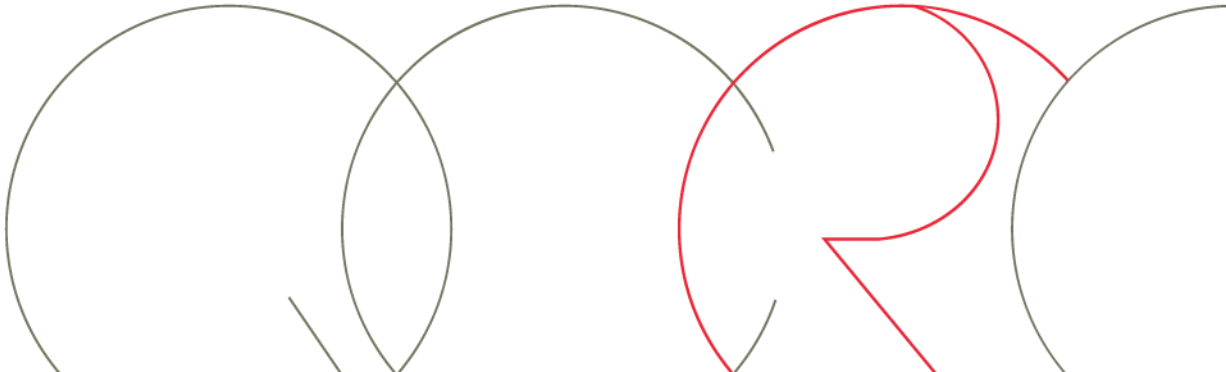
جامعة حمد بن خليفة
HAMAD BIN KHALIFA UNIVERSITY

Small Changes, Big Changes: An Updated View on the Android Permission System

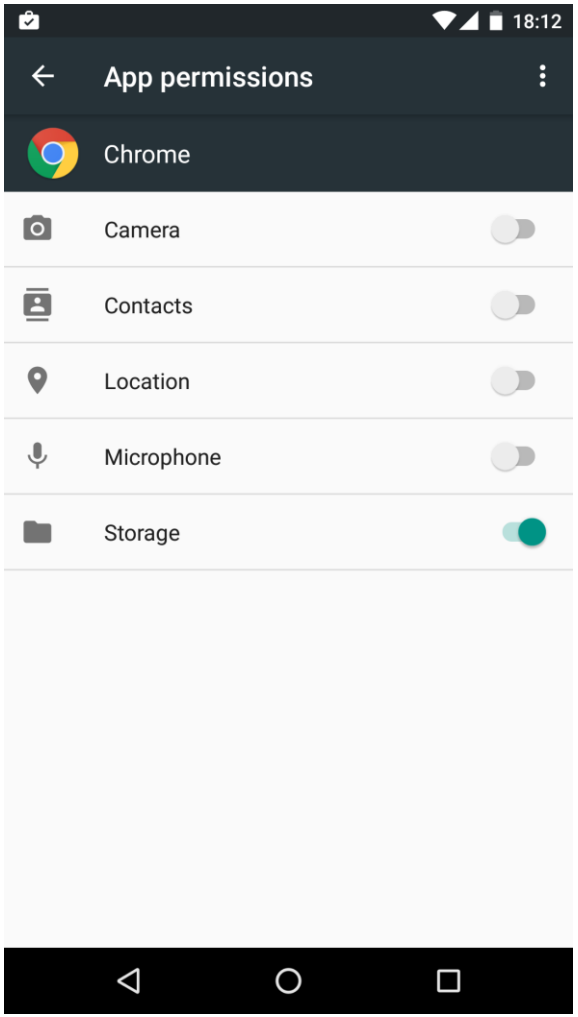
Yury Zhauniarovich
Olga Gadyatskaya



RAID 2016

Sensitive Resource Protection in Android

- Android is the most popular mobile OS:
  - ~ 2 billion of third-party apps only on Google Play
  - many more markets exist
  - **a lot** of malware/adware/greyware/…
- The end users can control access of third-party apps to their sensitive data via **permissions**
- Permissions is a general way how access to sensitive resources is controlled on Android

