

# Permissions

## GOOGLE

### Core app quality

- SC-P1
- SC-P2
- SC-P3
- SC-P4
- SC-P5
- SC-AC3

### App Security Best Practices

- Provide the right permissions
- Provide the right permissions  
Use intents to defer permissions
- Provide the right permissions  
Share data securely across apps

## Open Web Application Security Project (OWASP)

### Mobile Application Security Verification Standard (MASVS)

6.1 MSTG-PLATFORM-1

### Application Security Verification Standard 4.0.3 (ASVS)

V10.2 Malicious Code Search

## European Telecommunications Standards Institute (ETSI)

### ETSI TS 103 732

8.1.4 Security Management (FMT)  
FMT\_SMF.1\_Permissions  
Specification of Management Functions

## MITRE

### Application Developer Guidance

T1626  
Abuse Elevation Control Mechanism

## ioXt Alliance

### Mobile Application Profile

4.6. Security by Default SD113

## US National Institute of Standards and Technology (NIST)

### NIST Special Publication 800-190

4.5.5 Host file system tampering

## National Information Assurance Partnership (NIAP)

### Requirements for Vetting Mobile Apps from the Protection Profile for Application Software

- Secure by Default Configuration  
FMT\_CFG\_EXT.1.2
- User Consent for Transmission of Personally Identifiable Information  
FPR\_ANO\_EXT.1.1
- Anti-Exploitation Capabilities  
FPT\_AEX\_EXT.1.2

## Department for Digital, Culture, Media & Sport (DCMS)

### Code of practice for app store operators and app developers

- 2. Ensure apps adhere to baseline security and privacy requirements
- 2. Ensure apps adhere to baseline security and privacy requirements
- 5. Provide important security and privacy information to users in an accessible way

## Permissions

### 1. MITRE

#### 1.1. Application Developer Guidance

**Link:** <https://attack.mitre.org/mitigations/M1013/>

##### 1.1.1. T1626 Abuse Elevation Control Mechanism

Applications very rarely require administrator permission. Developers should be cautioned against using this higher degree of access to avoid being flagged as a potentially malicious application.

### 2. ioXt Alliance

#### 2.1. Mobile Application Profile

**Link:** [https://static1.squarespace.com/static/5c6dbac1f8135a29c7fbb621/t/604aa3fa668a8e3b50630433/1615504379349/Mobile\\_Application\\_Profile.pdf](https://static1.squarespace.com/static/5c6dbac1f8135a29c7fbb621/t/604aa3fa668a8e3b50630433/1615504379349/Mobile_Application_Profile.pdf)

##### 2.1.1. 4.6. Security by Default SD113

Only necessary permissions are requested.

### 3. US National Institute of Standards and Technology (NIST)

#### 3.1. NIST Special Publication 800-190

**Link:** <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-190.pdf>

##### 3.1.1. 4.5.5 Host file system tampering

Ensure that containers are run with the minimal set of file system permissions required. Very rarely should containers mount local file systems on a host. Instead, any file changes that containers need to persist to disk should be made within storage volumes specifically allocated for this purpose. In no case should containers be able to mount sensitive directories on a host's file system, especially those containing configuration settings for the operating system. Organizations should use tools that can monitor what directories are being mounted by containers and prevent the deployment of containers that violate these policies.

### 4. National Information Assurance Partnership (NIAP)

#### 4.1. Requirements for Vetting Mobile Apps from the Protection Profile for Application Software

**Link:** [https://www.niap-ccevs.org/MMO/PP/394.R/pp\\_app\\_v1.2\\_table-reqs.htm](https://www.niap-ccevs.org/MMO/PP/394.R/pp_app_v1.2_table-reqs.htm)

##### 4.1.1. Secure by Default Configuration FMT\_CFG\_EXT.1.2

The application shall be configured by default with file permissions which protect it and its data from unauthorized access.

Application Note: The precise expectations for file permissions vary per platform but the general intention is that a trust boundary protects the application and its data.

##### 4.1.2. User Consent for Transmission of Personally Identifiable Information FPR\_ANO\_EXT.1.1

The application shall [selection:

not transmit PII over a network ,

require user approval before executing [assignment: list of functions that transmit PII over a network ]]. Application Note: This requirement applies only to PII that is specifically requested by the application; it does not apply if the user volunteers PII without prompting from the application into a general (or inappropriate) data field. A dialog box that declares intent to send PII presented to the user at the time the application is started is sufficient to meet this requirement.

##### 4.1.3. Anti-Exploitation Capabilities FPT\_AEX\_EXT.1.2

The application shall [selection:

not allocate any memory region with both write and execute permissions ,allocate memory regions with write and execute permissions for only [assignment: list of functions performing just-in-time compilation]]. Application Note: Requesting a memory mapping with both write and execute permissions subverts the platform protection provided by DEP. If the application performs no just-in-time compiling, then the first selection must be chosen.

### 5. Department for Digital, Culture, Media & Sport (DCMS)

#### 5.1. Code of practice for app store operators and app developers

**Link:** <https://www.gov.uk/government/publications/code-of-practice-for-app-store-operators-and-app-developers/code-of-practice-for-app-store-operators-and-app-developers>

##### 5.1.1. 2. Ensure apps adhere to baseline security and privacy requirements

2.2 Developers shall ensure that the primary function of an app operates if a user chooses to disable its optional functionality and permissions  
2.2.1 If the user isn't presented with any optional functionalities, developers shall ensure that their app only requires the enabled functions and permissions necessary to operate.

#### 5.1.2. 2. Ensure apps adhere to baseline security and privacy requirements

2.3 Developers should not request permissions and privileges which are not functionally required by the app.  
2.3.1 Developers shall share the permissions and privileges requested by the app in the app manifest with the App Store Operator, to allow for this to be cross-checked.

