

XCTF_MOBILE13_基础android

原创

大雄_RE 于 2022-03-15 14:09:05 发布 2112 收藏

分类专栏: [CTF](#) 文章标签: [android 逆向 ctf](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: <https://blog.csdn.net/shadow20080578/article/details/123498783>

版权



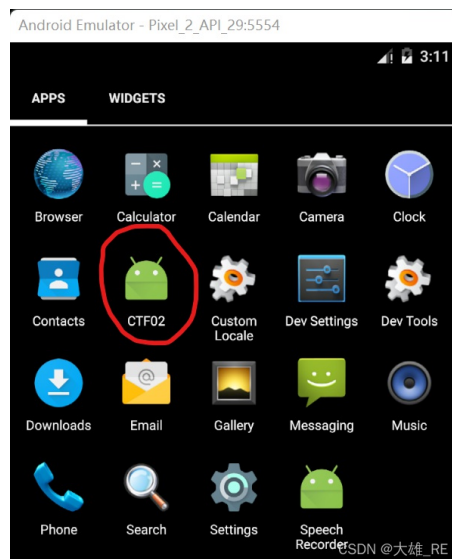
[CTF 专栏收录该内容](#)

17 篇文章 1 订阅

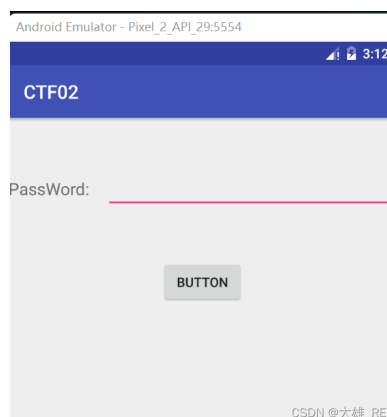
订阅专栏

初探

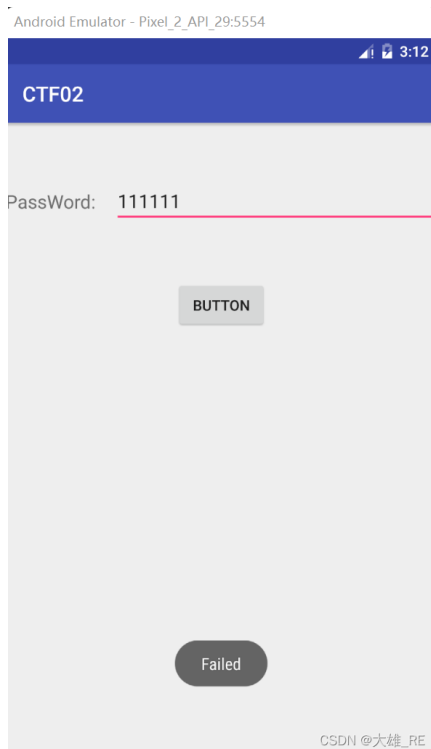
附件为一个apk, 在模拟器里安装一下:



运行后, 主界面很简单, 一个输入框, 一个按钮:



随便输入一个字符串，点击按钮，提示错误：



从使用上只有这些信息，接下来反编译看一看。

MainActivity

使用jadx打开apk，进行反编译。

先看MainActivity类，代码不多，可以轻松找到其中的按钮事件响应函数：

```
this.login.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        if (new Check().checkPassword(MainActivity.this.passWord.getText().toString())) {
            Toast.makeText(MainActivity.this, "Good,Please go on!", 0).show();
            MainActivity.this.startActivity(new Intent(MainActivity.this, MainActivity2.class));
            MainActivity.this.finish();
            return;
        }
        Toast.makeText(MainActivity.this, "Failed", 0).show();
    }
});
```

从代码看，点击按钮后，做四件事：

1. 调用checkPassword检查输入是否正确
2. 调用Toast提示输入正确“Good,Please go on!”
3. 调用startActivity开启MainActivity2
4. 调用finish结束当前Activity

也就是当我们输入正确的字符串后，就通过了验证，开启MainActivity2。

我们看一下检查函数checkPassword。

checkPassword

checkPassword函数是Check类的，反编译代码如下：

```
public boolean checkPassword(String str) {
    char[] pass = str.toCharArray();
    if (pass.length != 12) {
        return false;
    }
    for (int len = 0; len < pass.length; len++) {
        pass[len] = (char) (((255 - len) - 100) - pass[len]);
        if (pass[len] != '0' || len >= 12) {
            return false;
        }
    }
    return true;
}
```

