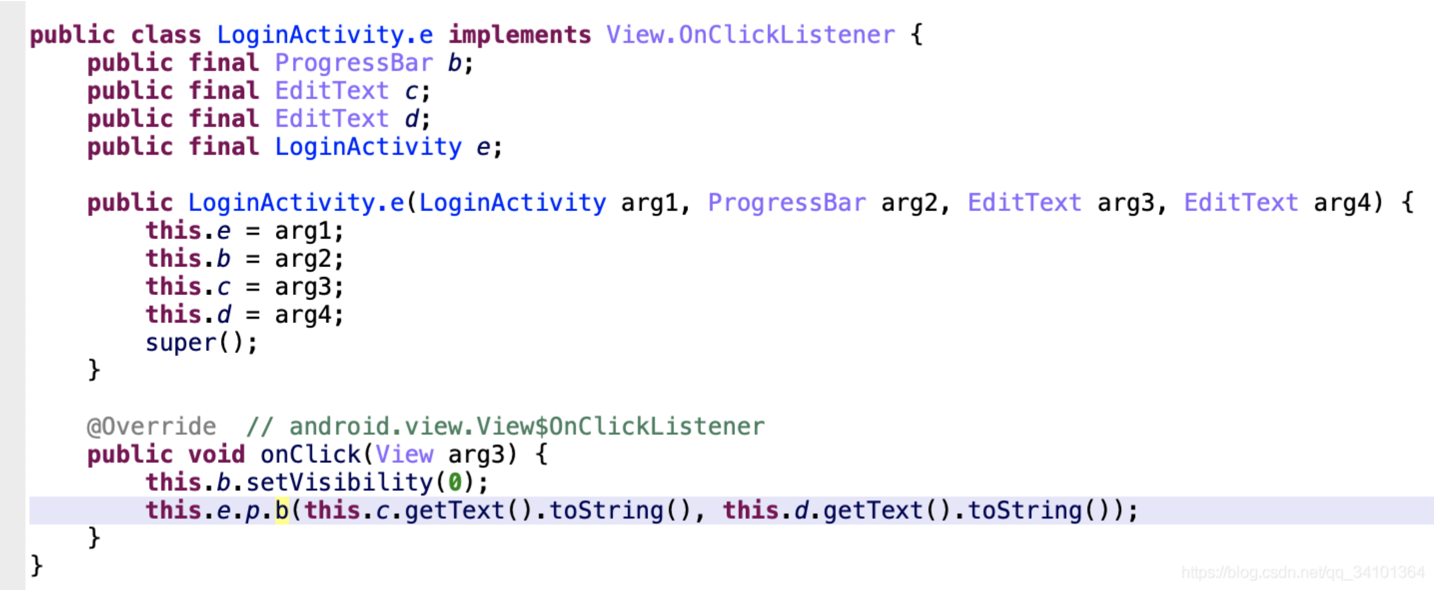
```
            if(password_s != null && password_s.trim().length() == 0x20) {
                v3 = 1;
            }

            if(v3 == 0) {
                v7.c.h(new b(null, ((int)0x7F0C001E)));  // string:invalid_password "Password must be 32 characters"
                return;
            }
```

5、返回LoginActivity，LoginActivity.e会进行登录检查，继续跟进



```
this.p = (d)v3;
EditText username = (EditText)this.findViewById(0x7F0700FA);  // id:username
EditText password = (EditText)this.findViewById(0x7F0700A8);  // id:password
Button login_btn = (Button)this.findViewById(0x7F070096);  // id:login
Button ub_btn = (Button)this.findViewById(0x7F0700F5);  // id:ub
ProgressBar processbar = (ProgressBar)this.findViewById(0x7F070095);  // id:loading
this.p.c.d(this, new LoginActivity.a(this, login_btn, username, password));
this.p.d.d(this, new LoginActivity.b(this, processbar));
LoginActivity.c v4 = new LoginActivity.c(this, username, password);
username.addTextChangedListener(v4);
password.addTextChangedListener(v4);
password.setOnEditorActionListener(new LoginActivity.d(this, username, password));
login_btn.setOnClickListener(new LoginActivity.e(this, processbar, username, password));  // 登录检查
ub_btn.setOnClickListener(new LoginActivity.f(this, username));
return;
```