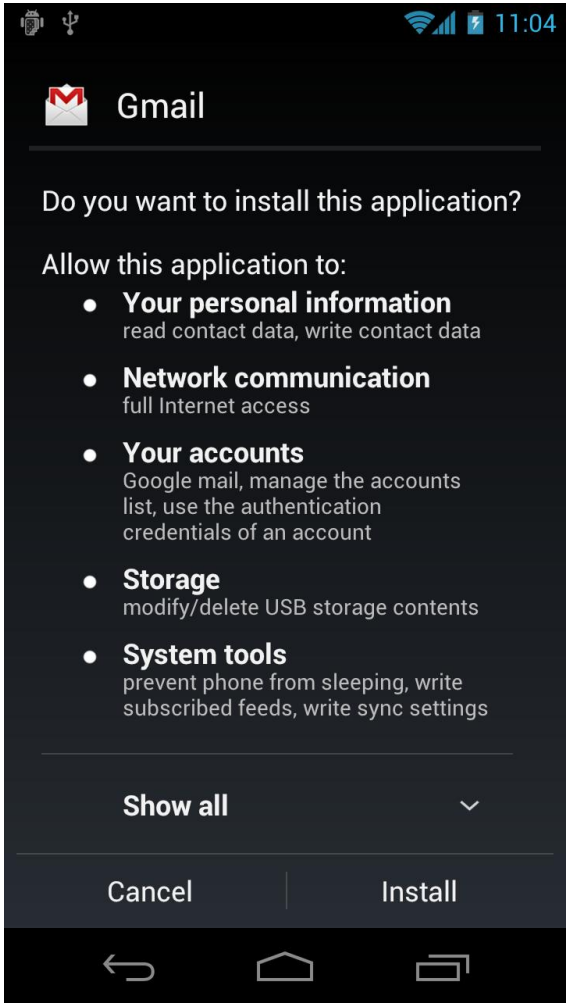


Android Permission System

- **Android Permission** is a security label assigned to a sensitive resource that protects the access to this component. Once an app is granted with the permission it receives access to the corresponding resource
- Permissions may belong to a ***permission group***, a category of permissions protecting similar functionality
- Permissions are declared in ***AndroidManifest.xml*** files
- Permissions:
  - ***Platform*** – defined within the sources of Android protecting **the components of the operating system**
  - ***Custom*** – declared by third-party developers protecting **the resources of the application**

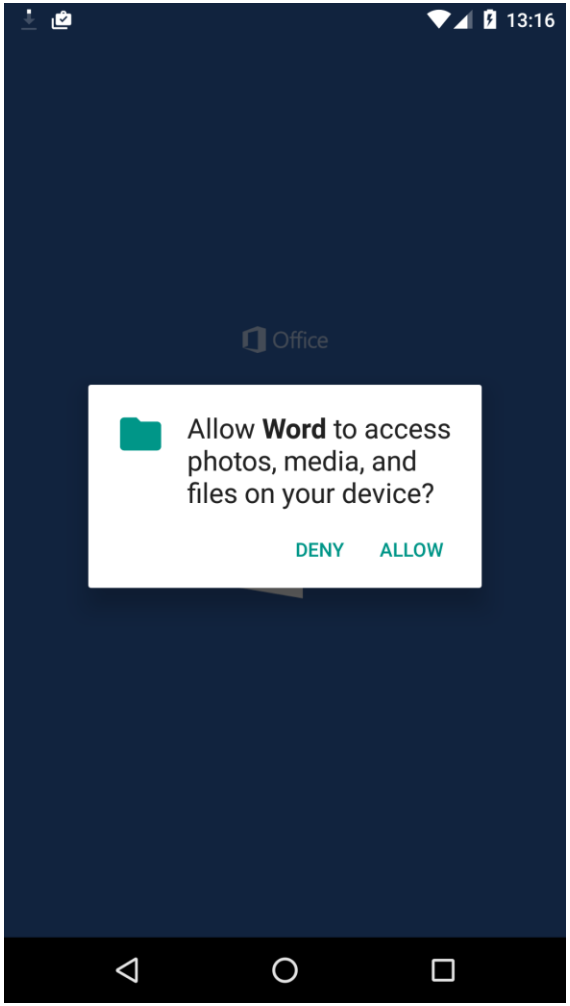
Established View on the Permission System

- All permissions are granted at the installation time or the application is not installed
- Granted permissions cannot be revoked
- There are 4 protection levels:
  - ***normal*** (granted automatically)
  - ***dangerous*** (granted after user's approval)
  - ***signature*** (granted only if the packages declaring and requesting permissions are signed with the same certificate)
  - ***signature|system*** (similar to signature, but also granted if the application is installed on the system image)
- Only dangerous permissions are approved by the user



Updated View on the Permission System (23+)

- Permissions:
  - ***Installation time*** (normal, signature, signature|system)
  - ***Runtime*** (dangerous)
- Installation time permissions are granted at install and cannot be revoked
- Runtime permissions are disabled by default, granted when required (according to an app developer) and can be revoked



Motivation

- Permission system is a central component for the Android security
- Permissions are used in many research articles exploring the Android security
- The detailed exploration of the Android permission system for the scientific community has been done in 2009 (“Understanding Android Security” by W. Enck et al.)
- In Android 6.0 (Marshmallow), the permission model has been considerably changed

Our Approach

- We analyzed 16 versions of Android resulted in API change (from 1.6 [Donut] up to 6.0 [Marshmallow])
- We developed scripts:
  - to extract declared permissions and their properties from the manifest files
  - to compare the extracted data for different versions of Android
- We applied our software to the considered versions and performed *quantitative* and *qualitative* analysis of the changes

- In this work we concentrated on the **platform permissions**
- We divide platform permissions into: ***core, package, sample, test***

BONUS:

- For this presentation, we also included the analysis of Android 7.0 [Nougat] released on August 22, 2016

