

[illegible]

<b>Name</b>	Reversing and Patching APK
<b>URL</b>	<a href="https://attackdefense.com/challengedetails?cid=1628">https://attackdefense.com/challengedetails?cid=1628</a>
<b>Type</b>	Android Pentesting : Basics

**Important Note:** This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

**Objective:** Perform the following activities:

1. Scan the APK with [SUPERAndroidAnalyzer](#) tool and identify the issue which allows anyone to backup your application data via adb.
2. Open the APK archive using apktool
3. Fix the issue by modifying the corresponding file
4. Repack the files into an APK archive
5. Sign the APK using jarsigner

**Activity 1:** Scan the APK with super-analyzer.

**Command:** super-analyzer UnCrackable-Level1.apk

```
root@attackdefense:~# super-analyzer UnCrackable-Level1.apk

Starting analysis of UnCrackable-Level1.
Application decompressed.
Jar file generated.
Application decompiled.
Results structure created.
WARN: Seems that the package in the AndroidManifest.xml is not the same as the application ID provided.
vell, manifest package: owasp.mstg.uncrackable1
If you need more information, try to run the program again with the -v flag.
Manifest analyzed.
Source code analyzed.

HTML report generated.
```

Check the report generated by super-analyzer.

### Commands:

ls -l

vim results/owasp.mstg.uncrackable1/index.html

```
root@attackdefense:~# ls -l
total 76
-rw-r--r-- 1 root root 66651 Jan 16 07:08 UnCrackable-Level1.apk
drwxr-xr-x 3 root root 4096 Jan 17 10:27 dist
drwxr-xr-x 3 root root 4096 Jan 17 10:27 results
root@attackdefense:~#
root@attackdefense:~# vim results/owasp.mstg.uncrackable1/
css/      img/      index.html  js/      src/
root@attackdefense:~# vim results/owasp.mstg.uncrackable1/index.html
```

Search for “backup” keyword in the report.

The backup of the application through adb is enabled. And, this is done by setting a parameter in AndroidManifest.xml

```
<li>
  <strong>Label:</strong> Allows Backup</li>
  <div style="display: none">
    <li>
      <strong>Description:</strong> This option allows backups of the application data via adb. Malicious people with physical access
      could use adb to get private data of your app into their PC.</li>
    <li>
      <strong>File:</strong>
      <a href="src/AndroidManifest.xml.html?start_line=4&end_line=4&criticality=medium#code-line-4">AndroidManifest.xml</a>
    </li>
    <li>
      <strong>Line
      :</strong>
      4
    </li>
  </div>
</li>
```

/backup

Delete the directories created by super-analyzer.

```
root@attackdefense:~#  
root@attackdefense:~# rm -rf dist/ results/  
root@attackdefense:~#
```

**Activity 2:** Open the APK archive using apktool. This will produce a directory with the same name as APK.

**Command:** apktool d UnCrackable-Level1.apk

```
root@attackdefense:~# apktool d UnCrackable-Level1.apk  
I: Using Apktool 2.4.1 on UnCrackable-Level1.apk  
I: Loading resource table...  
I: Decoding AndroidManifest.xml with resources...  
I: Loading resource table from file: /root/.local/share/apktool/framework/1.apk  
I: Regular manifest package...  
I: Decoding file-resources...  
I: Decoding values */* XMLs...  
I: Baksmaling classes.dex...  
I: Copying assets and libs...  
I: Copying unknown files...  
I: Copying original files...  
root@attackdefense:~#  
root@attackdefense:~# ls -l  
total 72  
drwxr-xr-x 5 root root 4096 Jan 17 10:29 UnCrackable-Level1  
-rw-r--r-- 1 root root 66651 Jan 16 07:08 UnCrackable-Level1.apk  
root@attackdefense:~#
```

**Activity 3:** Fix the issue by modifying the corresponding file. The setting is kept in AndroidManifest.xml file.

**Command:** vim UnCrackable-Level1/AndroidManifest.xml



```

root@attackdefense:~# ls -l UnCrackable-Level1/
total 20
-rw-r--r--  1 root root  663 Jan 17 10:29 AndroidManifest.xml
-rw-r--r--  1 root root  442 Jan 17 10:29 apktool.yml
drwxr-xr-x  3 root root 4096 Jan 17 10:29 original
drwxr-xr-x 12 root root 4096 Jan 17 10:29 res
drwxr-xr-x  3 root root 4096 Jan 17 10:29 smali
root@attackdefense:~#
root@attackdefense:~# vim UnCrackable-Level1/AndroidManifest.xml