进入LoginActivity.e函数，有用的是e.p.b函数，继续跟进



```
public class LoginActivity.e implements View.OnClickListener {
    public final ProgressBar b;
    public final EditText c;
    public final EditText d;
    public final LoginActivity e;

    public LoginActivity.e(LoginActivity arg1, ProgressBar arg2, EditText arg3, EditText arg4) {
        this.e = arg1;
        this.b = arg2;
        this.c = arg3;
        this.d = arg4;
        super();
    }

    @Override  // android.view.View$OnClickListener
    public void onClick(View arg3) {
        this.b.setVisibility(0);
        this.e.p.b(this.c.getText().toString(), this.d.getText().toString());
    }
}
```

下面是e.p.b函数，参数arg4为password，首先会进入v3.a进行判断，跟进v3.a函数

```
public void b(String arg3, String arg4) {
        c v4;
        a v3 = this.e.a;
        if(v3 != null) {
            try {
                v4 = v3.a(arg4) ? new c(new b.b.a.a.d.a(UUID.randomUUID().toString(), "for(int i=0;i<38;i++)\n{\n\ttable[i] = (int)flag.charAt(i)-(int)decode.charAt((base+i)%decode.length());\n}\n//int table[] = {-6, 10, 25, -5, 4, 25, 36, 5, -2, 10, 47, 2, -17, 17, 39, 48, 14, 0, 43, 55, 50, 22, 62, -20, -22, 19, -9, -3, 10, 13, 58, 29, 1, 2, 38, -10, 1, 35};")) : new b.b.a.a.c.b(new IOException("Permission deny"));
            }
            catch(Exception v3_1) {
                v4 = new b.b.a.a.c.b(new IOException("Error logging in", v3_1));
            }

            if((v4 instanceof c)) {
                this.d.h(new b.b.a.b.c(new b.b.a.b.a(((b.b.a.a.d.a)v4.a).a)));
                return;
            }

            this.d.h(new b.b.a.b.c(((int)0x7F0C0020)));  // string:login_failed "Login failed"
            return;
        }

        throw null;
    }
// v3.a函数
public boolean a(String arg15) {
        if(arg15.substring(arg15.length() - 2, arg15.length()).equals("==")) {
            byte[] v15 = Base64.decode(arg15.getBytes(), 0);
            return (v15[0] ^ v15.length << 12) != 90227 || v15[v15.length - 1] + v15[1] != 0xE4 || v15[1] + v15[2] + v15[v15.length - 3] + v15[v15.length - 16] + v15[15] + v15[13] + v15[v15.length - 14] + v15[10] + v15[v15.length - 11] + v15[v15.length - 6] != 0x45C || v15[1] - v15[2] + v15[v15.length - 3] != 104 || v15[v15.length - 6] * 2 + (v15[v15.length - 16] * 4 + v15[15] - v15[v15.length - 3]) != 650 || v15[v15.length - 11] * 13 + (v15[13] - v15[v15.length - 14] - v15[10] * 19) - v15[1] * 3 != 0xFFFFFB83 || v15[v15.length - 14] * 12 + (v15[2] * 10 + v15[15] * 9 - v15[10] * 11) - v15[v15.length - 6] * 13 != 661 || v15[v15.length - 14] - v15[13] + v15[v15.length - 16] - v15[v15.length - 3] + v15[2] - v15[1] != -15 || v15[v15.length - 11] * 3 + (v15[v15.length - 14] * 3 + (v15[15] * 3 + v15[v15.length - 3] * 3)) != 0x525 || v15[10] * 7 + v15[v15.length - 14] * 5 - v15[v15.length - 11] * 9 != 344 || v15[v15.length - 11] * 5 + (v15[v15.length - 14] * 4 + (v15[15] * 3 + (v15[v15.length - 3] * 2 + v15[1]))) != 1640 || v15[v15.length - 16] * 7 + (v15[10] * 4 + -v15[v15.length - 6] - v15[13] * 9) != 54 || v15[v15.length - 8] * v15[v15.length - 8] - v15[7] * 108 != 973 || v15[v15.length - 2] - v15[v15.length - 8] != -11 || v15[7] + v15[v15.length - 2] != 0xC7 || v15[3] * v15[4] * v15[5] != 0x15BF34 || v15[3] * v15[4] - v15[5] != 0x37D9 || v15[4] * v15[4] - v15[3] * v15[5] != 3202 || v15[6] * v15[6] * v15[v15.length - 5] + v15[14] * v15[v15.length - 13] != 0x1338C1 || v15[8] * v15[v15.length - 10] + v15[v15.length - 5] + v15[v15.length - 4] != 10309 || v15[6] * v15[v15.length - 5] != 0x2D8F || v15[6] * v15[v15.length - 13] != 0x2D24 || v15[14] * v15[v15.length - 10] - v15[6] * v15[v15.length - 4] != 200 ? 0 : 1;
        }

        return 0;
    }
```