#### 5.1.3. 5. Provide important security and privacy information to users in an accessible way

5.4 Developers shall provide information about the permissions which an app may request, such as access to contacts, location and the device's microphone, along with justifications for why each of these permissions are needed. This information shall be provided to app stores and any users who install the app without an app store. Operators shall display this information for all apps on their app store prior to purchase and download.

## 6. European Telecommunications Standards Institute (ETSI)

### 6.1. ETSI TS 103 732

**Link:** [https://www.etsi.org/deliver/etsi\\_ts/103700\\_103799/103732/01.01.01\\_60/ts\\_103732v010101p.pdf](https://www.etsi.org/deliver/etsi_ts/103700_103799/103732/01.01.01_60/ts_103732v010101p.pdf)

#### 6.1.1. 8.1.4 Security Management (FMT) FMT\_SMF.1\_Permissions Specification of Management Functions

FMT\_SMF.1.1/Permissions The TSF shall be capable of performing the following management functions by the human user:

- view permissions granted to an App; and
- revoke permission from a Downloaded App or Pre-installed non-system permission App to have read and/or write access to an object; and
- revoke permission from a Pre-installed system permission App to have read and/or write access to an object if the human user has granted this permission themselves; and
- grant permission to an App to have read and/or write access to an object; and
- select [selection: charge only mode by default, file transfer mode, [assignment: other wired charging mode]] when the TOE is connected via charging interface such as USB interface to a computer/laptop; and
- grant/revoke permission to/from a Downloaded App or Pre-installed non-system permission App to have access to accessibility service; and
- grant/revoke permission to/from a Downloaded App or Pre-installed non-system permission App to have access to device notification; and
- [assignment: list of management functions provided by the TSF].

NOTE 1: Object is defined in FDP\_ACC.1.1/Permissions.

NOTE 2: The selection for charging mode is per connection basis, and the ""charge only mode"" is the default mode when the human user does not select any mode.

## 7. Open Web Application Security Project (OWASP)

### 7.1. Mobile Application Security Verification Standard (MASVS)

**Link:** <https://github.com/OWASP/owasp-masvs/releases/tag/v1.4.2>

#### 7.1.1. 6.1 MSTG-PLATFORM-1

The app only requests the minimum set of permissions necessary.

### 7.2. Application Security Verification Standard 4.0.3 (ASVS)

**Link:** <https://raw.githubusercontent.com/OWASP/ASVS/v4.0.3/4.0/OWASP%20Application%20Security%20Verification%20Standard%204.0.3-en.pdf>

#### 7.2.1. V10.2 Malicious Code Search

10.2.2 Verify that the application does not ask for unnecessary or excessive permissions to privacy related features or sensors, such as contacts, cameras, microphones, or location.

## 8. GOOGLE

### 8.1. Core app quality

**Link:** <https://developer.android.com/docs/quality-guidelines/core-app-quality>

#### 8.1.1. SC-P1

The app requests only the absolute minimum number of permissions that it needs to support its use case at hand. For some permissions such as location, use coarse location in place of fine location if possible.

#### 8.1.2. SC-P2

The app requests permission to access sensitive data (such as SMS, Call Log, or Location) or services that cost money (such as Dialer or SMS) only when directly related to the core use cases of the apps. Implications related to these permissions should be prominently disclosed to the user.

Depending on how you are using the permissions, there might be an alternative way to fulfill your app's use case without relying on access to sensitive information. For example, instead of requesting permissions related to a user's contacts, it may be more appropriate to request access by using an implicit intent.

#### 8.1.3. SC-P3

The app requests runtime permissions in context, when the functionality is requested, rather than upfront during app startup.

#### 8.1.4. SC-P4

The app clearly conveys why certain permissions are needed or follow the recommended flow to explain why it needs a permission.

#### 8.1.5. SC-P5

The app should gracefully degrade when users deny or revoke a permission. The app should not prevent the user from accessing the app altogether.

#### 8.1.6. SC-AC3

All components that share content between your apps use `android:protectionLevel=""signature""` for custom permissions. This includes activities, services, broadcast receivers, and especially content providers. Apps should not rely on accessing a list of installed packages. The access has been restricted beginning in Android 11.

## 8.2. App Security Best Practices

**Link:** <https://developer.android.com/topic/security/best-practices>

### 8.2.1. Provide the right permissions

Your app should request only the minimum number of permissions necessary to function properly. When possible, your app should relinquish some of these permissions when they're no longer needed.

### 8.2.2. Provide the right permissions Use intents to defer permissions

Whenever possible, don't add a permission to your app to complete an action that could be completed in another app. Instead, use an intent to defer the request to a different app that already has the necessary permission.

The following example shows how to use an intent to direct users to a contacts app instead of requesting the `READ_CONTACTS` and `WRITE_CONTACTS` permissions:

In addition, if your app needs to perform file-based I/O—such as accessing storage or choosing a file—it doesn't need special permissions because the system can complete the operations on your app's behalf. Better still, after a user selects content at a particular URI, the calling app gets granted permission to the selected resource.

### 8.2.3. Provide the right permissions Share data securely across apps

Follow these best practices in order to share your app's content with other apps in a more secure manner:

Enforce read-only or write-only permissions as needed.

Provide clients one-time access to data by using the `FLAG_GRANT_READ_URI_PERMISSION` and `FLAG_GRANT_WRITE_URI_PERMISSION` flags.

When sharing data, use `content://` URIs, not `file://` URIs. Instances of `FileProvider` do this for you.

The following code snippet shows how to use URI permission grant flags and content provider permissions to display an app's PDF file in a separate PDF Viewer app: