セキュリティフォーラム 2025

『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

• セキュリティインテグレーショングループ

セキュアコーディングガイドのご紹介

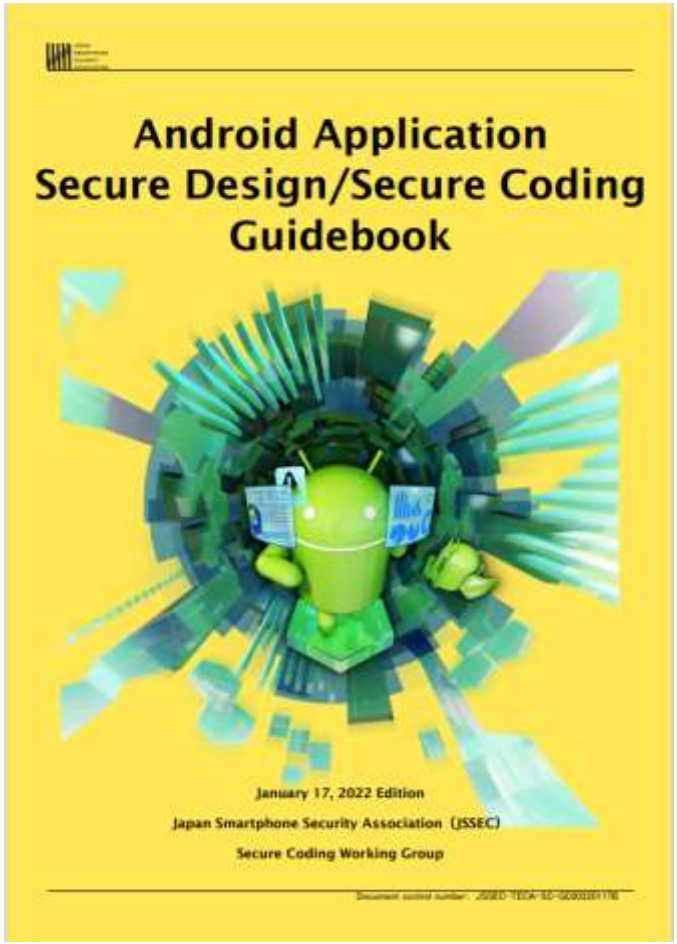
セキュア コーディング ガイド

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
第15版	2024年2月29日版	Android 14
第16版	2025年1月29日版	Android 15

ANDROID 15

2024年10月15日 Android 15がリリース

ガイド改定

2025年1月29日 最新版コーディングガイド公開

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』2025年1月29日版公開

2025年1月29日　2025年1月29日

報道関係各位

2025年1月29日

JSSEC、『Androidアプリのセキュア設計・セキュアコーディングガイド』
2025年1月29日版公開

一般社団法人日本スマートフォンセキュリティ協会

一般社団法人日本スマートフォンセキュリティ協会（JSSEC：会長 佐々木 良一）の技術部会 セキュアコーディングWG（リーダー 宮崎 力）は、2012年6月に公開した『Android アプリのセキュア設計・セキュアコーディングガイド』（以下 本ガイド）の16版目の改定版である2025年1月29日版を公開しました。

改定内容

最新版コーディングガイドの改定箇所。
セキュリティとプライバシーに関する更新箇所は全て網羅。

2025-1-29

下記の新しい記事を追加いたしました

- 4.1.3.11. Intent Filter 機能の拡張
- 4.1.3.12. Intentのセキュリティ強化
- 4.1.3.13. パッケージの停止状態の変更
- 4.3.3.1. コンテンツURIの権限管理
- 4.4.3.4. フォアグラウンド サービスの変更
- 4.6.3.6. Android 10以降のストレージアクセス (内部ストレージ・外部ストレージ)
- 4.6.3.7. Android 10以降のストレージアクセス (共有ストレージ)
- 4.6.3.10. 選択した写真へのアクセスについて、直近のユーザーの選択をクエリする
- 4.6.3.11. プライベート スペース
- 5.1.3.5. 認証情報マネージャーを自動入力と統合する
- 5.5.3.10. 画面録画の検出
- 5.5.3.11. 部分的画面共有
- 5.5.3.12. DNDモードのグローバル状態管理の変更点
- 5.6.3.7. バックアップデータの漏洩対策
- 5.6.3.8. ハードコードされた暗号シークレット