第一个if判断检查字符串长度是否为12。

之后的for循环要求，对于字符串的每个字符， $(255 - \text{字符索引} - 100 - \text{字符}) == 0$ 。

也就是，第一个字符为155 ($255 - 0 - 100 - 155 == 0$)，第二个字符为154 ($255 - 0 - 100 - 154 == 0$)，以此类推。

但是十进制值155大于127，没有对应的ascii码字符，没办法通过键盘直接输入。

这里我想到两个思路：

- 1、修改smali代码，修改 if 语句的判断条件，强行通过验证。
- 2、继续向下看后续反编译代码，看看能否通过静态分析找到flag。

下面我是按照第二个思路完成了解题，该题目后续并不难。如果有小伙伴知道如何能给该app提供ascii码以外的输入，麻烦教给我~~~~

MainActivity2

上面说到，在MainActivity中，如果checkPassword验证成功，就创建MainActivity2，接下来就看看这个类。

这个类代码很少，除了一个简单的init函数，就是一个按钮事件响应函数：

```
this.button.setOnClickListener(new View.OnClickListener() { /
    public void onClick(View v) {
        MainActivity2.this.sendBroadcast(new Intent(MainActivity2.this.editText.getText().toString(
    })
});
```

这个按钮事件响应函数就干了一件事，通过sendBroadcast发送广播，广播内容为一个editText控件的字符串。

我们需要找一找谁在处理这个广播消息。

不了解android广播的小伙伴可以看一下[这篇文章](#)。其中最重要的就是，广播接收器需要实现为BroadcastReceiver类的子类，并重写onReceive()方法来接收以Intent对象为参数的消息。

这道题的apk中类不多，只有一个为BroadcastReceiver的子类，就是GetAndChange类。这个类的onReceive函数功能就一句话：

```
public void onReceive(Context context, Intent intent) {
    context.startActivity(new Intent(context, NextContent.class));
}
```

也就是收到广播消息后，又启动了一个新Activity，NextContent。

顺着这个流程，我们来看看NextContent类。

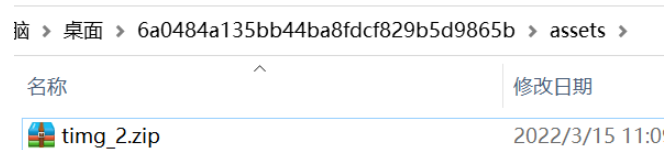
NextContent

NextContent类的onCreate函数中除了进行一些初始化以外，调用了NextContent类的Change函数，删除掉一些不重要的代码，该函数的核心功能代码可以简化为：

```
public void Change() {
    String strFile = getApplicationContext().getDatabasePath("img.jpg").getAbsolutePath();
    File f = new File(strFile);
    InputStream is = getApplicationContext().getResources().getAssets().open("timg_2.zip");
    FileOutputStream fos = new FileOutputStream(strFile);
    byte[] buffer = new byte[1024];
    while (true) {
        int count = is.read(buffer);
        if (count <= 0) {
            break;
        }
        fos.write(buffer, 0, count);
    }
    fos.flush();
    fos.close();
    is.close();
}
}
```

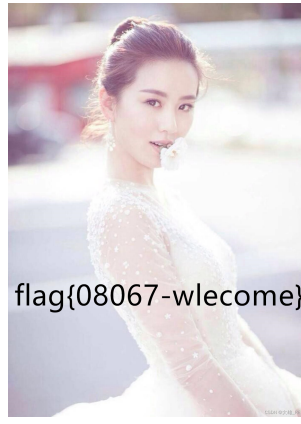
就是将资源里的一个timg_2.zip的资源文件，写到名为imp.jpg的文件。

获得资源文件很简单，直接将apk的后缀名改为zip，进行解压。在解压后的assets目录下就能找到资源文件：



名称	修改日期
timg_2.zip	2022/3/15 11:0

对资源文件进行重命名后，得到一个图片：



flag就在图片上了。