Permission Declaration

```
<!-- Allows an application to send SMS messages.
     <p>Protection level: dangerous
-->
<permission android:name="android.permission.SEND_SMS"
    android:permissionGroup="android.permission-group.SMS"
    android:label="@string/permlab_sendSms"
    android:description="@string/permdesc_sendSms"
    android:permissionFlags="costsMoney"
    android:protectionLevel="dangerous" />
```

Permission Declaration

```
<!-- Allows an application to send SMS messages.
     <p>Protection level: dangerous
-->
<permission android:name="android.permission.SEND_SMS"
    android:permissionGroup="android.permission-group.SMS"
    android:label="@string/permlab_sendSms"
    android:description="@string/permdesc_sendSms"
    android:permissionFlags="costsMoney"
    android:protectionLevel="dangerous" />
```

Permission Declaration

```
<!-- Allows an application to send SMS messages.
     <p>Protection level: dangerous
-->
<permission android:name="android.permission.SEND_SMS"
    android:permissionGroup="android.permission-group.SMS"
    android:label="@string/permlab_sendSms"
    android:description="@string/permdesc_sendSms"
    android:permissionFlags="costsMoney"
    android:protectionLevel="dangerous" />
```

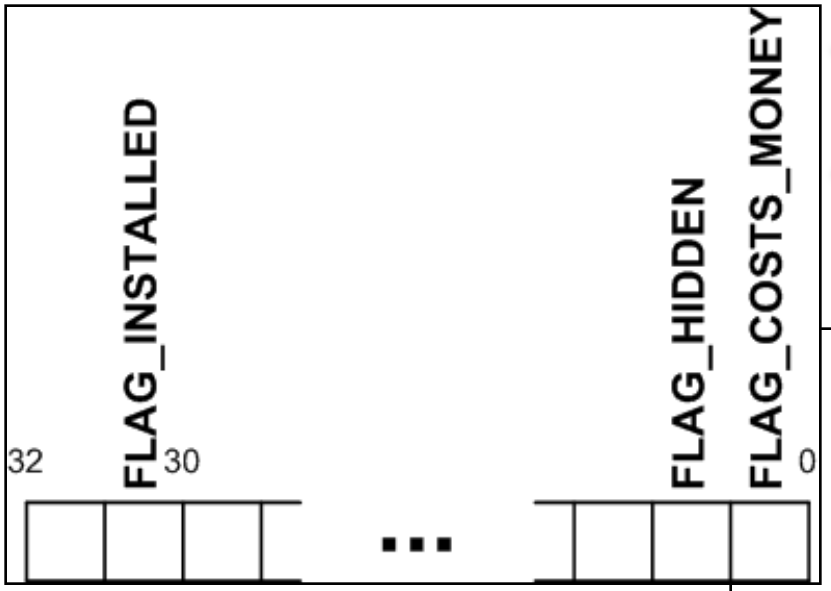
Permission Declaration

```
<!-- Allows an application to send SMS messages.
     <p>Protection level: dangerous
-->
<permission android:name="android.permission.SEND_SMS"
    android:permissionGroup="android.permission-group.SMS"
    android:label="@string/permlab_sendSms"
    android:description="@string/permdesc_sendSms"
    android:permissionFlags="costsMoney"
    android:protectionLevel="dangerous" />
```

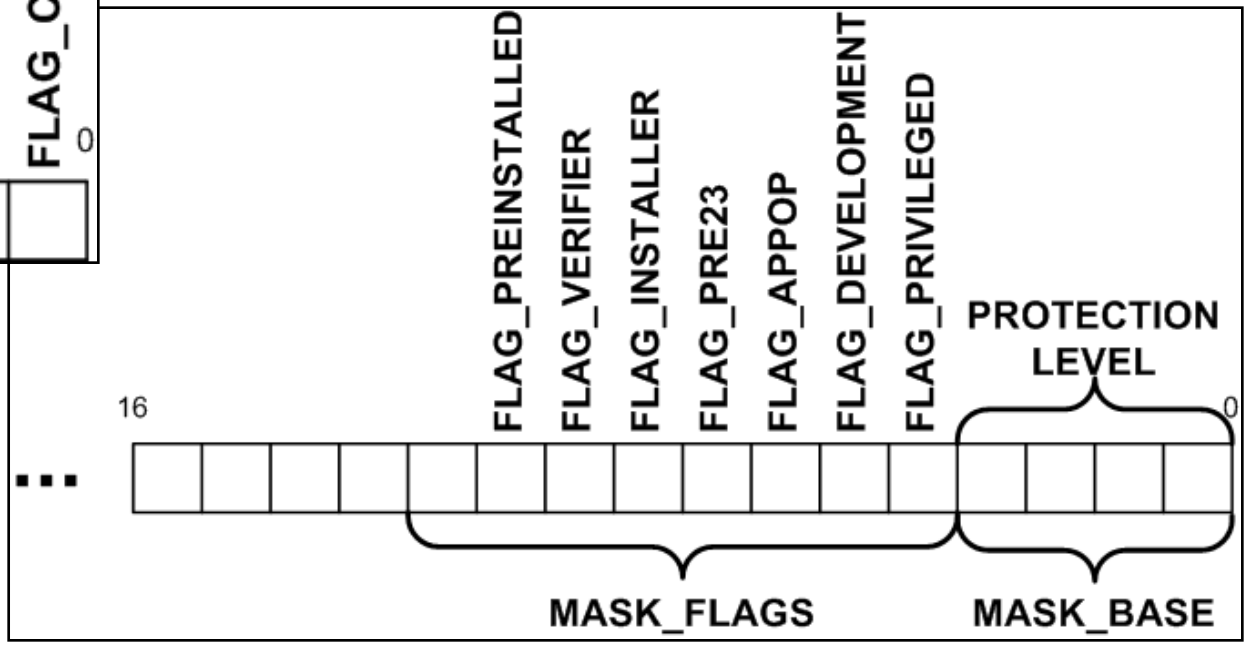
Permission Declaration

```
<!-- Allows an application to send SMS messages.
     <p>Protection level: dangerous
-->
<permission android:name="android.permission.SEND_SMS"
    android:permissionGroup="android.permission-group.SMS"
    android:label="@string/permlab_sendSms"
    android:description="@string/permdesc_sendSms"
    android:permissionFlags="costsMoney"
    android:protectionLevel="dangerous" />
```