```

In the AndroidManifest.xml, one can observe allowBackup="true" parameter value pair.

```

<?xml version="1.0" encoding="utf-8" standalone="no"?><manifest xmlns:android="http://schemas.android.com/apk/res/android" package="owasp.mstg.uncrackable1">
  <application android:allowBackup="true" android:icon="@mipmap/ic_launcher" android:label="@string/app_name" android:theme="@style/AppTheme"
  >
    <activity android:label="@string/app_name" android:name="sg.vantagepoint.uncrackable1.MainActivity">
      <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
        <category android:name="android.intent.category.LAUNCHER"/>
      </intent-filter>
    </activity>
  </application>
</manifest>

```

Change the true to false.

```

<?xml version="1.0" encoding="utf-8" standalone="no"?><manifest xmlns:android="http://schemas.android.com/apk/res/android" package="owasp.mstg.uncrackable1">
  <application android:allowBackup="false" android:icon="@mipmap/ic_launcher" android:label="@string/app_name" android:theme="@style/AppTheme"
  ">
    <activity android:label="@string/app_name" android:name="sg.vantagepoint.uncrackable1.MainActivity">
      <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
        <category android:name="android.intent.category.LAUNCHER"/>
      </intent-filter>
    </activity>
  </application>
</manifest>

```

Save the changes and exit the file.

#### Activity 4: Repack the files into an APK archive

First delete the older APK and then use apktool to repack the opened files.

### Commands:

```
rm UnCrackable-Level1.apk
apktool b UnCrackable-Level1/
```

```
root@attackdefense:~# rm UnCrackable-Level1.apk
root@attackdefense:~#
root@attackdefense:~# apktool b UnCrackable-Level1/
I: Using Apktool 2.4.1
I: Checking whether sources has changed...
I: Smaling smali folder into classes.dex...
I: Checking whether resources has changed...
I: Building resources...
I: Building apk file...
I: Copying unknown files/dir...
I: Built apk...
root@attackdefense:~#
```

The newly created APK can be found in “UnCrackable-Level1/dist” directory.

Move the APK to present working directory.

**Command:** mv UnCrackable-Level1/dist/UnCrackable-Level1.apk .

```
root@attackdefense:~# ls -l UnCrackable-Level1/
total 28
-rw-r--r--  1 root root  665 Jan 17 10:31 AndroidManifest.xml
-rw-r--r--  1 root root  442 Jan 17 10:29 apktool.yml
drwxr-xr-x  3 root root 4096 Jan 17 10:32 build
drwxr-xr-x  2 root root 4096 Jan 17 10:32 dist
drwxr-xr-x  3 root root 4096 Jan 17 10:29 original
drwxr-xr-x 12 root root 4096 Jan 17 10:29 res
drwxr-xr-x  3 root root 4096 Jan 17 10:29 smali
root@attackdefense:~#
root@attackdefense:~# mv UnCrackable-Level1/dist/UnCrackable-Level1.apk .
root@attackdefense:~#
```

Run super-analyzer again on newly repacked APK.

**Command:** super-analyzer UnCrackable-Level1.apk

```
root@attackdefense:~# super-analyzer UnCrackable-Level1.apk

Starting analysis of UnCrackable-Level1.
Application decompressed.
Jar file generated.
Application decompiled.
Results structure created.
WARN: Seems that the package in the AndroidManifest.xml is not the same as the application ID provided.
vell1, manifest package: owasp.mstg.uncrackable1
If you need more information, try to run the program again with the -v flag.
Manifest analyzed.
Source code analyzed.

HTML report generated.
root@attackdefense:~#
```

Check the report generated by super-analyzer

**Commands:**

ls -l

vim results/owasp.mstg.uncrackable1/index.html

```
root@attackdefense:~# ls -l
total 72
drwxr-xr-x 7 root root 4096 Jan 17 10:32 UnCrackable-Level1
-rw-r--r-- 1 root root 60104 Jan 17 10:32 UnCrackable-Level1.apk
drwxr-xr-x 3 root root 4096 Jan 17 10:33 dist
drwxr-xr-x 3 root root 4096 Jan 17 10:33 results
root@attackdefense:~#
root@attackdefense:~#
root@attackdefense:~# vim results/owasp.mstg.uncrackable1/
css/      img/      index.html  js/      src/
root@attackdefense:~# vim results/owasp.mstg.uncrackable1/index.html
```

