




[原创]安卓逆向之2016年华山杯CTF安卓writeUp

转载

双刃剑客  于 2017-06-17 23:56:47 发布  2124  收藏

分类专栏: [android 逆向](#)



[android 逆向 专栏收录该内容](#)

126 篇文章 7 订阅

订阅专栏

转: <http://bbs.pediy.com/thread-218555.htm>

第一题:

题目下载: <http://download.csdn.net/detail/darmao/9873200>

题目打开是这样的, 有个序列号, 然后根据这个序列号生成一个注册码, 然后输入即可解锁。

首先我们, 这个序列号是不停更新的, 所以我们需要先反编译, 将其变化的周期改长一些:

点击提交按钮的代码逻辑是这样的:

调用了 encryption01.MyEncryption() 函数, 用输入的注册码和这个函数的返回值相比较, 一般的思路就是开始看着搞函数的加密算法, 但是我们看看 smali 代码:

这里将 encryption01.MyEncryption() 的返回值放到了 v1 寄存器里, 这个时候可以添加一个 toast, 将注册码弹出来:

重新打包, 触发 onClick 函数, 这时就会将正确的注册码弹出来了。

第二题:

题目下载: <http://download.csdn.net/detail/darmao/9873200>

安装后打开是这样的:

输入用户名和密码登陆

扔进 jeb 里看看:

加壳了, 看看是如何动态加载的:

**依次调用了 readDexFileFromApk() -> splitPayloadFromDex()

第一个函数先将 apk 解压, 将里边的 dex 文件读入到一个 byte[] 里, 重点在 splitPayloadFromDex() 函数: **

```

int v5 = arg25.length; //apk中dex文件的长度
byte[] v8 = new byte[4];
System.arraycopy(arg25, v5 - 4, v8, 0, 4);//将dex文件中的最后四个字节读取到v8
int v19 = new DataInputStream(new ByteArrayInputStream(v8)).readInt();//将四个
byte转成一个int,也就是未加壳的apk的长度
System.out.println(Integer.toHexString(v19)); byte[] v18 = new byte[v19];
System.arraycopy(arg25, v5 - 4 - v19, v18, 0, v19);//这里是从length-4-v19开始copy
v18 = this.decrypt(v18);//解密
File v9 = new File(this.apkFileName);1234567891012345678910

```

通过以上分析，我们可以得到加壳了以后的apk的结构应该是：

```

+-----+-----+-----+
+ 壳子 | 原始apk | 原始apk的长度 |
+-----+-----+-----+

```

我们需要做的就是将原始apk读取出来，然后解密，看看解密算法：

**直接和0xff或一下即可

Java代码如下：**

```

public static byte[] newByte(String path) throws IOException
    { //path是apk中的dex文件的路径
        File file = new File(path); byte[] srcLen = new byte[(int) file.length()]; byte[] temp = new byte[1024]; int tempLen = 0; int i
= 0;
        BufferedInputStream bis = new BufferedInputStream(new FileInputStream(file)); while (-1 != (tempLen = bis.read(temp)))
        {
            System.arraycopy(temp, 0, srcLen, i, tempLen);
            i += tempLen;
        } byte[] decryptLenByte = new byte[4];
        System.arraycopy(srcLen, srcLen.length - 4, decryptLenByte, 0, 4);
        ByteArrayInputStream byteInput = new ByteArrayInputStream(decryptLenByte);
        DataInputStream dataOutput = new DataInputStream(byteInput); int decryptLen = dataOutput.readInt();

        System.out.println("长度是: " + Integer.toHexString(decryptLen)); byte[] newDexByte = new byte[decryptLen];
        System.arraycopy(srcLen, srcLen.length - 4 - decryptLen, newDexByte, 0, decryptLen); for (int j = 0; j < decryptLen; j++)
        {
            newDexByte[j] = (byte) ((newDexByte[j] ^ 255) & 0xff);
        }
        BufferedOutputStream bos = new BufferedOutputStream(new FileOutputStream(new File("D:\\test\\new_write_dex.apk")));
        bos.write(newDexByte); return newDexByte;
    } 12345678910111213141516171819202122232425262728293031321234567891011121314151617181920212223242526272829303132

```

**将这个读取出来的新apk拖到jeb里：

onClick函数：**

这里调用了checkNameAndPassword这个函数，其中第一个参数是：用户名，第二个是密码，还有另外一个参数，我们看看这个函数：

**首先对用户名做了使用sha-1进行了哈希，然后读取前16个字节

进行判断的部分：密码要等于这前十六个字节，同时用户名等于其传进去的第三个参数，第三个参数是什么呢？**

U2hlMTFfTjZSYw== 这个字符串的解码

总结：用户名：base64.decode("U2hlMTFfTjZSYw==");
密码：sha1("用户名").subString(0,16);
用户名：She11_N6Rc
密码：6acbbca78fdca0c5

第三题：

题目下载：<http://download.csdn.net/detail/darmao/9873200>

拿到这个题，apk无法安装，拖到jeb里也无法反编译，试了一下zip伪加密，将504B0102后边的第五第六个字节都改成偶数，两处需要修改：

然后安装就能打开了。

**首先打开有个闪屏：提示未获取权限 5秒之后就退出了

这里有三种解决方法：

第一种：最简单的方法**

**manifest文件中注册了两个activity,打开的闪屏的是主activity，我们的目的是打开MainActivity，所有直接打开adb shell，获得shell,提升到root权限，执行：

am start -n com.example.testndk4/com.example.testndk4.MainActivity 直接就来到了输入密码的界面。 **

*第二种：*修改smali

我们来看看闪屏activity的逻辑： **

**调用了isExit方法，如果返回值为true就打开mainActivity，否则准备关闭

看看对应的smali:**

**将这里的if-eqz 改成if-nez即可

第三种：我们去看看native层的isExit方法： **

**这里返回值恒为0，所以只需要将对应处改成1，然后重新打包即可。

经过以上步骤，终于来到了mainActivity,直接需要输入一个密码:**

我们去看看这里的代码逻辑：

逻辑很简单，将输入的密码传递到native层的encodePassword进行了处理，我们去动态调试一下吧

打开ida,断点下在这个函数的处:

**v2是从Java层拿到的密码, v4是正确的密码, 这两个进行比较
单步到v4=(encodePS)(amp;10)处**

**这里调用encodePS函数, 返回值就是正确的password,
执行完看返回值, 在R0寄存器: **

这里就是正确密码啦

**password:kienietIeAehfyih