Contents

1 Permissions

1.1 Old Permissions

Show permissions required at install time. Not prompted again regarding permissions at run-time.

- Not yet made a commitment (financial, mental) to the application. Can compare to other applications
- **Not per session / at run-time**
- Seamless switching between Activities / applications
- Would slow down the user experience
- Train users to click ok repeatedly without considering the implications

1.2 New permissions

- No access by default. Control access to specific mechanisms
- Applications can offer protected access to resources and data with permissions. Permissions explicitly granted by users / the system
- Permission architecture
  - Applications statically declare permissions
  - Required of components interacting with them. **You must** have this permission to interact with me
  - Required by components they interact with. **I will** need these permissions
  - Android requires users consent for specific permissions

2 Normal or Dangerous

Normal

- Do not directly risk the users privacy
- Network state, Internet, Alarms, Wallpaper
- Granted automatically
- But still must be declared in the manifest

Dangerous

- Potentially do risk the users privacy
- The user must explicitly approve the permission request
- Need to think what to do if permission is denied

Some permissions:

- Cost-Sensitive APIs: Telephony, SMS/MMS, In-App Billing, NFC Access
- Personal Information. Contacts, calendar, messages, emails
- Device Meta-data . System logs, browser history, network identifiers
- Sensitive Input Devices. Interaction with the surrounding environment: Camera, microphone, GPS

3 Component Permissions

Activity

- Restricts which components can start the activity
- Checked within execution of: `startActivity(), startActivityForResult()`

Service

- Restricts which components can start or bind to the associated service
- Checked within execution of: `Context.startService(), Context.stopService(), Context.bindService()`

Others

- ContentProvider: Restricts which components can read or write to a ContentProvider
- BroadcastReceiver: Restricts which components can register to receive a certain Broadcast
- Throw SecurityException on permissions failure: Usually as weve forgotten to ask for permission during installation

4 Runtime Permissions

Each time we want to do something dangerous

- `shouldShowRequestPermissionRationale()` True if the user has previously denied the request
- `requestPermissions()`
- `onRequestPermissionsResult()` Either do the dangerous thing, or gracefully degrade the functionality of the app

5 Permissions vs Use

- Read your text messages: To confirm your phone number via text message (if youve added it to your account)
- Read/write contacts: To import and sync your phones contacts to Facebook, or vice versa (think updating contact images)
- Add and/or modify calendar events and send emails to guests without your knowledge: To see your Facebook events in your phones calendar
- Read calendar events plus confidential information: To check your calendar for you to see if you have something already scheduled for the time of the Facebook event youre currently viewing
- Good practice to explain why an application needs a permission, especially if now not having it will prevent it from functioning

6 Temporary URI Permissions

Applications making use of multiple Activities. Access to the mail should be protected by permissions, since this is sensitive user data. However, if a URI to an image attachment is given to an image viewer, that image viewer will not have permission to open the attachment since it has no reason to hold a permission to access all email. **Allow access to specific URIs, not the whole provider**.

Temporary URI permissions last while the stack of the receiving Activity is active

Reference section

placeholder