Search for “backup” string in the report file. It should not result in anything.



```

        <strong>Label:</strong> Super user privileges.</li>
      <div style="display: none">
        <li>
          <strong>Description:</strong> This applications may require super user privileges.</li>
          <li>
            <strong>File:</strong>
            <a href="src/classes/sg/vantagepoint/a/c.java.html?start_line=35&end_line=35&es/sg/vantagepoint/a/c.java">
es/sg/vantagepoint/a/c.java</a>
          </li>
          <li>
            <strong>Line
              :</strong>
            35
          </li>
          <li>

```

E486: Pattern not found: backup

This means the issue has been fixed.

Delete all directories created by apktool and super-analyzer

**Command:** `rm -rf UnCrackable-Level1 dist/ results/`

```

root@attackdefense:~# rm -rf UnCrackable-Level1 dist/ results/
root@attackdefense:~#
root@attackdefense:~# ls -l
total 60
-rw-r--r-- 1 root root 60104 Jan 17 10:32 UnCrackable-Level1.apk
root@attackdefense:~#

```

**Activity 5:** Sign the APK using jarsigner

A key is required for signing. Use keytool to generate a key.

**Command:** `keytool -genkey -v -keystore dummy-release-key.keystore -alias dummy_alias_name -keyalg RSA -keysize 2048 -validity 10000`



```

root@attackdefense:~# keytool -genkey -v -keystore dummy-release-key.keystore -alias dummy_alias_name -keyalg RSA -keysize 2048 -validity 10000
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]: dummy
What is the name of your organizational unit?
[Unknown]: dummyorg
What is the name of your organization?
[Unknown]: dummyo
What is the name of your City or Locality?
[Unknown]:
What is the name of your State or Province?
[Unknown]:
What is the two-letter country code for this unit?
[Unknown]:
Is CN=dummy, OU=dummyorg, O=dummyo, L=Unknown, ST=Unknown, C=Unknown correct?
[no]: yes

Generating 2,048 bit RSA key pair and self-signed certificate (SHA256withRSA) with a validity of 10,000 days
for: CN=dummy, OU=dummyorg, O=dummyo, L=Unknown, ST=Unknown, C=Unknown
[Storing dummy-release-key.keystore]
root@attackdefense:~#

```

The keytool program will prompt for different information, fill the information and also assign a password to the key.

This will generate the required key and a self-signed certificate..

Now, this can be used to sign the APK using jarsigner.

**Command:** jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore dummy-release-key.keystore UnCrackable-Level1.apk dummy\_alias\_name

```


root@attackdefense:~# jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore dummy-release-key.keystore UnCrackable-Level1.apk dummy_
alias_name
Enter Passphrase for keystore:
adding: META-INF/MANIFEST.MF
adding: META-INF/DUMMY_AL.SF
adding: META-INF/DUMMY_AL.RSA
signing: res/layout/activity_main.xml
signing: res/mipmap-xxxhdpi/ic_launcher.png
signing: res/mipmap-hdpi/ic_launcher.png
signing: res/mipmap-xxhdpi/ic_launcher.png
signing: res/menu/menu_main.xml
signing: res/mipmap-mdpi/ic_launcher.png
signing: res/mipmap-xhdpi/ic_launcher.png
signing: AndroidManifest.xml
signing: classes.dex
signing: resources.arsc

>>> Signer
X.509, CN=dummy, OU=dummyorg, O=dummyo, L=Unknown, ST=Unknown, C=Unknown
[trusted certificate]

jar signed.

Warning:
The signer's certificate is self-signed.
root@attackdefense:~#

```



The APK is signed now.

### References:

1. Android mobile pentesting 101:  
[https://github.com/tsug0d/AndroidMobilePentest101/blob/master/english/AndroidMobilePentest101\\_Lecture4.pdf](https://github.com/tsug0d/AndroidMobilePentest101/blob/master/english/AndroidMobilePentest101_Lecture4.pdf)