Implementation Details



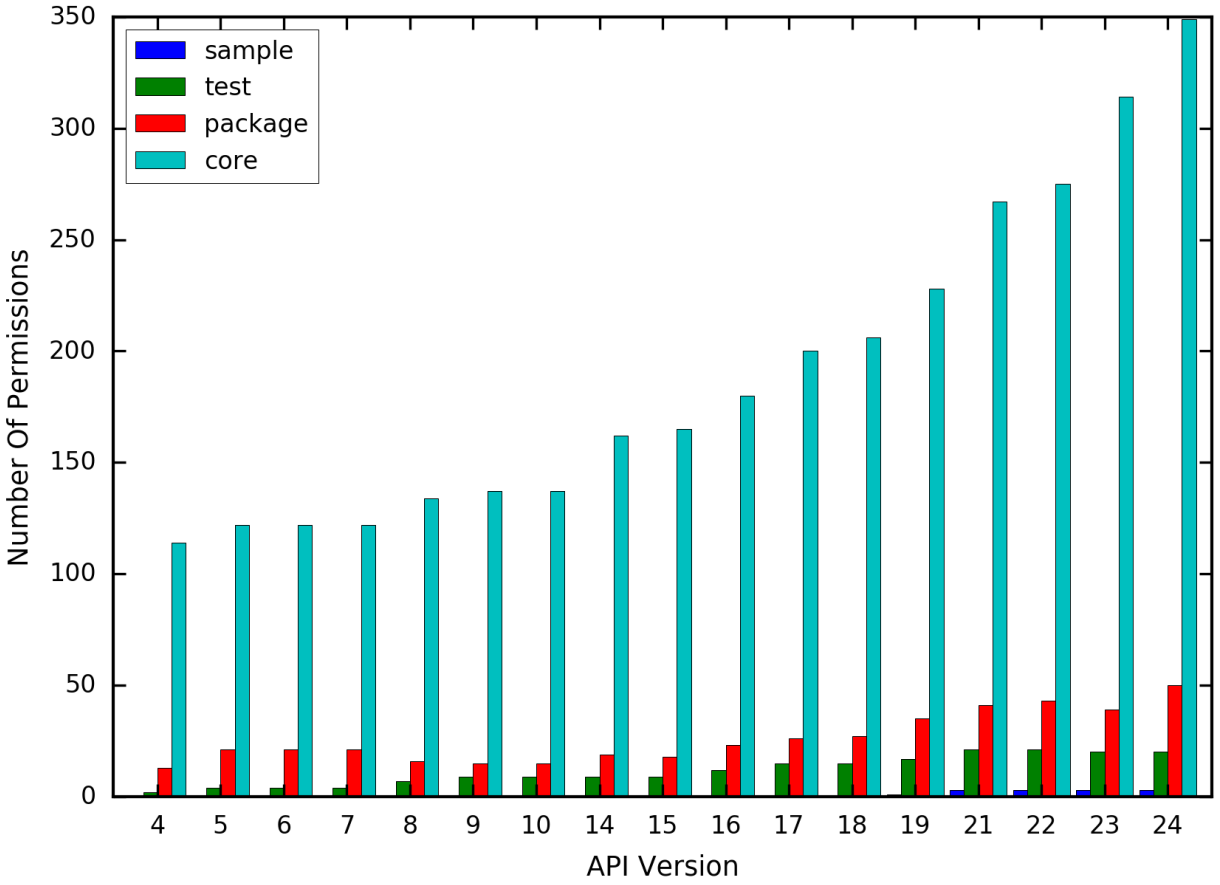
PermissionInfo.flags
parsed from
android:permissionFlags



