Permissions

Permissions

Android protects resources & data with permissions

Used to limit access to:

- User information – e.g, contacts
- Cost-sensitive API's – e.g., SMS/MMS
- System resources – e.g., Camera

Permissions

Permissions are represented as strings

In AndroidManifest.xml, apps declare the permissions

They use themselves

They require of other components

Using Permissions

Applications specify permissions they use through a <uses-permission> tag

Users must accept these permissions before an application can be installed

Using Permissions

```
<manifest ... > !
...!
<uses-permission  android:name=!
                        "android.permission.CAMERA”/>
<uses-permission  android:name=!
                      "android.permission.INTERNET”/>
<uses-permission  android:name=!
   “android.permission.ACCESS_FINE_LOCATION”/>
...
</manifest >
```

See: http://developer.android.com/reference/!
android/Manifest.permission.html

MapLocationFromContacts

Select a contact from contacts database

Display a map centered on selected contact’s address

MapLocationFromContacts

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="course.examples.MapLocationFromContacts"
    android:versionCode="1"
    android:versionName="1.0" >

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

    <uses-sdk
```

Defining Permissions

Apps can also define and enforce their own permissions

Defining Permissions

Suppose your application performs a privileged/dangerous operation

You might not want to allow just any application to invoke yours

So you can define & enforce your own permission

PermissionExampleBoom

Simple Application that performs a (just pretend) dangerous action


10:44
Permission Example Boom
BOOM!

Define & Enforcing Permissions

IF you don’t want just anyone to run PermissionExampleBoom

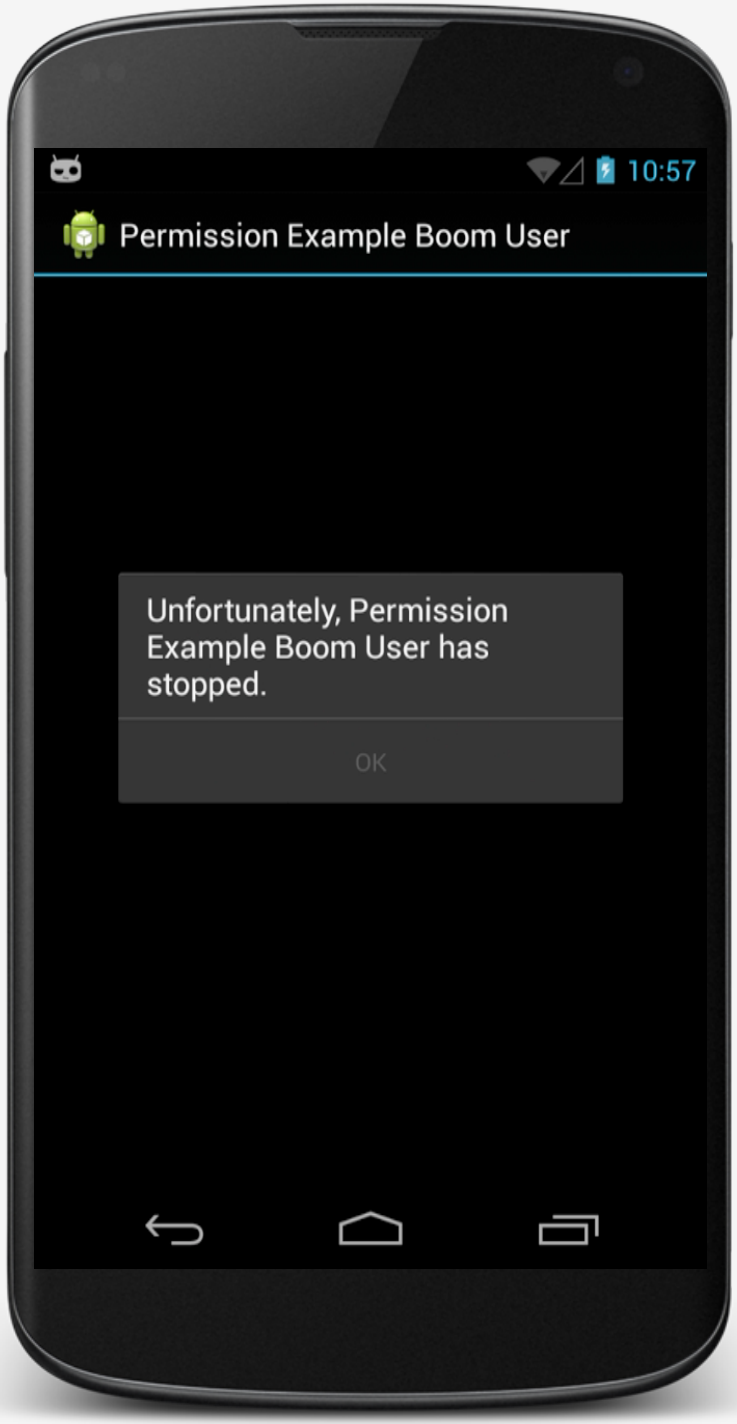
Define & enforce an application-specific permission

PermissionExampleBoom

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="course.examples.permissionexample.boom"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="13"
        android:targetSdkVersion="20" />

    <!-- Defines a custom permission -->
    <permission
        android:name="course.examples.permissionexample.BOOM_PERM"
        android:description="@string/boom_perm_string"
        android:label="@string/boom_permission_label_string" >
    </permission>

    <!--  Enforces the BOOM_PERM permission on users of this application  -->
    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:permission="course.examples.permissionexample.BOOM_PERM" >
        <activity
            android:name=".BoomActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
```

Using the new Permission

Apps that want to use PermissionExampleBoom must now acquire the correct permission


10:57
Permission Example Boom User
Unfortunately, Permission Example Boom User has stopped.
OK

Uses-Permission

An application can declare/accept the permissions of the Applications it uses

PermissionExampleBoomUser

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="course.examples.permissionexample.boomUser"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="13"
        android:targetSdkVersion="20" />

    <!--  grant the "...BOOM_PERM permission to this application -->
    <uses-permission android:name="course.examples.permissionexample.BOOM_PERM" />
```


10:59
Permission Example Boom
BOOM!

Component Permissions

Individual components can set their own permissions, restricting which other components can access them

Component permissions take precedence over application-level permissions

Activity Permissions

Restricts which components can start the associated activity

Checked within execution of

- startActivity()
- startActivityForResult()

Throws SecurityException on permissions failure

Service Permissions

Restricts which components can start or bind to the associated service

Checked within execution of

- Context.startService()
- Context.stopService()
- Context.bindService()

Throws SecurityException on permissions failure

BroadcastReceiver Permissions

Restricts which components can send & receive broadcasts

Permissions checked in multiple places

More on this when we discuss BroadcastReceivers

ContentProvider Permissions

Restrict which components can read & write the data in a ContentProvider

More on this when we discuss ContentProviders