6、接下来重点分析上面的v3.a函数，从函数中我们可以知道：

条件一：输入以“==”结尾的base64字符串，并且password的长度为32，根据base64的规则，解码后的v15长度为22位：

```
32*6/8=24
24-2=22 //有几个=号，减去多少
```

条件二：(v15[0] ^ v15.length << 12) == 90227

```
v15[0] = 90227 ^ 22<<12 = 115
```

条件三：多元一次方程：

```
v15[22-1] + v15[1] == 0xE4
v15[1] + v15[2] + v15[22- 3] + v15[22- 16] + v15[15] + v15[13] + v15[22- 14] + v15[10] + v15[22-11] + v15[22-6] = 0x45C
v15[1] - v15[2] + v15[22- 3] = 104
v15[22- 6] * 2 + (v15[22- 16] * 4 + v15[15] - v15[22- 3]) = 650
v15[22- 11] * 13 + (v15[13] - v15[22- 14] - v15[10] * 19) - v15[1] * 3 = 0xFFFFFB83
v15[22- 14] * 12 + (v15[2] * 10 + v15[15] * 9 - v15[10] * 11) - v15[22- 6] * 13 = 661
v15[22- 14] - v15[13] + v15[22- 16] - v15[22- 3] + v15[2] - v15[1] = -15
v15[22- 11] * 3 + (v15[22- 14] * 3 + (v15[15] * 3 + v15[22- 3] * 3)) = 0x525
v15[10] * 7 + v15[22- 14] * 5 - v15[22- 11] * 9 = 344
v15[22- 11] * 5 + (v15[22- 14] * 4 + (v15[15] * 3 + (v15[22- 3] * 2 + v15[1]))) = 1640
v15[22- 16] * 7 + (v15[10] * 4 + -v15[22- 6] - v15[13] * 9) = 54
v15[22- 8] * v15[22- 8] - v15[7] * 108 = 973
v15[22- 2] - v15[22- 8] = -11
v15[7] + v15[22- 2] = 0xC7
v15[3] * v15[4] * v15[5] = 0x15BF34
v15[3] * v15[4] - v15[5] = 0x37D9
v15[4] * v15[4] - v15[3] * v15[5] = 3202
v15[6] * v15[6] * v15[22- 5] + v15[14] * v15[22- 13] = 0x1338C1
v15[8] * v15[22- 10] + v15[22- 5] + v15[22- 4] = 10309
v15[6] * v15[22- 5] = 0x2D8F
v15[6] * v15[22- 13] = 0x2D24
v15[14] * v15[22- 10] - v15[6] * v15[22- 4] = 200
```