PermissionInfo.protectionLevel
parsed from
android:protectionLevel

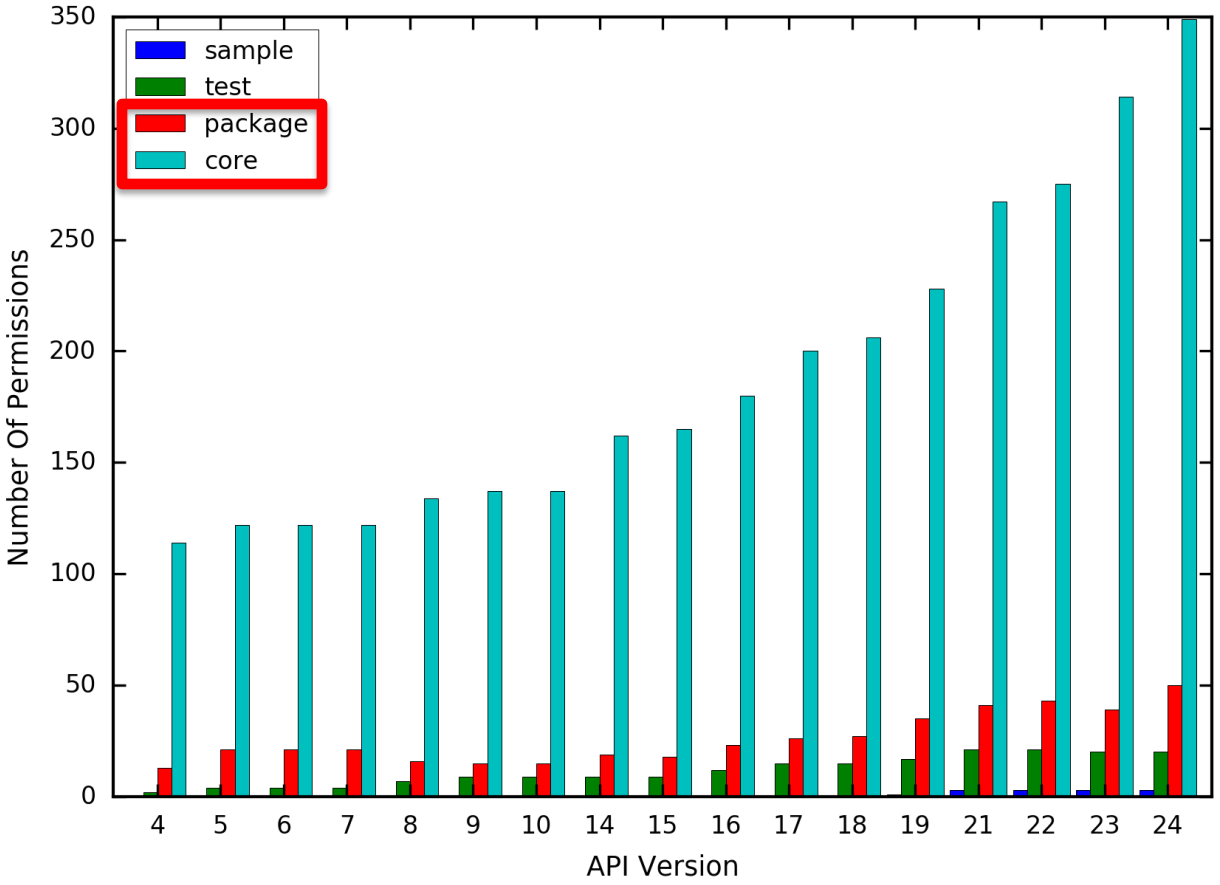
Quantitative Analysis

Permission Amount of Different Manifest Types



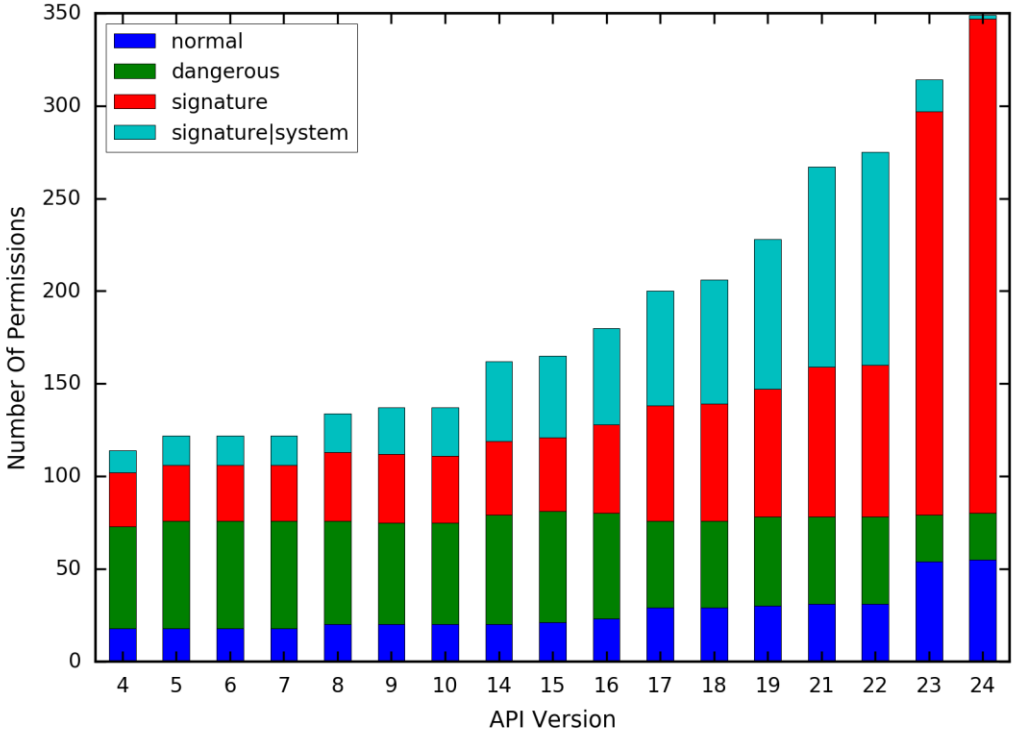
- Number of core permissions is considerably higher than others
- Permission number grows:
  - New platforms (TV, Auto)
  - New packages (Launcher3, etc)
