セキュリティフォーラム 2024

『Androidアプリのセキュア設計・セキュアコーディングガイド』のご紹介

一般社団法人日本スマートフォンセキュリティ協会（JSSEC）
宮崎 力

自己紹介

宮崎　力

日本スマートフォンセキュリティ協会(JSSEC)

・セキュアコーディングWG リーダー

株式会社ラック

• テクノロジーリスクコンサルティング部

JSSEC

一般社団法人 日本スマートフォンセキュリティ協会

Japan Smartphone Security Forum

「スマートフォンの安全な利活用を図り普及を促進するために、2011年5月25日に任意団体としてスタートし、2012年4月1日より一般社団法人として活動している団体です。」

幹事会員（11社）

株式会社NTTドコモ

株式会社EMPRESS SOFTWARE JAPAN

KDDI株式会社

サイバートラスト株式会社

株式会社SHIFT SECURITY

ソフトバンク株式会社

株式会社TwoFive

学校法人 東京電機大学

株式会社日本総合研究所

日本ヒューレット・パッカード合同会社

株式会社ラック

正会員（46社）

ア カ サ タ ナ ハ マ ヤ ラ

ア アイマトリックス株式会社

株式会社アズジェント

株式会社アナハイム・テクノロジー

アルプスシステムインテグレーション株式会社

株式会社アンラボ

イ インヴェンティット株式会社

Intertrust Technologies Japan株式会社

ウ 株式会社ウィザース

エ NECフィールディング株式会社

エヌ・ティ・ティ・コミュニケーションズ株式会社

株式会社NTTデータグループ

株式会社NTTデータ MSE

株式会社NTTデータ NJK

セキュアコーディングガイドのご紹介

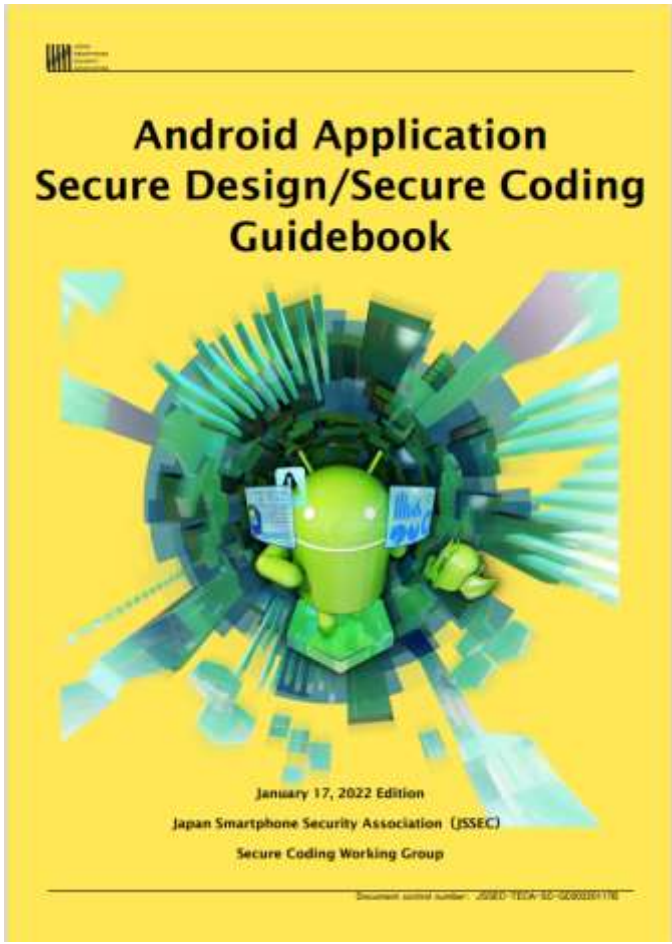
セキュア コーディング ガイド

JSSEC技術部会が発行している、Android アプリケーション開発者向けのセキュア設計、セキュアコーディングのノウハウをまとめた Tips 集。

日本語版をはじめ英語版も公開されている。

JSSECの公式サイトより無料で閲覧・ダウンロードすることができる。

豊富なサンプルコードも。



改訂履歴

版数	版名	対象Android
第1版	2012年6月版	
第2版	2012年11月版	
第3版	2013年4月版	
第4版	2014年7月版	
第5版	2015年6月版	
第6版	2016年2月版	
