Mobile Application Security:

Android

Introductions

Tony Trummer

Director of Security Engineering - Tinder

Tushar Dalvi

Staff Security Engineer - LinkedIn

Android Intro

The Android Platform

- Based on Linux Kernel
- Apps are written in a Java
- Uses Dalvik or ART VM (no JVM)

OpenJDK

Android OS

Layers

- ❑ Applications
- ❑ Application Framework
- ❑ Libraries
- ❑ Android Runtime
- ❑ Linux Kernel

APK Structure

Dalvik VM runs dex (Dalvik Executables)

ART runs machine code

Apps built into APK files

- AndroidManifest.xml
- classes.dex
- Directories: libs, assets, res, META-INF
- resources.arsc

Tools

APKTOOL

- ❑ Reversing APKs

DEX2JAR

- ❑ Converts dex files to jar

JDGUI, JADX and PROCYON

- ❑ Java Decompilers

Reversing APKs

GoatDroid - GitHub

- http://github.com/jackMannino/OWASP-GoatDroid-Project/downloads

Unzip

- OWASP-GoatDroid-0.9.zip

Locate

- ./OWASP-GoatDroid-0.9/goatdroid_apps/FourGoats/android_app/OWASP GoatDroid- FourGoats Android App.apk

Reversing APKs (cont.)

Decode AndroidManifest.xml (and other resources)

- apktool d OWASP GoatDroid- FourGoats Android App.apk

Copy/rename

- cp OWASP\ GoatDroid-\ FourGoats\ Android\ App.apk FourGoats.zip

Unzip

- Locate classes.dex

Reversing APKs (cont.)

Convert dex file to jar

- ./dex2jar/dex2jar-2.0/d2j-dex2jar.sh classes.dex
- chmod +x ./d2j-dex2jar.sh & chmod +x ./d2j_invoke.sh
  - This creates classes-dex2jar.jar

Open jar file with JD-GUI

- You can now navigate the project
- If you want to view in another program:
  - File > Save All Sources
  - Saves classes-dex2jar.jar.src.zip

AndroidManifest.xml

Locate and open

- ./OWASP\ GoatDroid-\ FourGoats\ Android\ App/AndroidManifest.xml
- Should be in the directory you ran apktool from

Defines attack surface

- Supported versions
- Exported components
- Permissions

Components

Activities

- all the visual screens

Services

- background processes

Content Providers

- manages access to structured data

Broadcast Receivers

- processes broadcast intents

Components in the Manifest

```
<activity android:label="@string/app_name" android:name=".activities.Main">
    <intent-filter>
        <action android:name="android.intent.action.MAIN"/>
        <category android:name="android.intent.category.LAUNCHER"/>
    </intent-filter>
</activity>
```

Permissions & Protection Levels

Requested permissions

- Access GPS, read Contacts, use a custom permission, etc.

Declared permissions

- Allows apps to create their own permissions

Protection levels

- Normal, Dangerous, signature and signatureOrSystem

Requested Permissions in Manifest

```
<uses-permission android:name="android.permission.SEND_SMS"/>

<uses-permission android:name="android.permission.CALL_PHONE"/>

<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>

<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>

<uses-permission android:name="android.permission.INTERNET"/>
```

Declared Permissions in Manifest

```
<permission android:name="custom.PERMISSION"
            android:label="@string/label_permission"
            android:description="@string/desc_permission"
            android:protectionLevel="signature">
</permission>
```

IPC Mechanisms

Intents

- ❑ Implicit - no target specified
- ❑ Explicit - targets a specific component
- ❑ Broadcast - can be resolved to multiple apps
- ❑ Sticky - can persist over time (deprecated)
- ❑ Pending - allows one app to act on behalf of another

Extras

- ❑ carry data for Intents

Example Intent

```
public void sendMessage(View view) {
    Intent intent = new Intent(this, DisplayMessageActivity.class);
    EditText editText = (EditText) findViewById(R.id.edit_message);
    String message = editText.getText().toString();
    intent.putExtra(EXTRA_MESSAGE, message);
    startActivity(intent);
}
```

Important Methods

Activities

- ❑ onCreate
- ❑ onRestart
- ❑ onStart
- ❑ onResume
- ❑ onPause
- ❑ onDestroy

Services

- ❑ onBind
- ❑ onCreate
- ❑ onDestroy
- ❑ onStartCommand

Receivers

- ❑ onReceive

Providers

- ❑ ContentResolver.query()
- ❑ ContentResolver.call()
- ❑ ContentResolver.insert()
- ❑ ContentResolver.update()
- ❑ ContentResolver.delete()
- ❑ ContentResolver.bulkInsert()

Example Vulnerability

org/owasp/goatdroid/fourgoats/broadcastreceivers/SendSMSNowReceiver.java

```
public class SendSMSNowReceiver
  extends BroadcastReceiver
{
  Context context;

  public void onReceive(Context paramContext, Intent paramIntent)
  {
    this.context = paramContext;
    paramContext = SmsManager.getDefault();
    paramIntent = paramIntent.getExtras();
    paramContext.sendTextMessage(paramIntent.getString("phoneNumber"), null, paramIntent.getString("message"), null, null);
```

Testing Tools

QARK

- ❑ Automates decompiling
- ❑ Automates decompressing
- ❑ Static Code Analysis
- ❑ Automated Exploitation
- ❑ SDLC integration
- ❑ Education
- ❑ https://github.com/linkedin/qark
- ❑ Free & OpenSource

ADB

- ❑ CLI for Android devices
- ❑ Debugging
- ❑ Manual Testing
- ❑ Bundled in Android Studio

ADB

Commands

- adb shell : interactive shell
  - Once connected you can omit the “adb shell”
- (adb shell) pm list packages : shows all installed apks
- (adb shell) pm path <package name> : returns path to the apps files
- (adb shell) am start : start an activity (hit Enter to see options)
- (adb shell) am startservice : start a service
- (adb shell) am broadcast : send a broadcast
- (adb shell) input [text|keyevent] : send keystrokes to device

ADB

Commands

- adb pull <device path> <local path>
- adb push <local file> <device file>
- adb help
- exit

More info

- http://developer.android.com/tools/help/shell.html

Contact

Tony Trummer

- @SecBro1
- https://www.linkedin.com/in/tonytrummer

Tushar Dalvi

- @tushardalvi
- https://www.linkedin.com/in/tdalvi