  - Old packages are not removed (e.g., Launcher2)

Permission Amount of Different Manifest Types

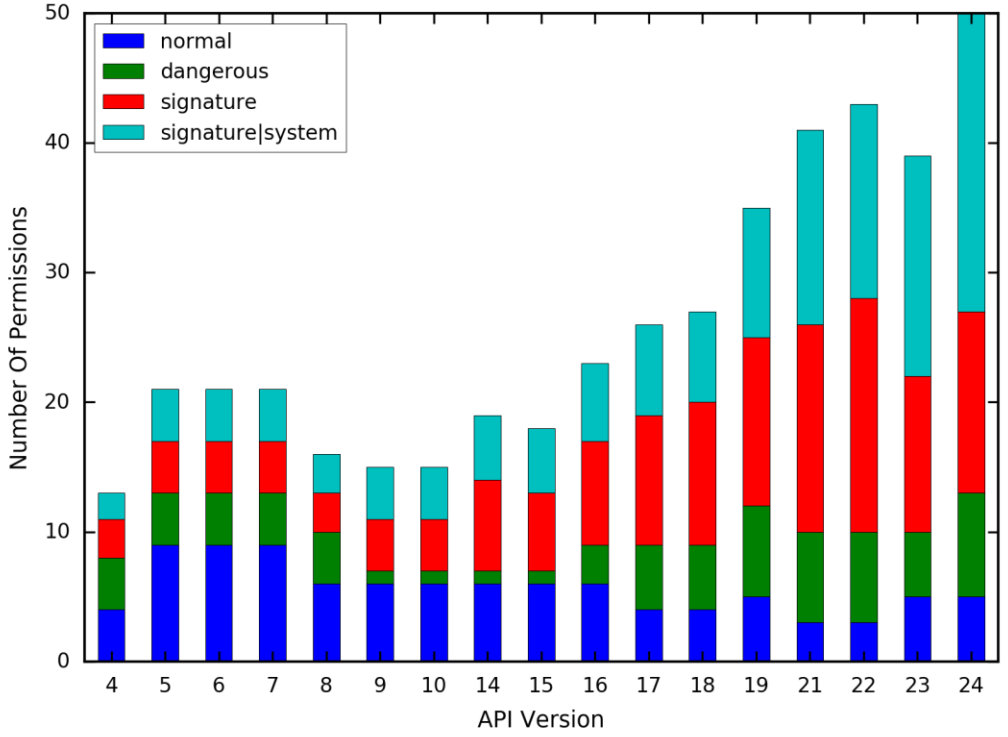


- Number of core permissions is considerably higher than others
- Permission number grows:
  - New platforms (TV, Auto)
  - New packages (Launcher3, etc)
  - Old packages are not removed (e.g., Launcher2)

Permission Number of Different Protection Levels

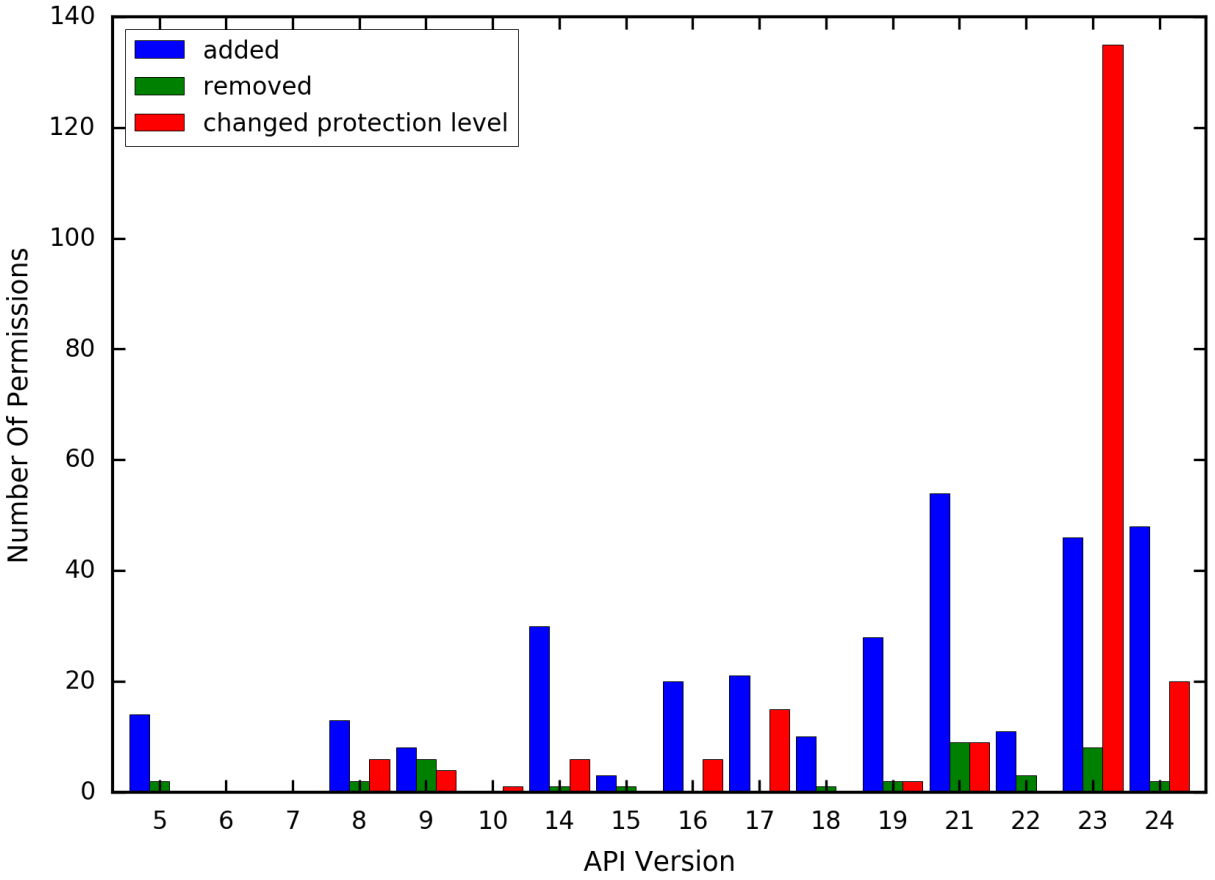


Core



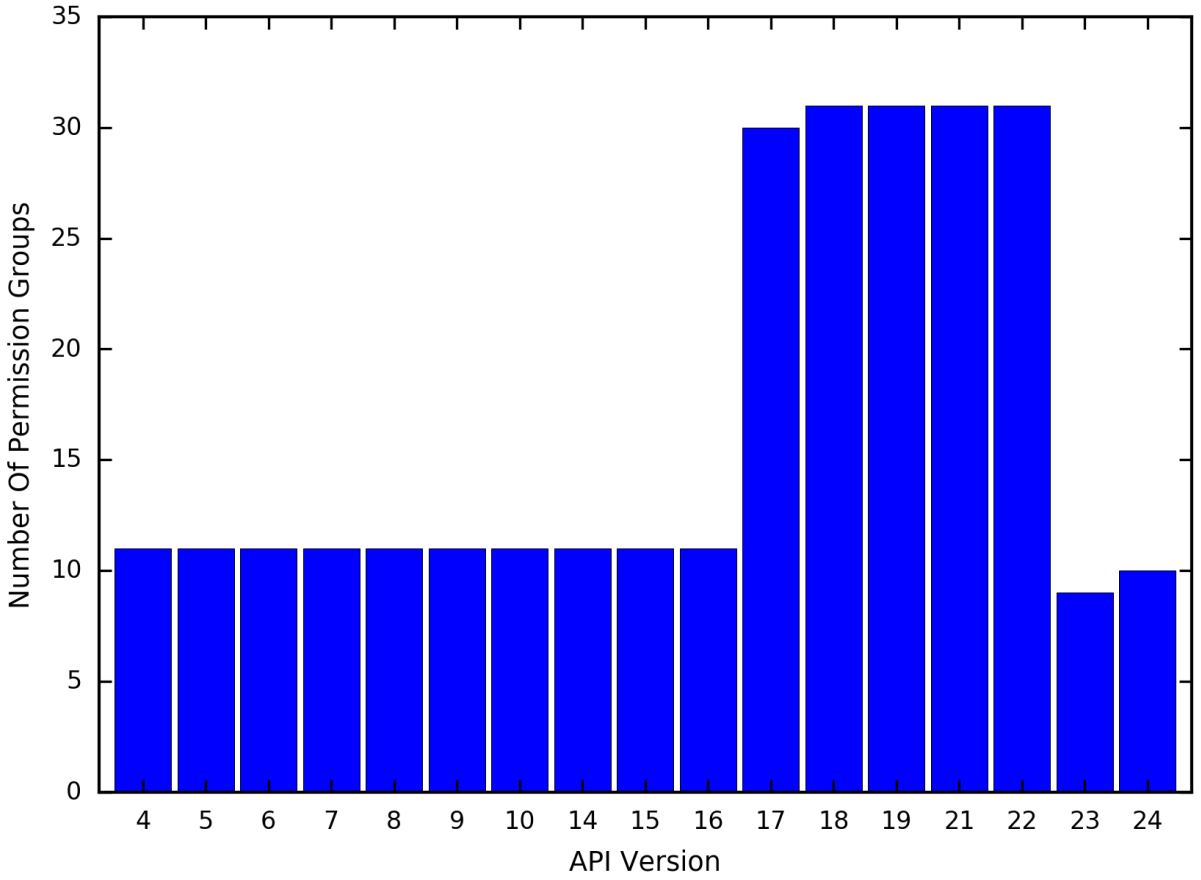
Package

Analysis of Permission Number Changes



- Android 6.0 (API 23) – signature|system is deprecated

Amout of Permission Groups

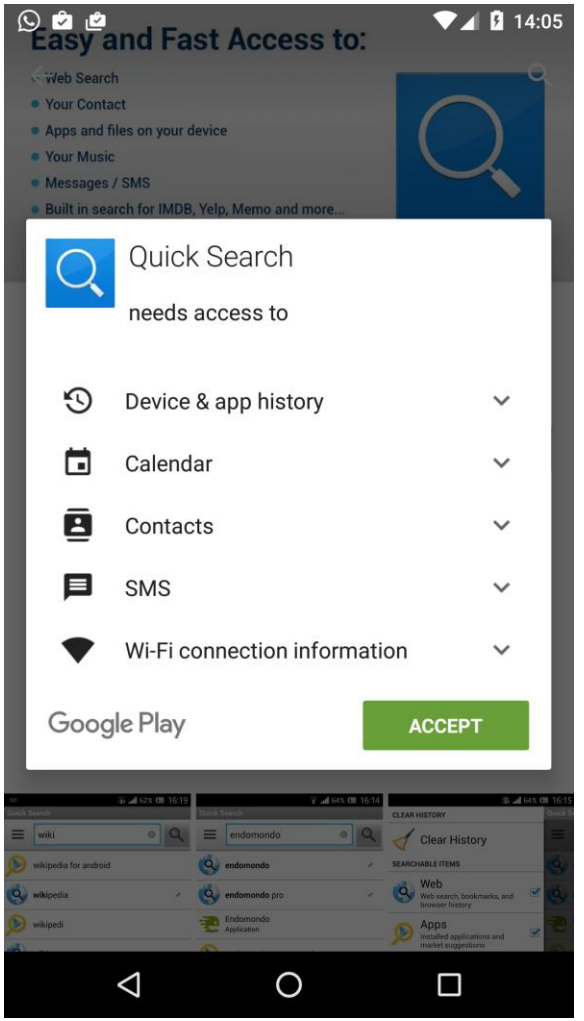


- Android 6.0 (API 23) – permissions are granted on **per group** basis

Qualitative Analysis

Important Changes

- Runtime permissions
  - User can revoke runtime permissions
  - Granted per permission group
- Not uniform behavior of apps:
  - Backward compatibility of old apps with the new platform
    - Old style installation process (all permissions are granted)
    - Permissions are granted and revoked through AppOps system
    - Only core permissions are “truly” runtime
  - Forward compatibility of new apps with older platforms
    - Developers must add additional checks for some permissions (e.g., WRITE_CALL_LOG, READ_CALL_LOG, READ_EXTERNAL_STORAGE)



Important Changes

- UID sharing:
  - Changes of runtime permission in one application influence on the permission state in other application
- Signature permissions can be requested by third-party apps:
  - Appop permissions can be granted by the user (PACKAGE_USAGE_STATS, SYSTEM_ALERT_WINDOW, WRITE_SETTINGS)
  - Development permissions can be granted to third-party applications through “pm grant” command
  - Permissions with FLAG_PRE23 set are granted automatically to apps with the target SDK level below 23
- Some dangerous permissions are now normal:
  - Some highly sensitive dangerous permissions are now normal (e.g., INTERNET, NFC, BLUETOOTH, etc.)

Summary

- Permission system is far from being stable
- Amount of permissions grows with every new release
- Considerable changes in Android Marshmallow
- Permission changes protection level often => security researchers must acknowledge this in their tools

Thank you!

معهد قطر لبحوث الحوسبة
Qatar Computing Research Institute

جامعة حمد بن خليفة
HAMAD BIN KHALIFA UNIVERSITY