第7版	2016年9月版	
第8版	2017年2月版	
第9版	2018年2月1日版	Android 8
第10版	2018年9月1日版	Android 9
第11版	2019年9月1日版	Android 10
	2019年12月1日版	Android 10
第12版	2020年11月1日版	Android 11
第13版	2021年10月19日版	Android 12
	2022年1月17日版	Android 12
第14版	2022年8月29日版	Android 13

ANDROID 14

Android 14がリリース

ガイド改定

2024年2月29日 最新版コーディングガイド公開

ホーム > 実践ノウハウ・調査結果 > 技術部会

実践ノウハウ・調査結果　技術者向け

発信元：技術部会　書いた人:技術部会

『Android アプリのセキュア設計・セキュアコーディングガイド』（以下 本ガイド）の15版目の改定版である2024年2月29日版を公開しました。

2024年2月29日　2024年2月28日

2024年2月29日

『Android アプリのセキュア設計・セキュアコーディングガイド』（以下 本ガイド）の15版目の改定版である2024年2月29日版を公開しました。

・『Android アプリのセキュア設計・セキュアコーディングガイド』【2024年2月29日版】

・「サンプルコード一式」　【2024年2月29日版】

何が変わった？

最新版コーディングガイドの改定箇所。
Android 14における変更点のうちセキュリティに関するものは全て網羅。

下記の新しい記事を追加いたしました

- 4.1.3.9. 暗黙的 *Intent* とペンディング *Intent* の制限
- 4.4.3.3. 必須となった *Service* タイプ指定
- 4.4.3.4. バックグラウンドからのアクティビティの起動に関する追加の制限
- 4.6.3.9. *Android 14（API Level 34）* における画像と動画の部分的アクセス
- 4.6.3.10. *DCL (Dynamic Code Loading)* の安全性の強化
- 4.6.3.11. *Zip* ファイルの *Path Traversal* 対策
- 4.10.3.5. 進行中を示す *Notification* の動作の変更
- 5.2.3.12. インストール可能な最小対象 *API Level*
- 5.2.3.13. メディア所有者のパッケージ名