那就求解呗，这里使用python3，需要安装z3-solver库：

```python
# python3 -m pip install z3-solver
# 5wimming

import z3
import base64


def main():
    v15 = [z3.Int('x%d' % i) for i in range(22)]
    z3solver = z3.Solver()

    z3solver.add(v15[22 - 1] + v15[1] == 228)
    z3solver.add(v15[1] + v15[2] + v15[22 - 3] + v15[22 - 16] + v15[15] + v15[13] + v15[22 - 14] + v15[10] + v15[22 - 11] + v15[22 - 6] == 1116)
    z3solver.add(v15[1] - v15[2] + v15[22 - 3] == 104)
    z3solver.add(v15[22 - 6] * 2 + (v15[22 - 16] * 4 + v15[15] - v15[22 - 3]) == 650)
    z3solver.add(v15[22 - 11] * 13 + (v15[13] - v15[22 - 14] - v15[10] * 19) - v15[1] * 3 == -1149)
    z3solver.add(v15[22 - 14] * 12 + (v15[2] * 10 + v15[15] * 9 - v15[10] * 11) - v15[22 - 6] * 13 == 661)
    z3solver.add(v15[22 - 14] - v15[13] + v15[22 - 16] - v15[22 - 3] + v15[2] - v15[1] == -15)
    z3solver.add(v15[22 - 11] * 3 + (v15[22 - 14] * 3 + (v15[15] * 3 + v15[22 - 3] * 3)) == 1317)
    z3solver.add(v15[10] * 7 + v15[22 - 14] * 5 - v15[22 - 11] * 9 == 344)
    z3solver.add(v15[22 - 11] * 5 + (v15[22 - 14] * 4 + (v15[15] * 3 + (v15[22 - 3] * 2 + v15[1]))) == 1640)
    z3solver.add(v15[22 - 16] * 7 + (v15[10] * 4 + -v15[22 - 6] - v15[13] * 9) == 54)
    z3solver.add(v15[22 - 8] * v15[22 - 8] - v15[7] * 108 == 973)
    z3solver.add(v15[22 - 2] - v15[22 - 8] == -11)
    z3solver.add(v15[7] + v15[22 - 2] == 199)
    z3solver.add(v15[3] * v15[4] * v15[5] == 1425204)
    z3solver.add(v15[3] * v15[4] - v15[5] == 14297)
    z3solver.add(v15[4] * v15[4] - v15[3] * v15[5] == 3202)
    z3solver.add(v15[6] * v15[6] * v15[22 - 5] + v15[14] * v15[22 - 13] == 1259713)
    z3solver.add(v15[8] * v15[22 - 10] + v15[22 - 5] + v15[22 - 4] == 10309)
    z3solver.add(v15[6] * v15[22 - 5] == 11663)
    z3solver.add(v15[6] * v15[22 - 13] == 11556)
    z3solver.add(v15[14] * v15[22 - 10] - v15[6] * v15[22 - 4] == 200)

    z3solver.check()
    model = z3solver.model()

    result = [115]
    for i in v15[1:]:
        result.append(model.eval(i).as_long())
    print(result)

    str_result = ""
    for i in result:
        str_result += chr(i)
    print('origin result', str_result)

    base64_result = base64.b64encode(str_result.encode())
    print('base64 result', base64_result)


if __name__ == '__main__':
    main()
```

求出结果：

```
[115, 108, 114, 118, 122, 99, 107, 101, 101, 108, 121, 112, 100, 119, 109, 116, 108, 109, 100, 110, 98, 120]
origin result:  slrvzckeelypdwmtlmdnbx
base64 result:  b'c2xydnpja2VlbHlwZHdtdGxtZG5ieA=='
```

7、继续看接下来的函数，字符串里面就是将flag和table进行运算，得到password，所以根据该函数可以倒推flag

```
v4 = v3.a(password) ? new c(new b.b.a.a.d.a(UUID.randomUUID().toString(), "for(int i=0;i<38;i++)\n{\n\ttable[i] = (int)flag.charAt(i)-(int)decode.charAt((base+i)%decode.length());\n}\n//int table[] = {-6, 10, 25, -5, 4, 25, 36, 5, -2, 10, 47, 2, -17, 17, 39, 48, 14, 0, 43, 55, 50, 22, 62, -20, -22, 19, -9, -3, 10, 13, 58, 29, 1, 2, 38, -10, 1, 35};")) : new b.b.a.a.c.b(new IOException("Permission deny"));
```

写出倒推函数：

```
# python3
# 5wimming
def get_flag():
    password = 'c2xydnpja2VlbHlwZHdtdGxtZG5ieA=='
    table = [-6, 10, 25, -5, 4, 25, 36, 5, -2, 10, 47, 2, -17, 17, 39, 48, 14, 0, 43, 55, 50, 22, 62, -20, -22, 19, -9, -3, 10, 13, 58, 29, 1, 2, 38, -10, 1, 35]
    for base in range(32):
        flag = ''
        for i in range(38):
            flag += chr(table[i] + ord(password[(base + i) % len(password)]))
        if 'flag' in flag:
            print(flag)
            break


if __name__ == '__main__':
    get_flag()
```

得到结果flag：

```
flag{slirnvzcknewe1toypdcwemtnlsmdnbx}
```