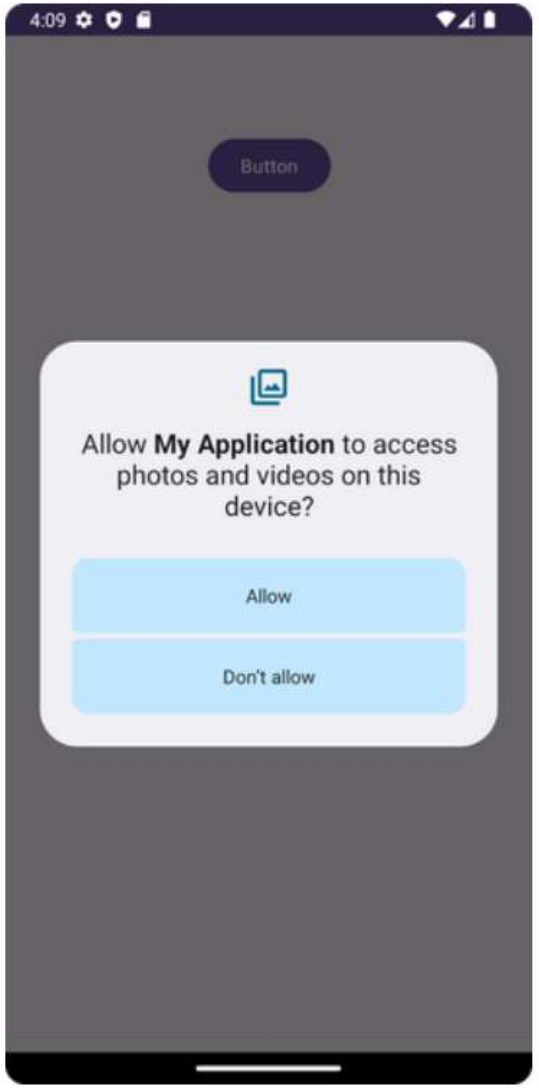
下記の構成・内容を見直し拡充いたしました

- 4.1.3.4. ルート *Activity* について
- 4.2.3.8. 動的 *Broadcast Receiver* の安全性の強化

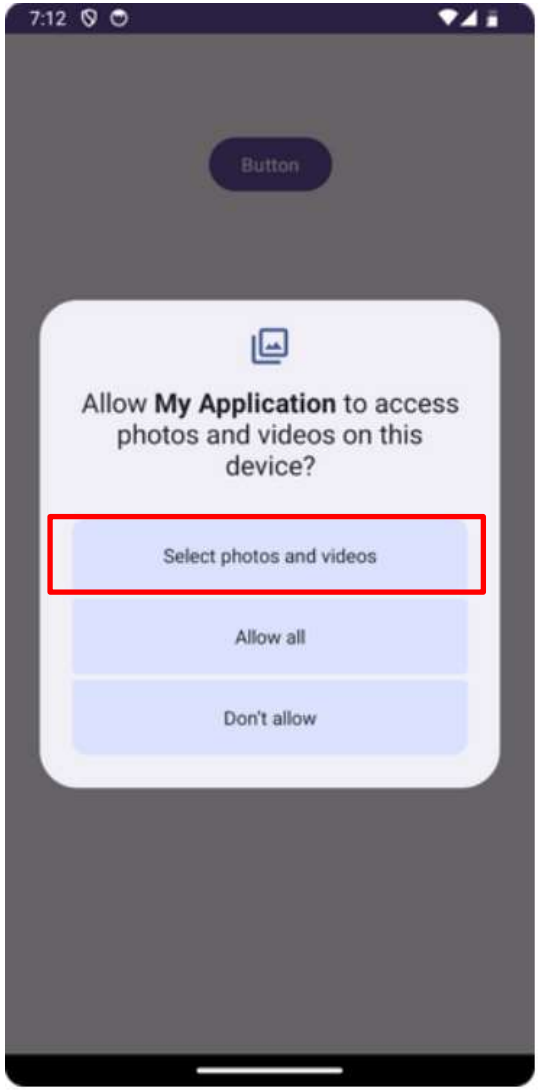
画像と動画の部分的アクセス

部分的アクセスって？

Android 13



Android 14



部分的アクセス

Android 14からファイル単位のアクセス許可が可能に。

ユーザー視点で見ると、より柔軟な運用が可能に。

開発者視点で見ると、追加の権限が必要になった。

追加で必要な権限

Android 13

音楽	READ_MEDIA_AUDIO
動画	READ_MEDIA_VIDEO
ピクチャ	READ_MEDIA_IMAGES

Android 14

音楽	READ_MEDIA_AUDIO
動画	READ_MEDIA_VIDEO READ_MEDIA_VISUAL_USER_SELECTED, ACCESS_MEDIA_LOCATION
ピクチャ	READ_MEDIA_IMAGES READ_MEDIA_VISUAL_USER_SELECTED, ACCESS_MEDIA_LOCATION

権限リクエスト方法はいつも通り

AndroidManifest.xml

```
<uses-permission android:name="android.permission.READ_MEDIA_IMAGES" />
<uses-permission android:name="android.permission.READ_MEDIA_VISUAL_USER_SELECTED" />
<uses-permission android:name="android.permission.ACCESS_MEDIA_LOCATION" />
```

MainActivity.java

```
String[] PERMISSIONS = {
    Manifest.permission.READ_MEDIA_IMAGES,
    Manifest.permission.READ_MEDIA_VISUAL_USER_SELECTED,
    Manifest.permission.ACCESS_MEDIA_LOCATION,
};

protected void onCreate(Bundle savedInstanceState) {
    // 中略
    for (String perm : PERMISSIONS) {
        if (ActivityCompat.checkSelfPermission(this, perm)
            != PackageManager.PERMISSION_GRANTED) {
            requestPermissionsLauncher.launch(PERMISSIONS);
        }
    }
```

修正は難しくない

メディア所有者のパッケージ名

メディアファイルをクエリ

共有ストレージにあるメディアファイルを全検索し、ヒットしたファイルの所有者「OWNER_PACKAGE_NAME」を表示するプログラムを考えてみる

まずはメディアファイルを全検索

```
cursor = contentResolver.query(
        MediaStore.Audio.Media.EXTERNAL_CONTENT_URI,
        null,null,null,null);
```

ヒットしたら所有者をLogcatで出力

```
Log.d(TAG,
cursor.getString(cursor.getColumnIndex(MediaStore.Audio.Media.OWNER_PACKAGE_NAME)));
```

出力結果比較

Android 13

```
D  voicerecorder.audiorecorder.voice
```

Android 14

```
E  Unknown message received from debugger! ''
```

Android 14ではパッケージ名が出力されていない。

そのメディアファイルの所有者がわかると、ユーザーがどんなアプリをインストールしているかわかってしまうため。

インストール済みアプリの情報はユーザーの個人情報および機密情報とみなされるようになった。

取り上げた2つの変更点から言えること

「画像と動画の部分的アクセス」
「メディア所有者のパッケージ名」

メディアファイルにアクセスする場合はMedia Store API+ContentResolverを使用すること。

Android 9まで

```
try {
            FileOutputStream fileOutputStream = new FileOutputStream(file);
            Bitmap bmp = BitmapFactory.decodeResource(getResources(), R.drawable.picture);
            bmp.compress(Bitmap.CompressFormat.JPEG, 70, fileOutputStream);
} catch (FileNotFoundException e) {
```

Android 10以降

```
ContentResolver resolver = getApplicationContext().getContentResolver();
ContentValues values = new ContentValues();
values.put(MediaStore.Images.Media.DISPLAY_NAME, display_name);
Uri collection = MediaStore.Images.Media.getContentUri(MediaStore.VOLUME_EXTERNAL_PRIMARY);
Uri item = resolver.insert(collection, values);

try (OutputStream outstream = resolver.openOutputStream(item)) {
            Bitmap bmp = BitmapFactory.decodeResource(getResources(), R.drawable.picture);
            bmp.compress(Bitmap.CompressFormat.JPEG, 70, outstream);
}
```

修正範囲が大きいのは見ての通り。
しかしOSの仕様であること、今後の保守性を考慮するなら、なるはやで仕様追従を。

ファイルアクセス方法をいまいちど整理。

現在に至るまで細かい仕様変更が繰り返されているが大枠は変わっていない。

判断に迷ったら以下の基準でアクセス方法を決める。

	アクセス方法	必要な権限
自アプリでのみで使うファイルである	制限なし getExternalFilesDir()	なし
他のアプリと共有するファイルである	Storage Access Framework ＋ ContentResolver	なし
他のアプリと共有するメディアファイルである	MediaStore API ＋ ContentResolver	READ_MEDIA_AUDIO READ_MEDIA_VIDEO READ_MEDIA_IMAGES (READ_MEDIA_VISUAL_USER_SELECTED, ACCESS_MEDIA_LOCATION)

セキュアコーディングWGについて

4つの部会と主な活動

JSSEC
4つの部会

利用部会

パブリックリレーションズ部会

啓発事業部会

技術部会
- ネットワークWG
- セキュアコーディングWG　👈ここ
- マルウエア対策WG
- デバイス系WG
- メタバースセキュリティWG

執筆者募集

最新版である第15版は下記のメンバーにより制作。

制作

一般社団法人 日本スマートフォンセキュリティ協会 技術部会 セキュアコーディングWG

リーダー	宮崎 力	株式会社ラック
メンバー	塩田 明弘	株式会社NTTデータグループ
	本間 輝彰	KDDI 株式会社
	上松 晴信	KDDI 株式会社
	小笠原 徳彦	株式会社SHIFT SECURITY
	才田 好則	日本電気株式会社
	青柳 亨	日本電気株式会社

（執筆関係者、社名五十音順）

執筆者は常に募集しております。

問い合わせ窓口
https://www.jssec.org/contact

ご清聴ありがとうございました。