下記の構成・内容を見直し拡充いたしました

- 4.1.3.10. 安全なバックグラウンド アクティビティの起動
- 5.2.3.12. インストール可能な最小対象 API Level

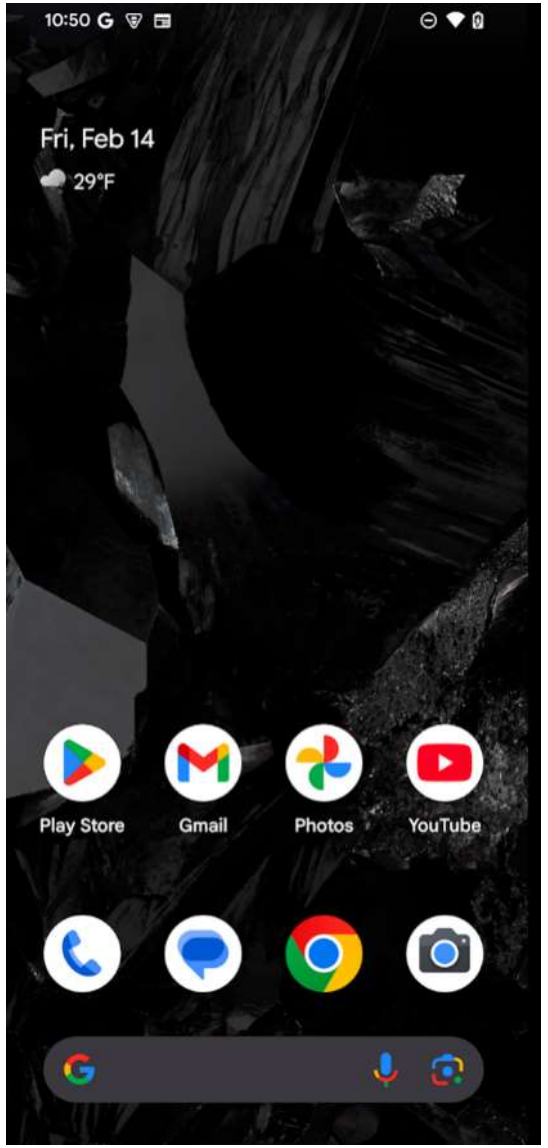
下記の記事を削除致しました

- 4.6.3.6. Android 10(API Level 29)における外部ストレージへのアクセスに関する仕様について
- 4.6.3.7. Android 11（API Level 30）における対象範囲別ストレージの適用について
- 4.6.3.8. Android 13（API Level 33）におけるメディアコレクション権限について
- 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

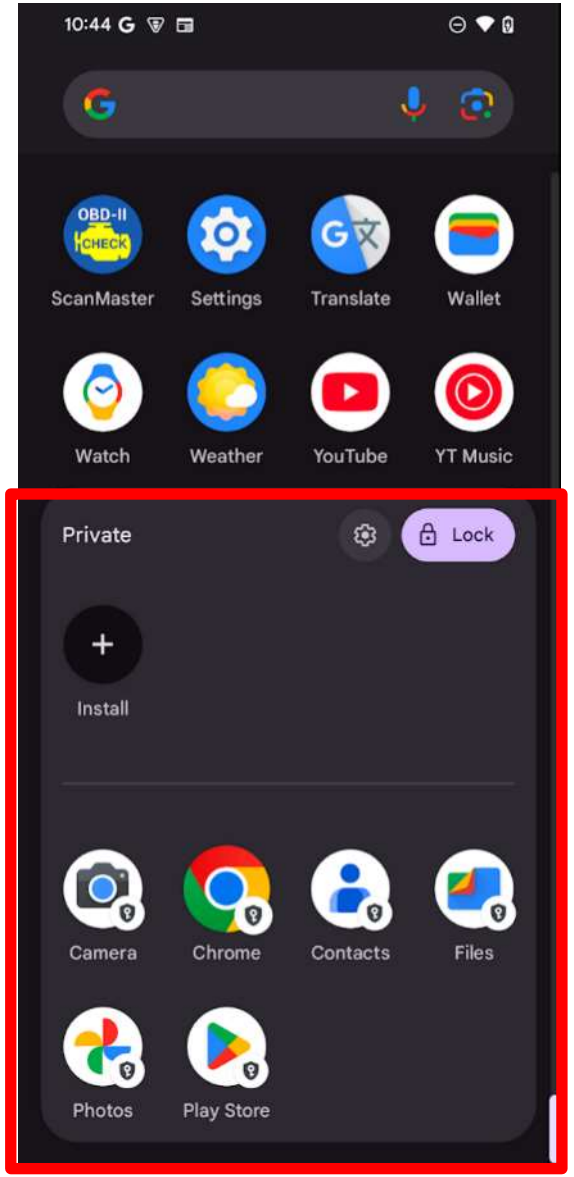
プライベートスペース

プライベートスペース

ホーム



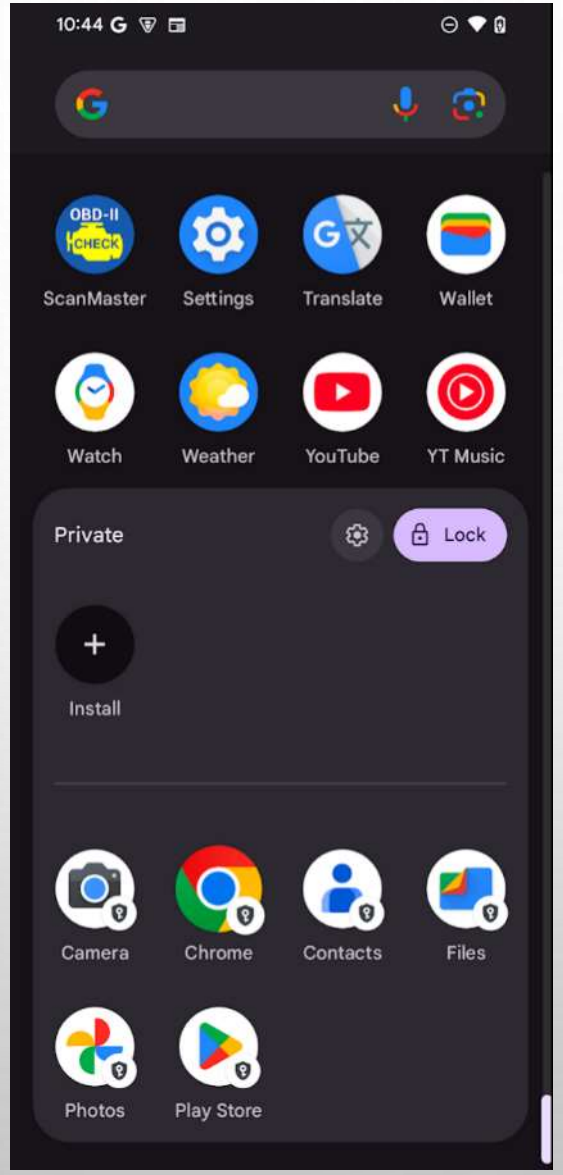
プライベートスペース



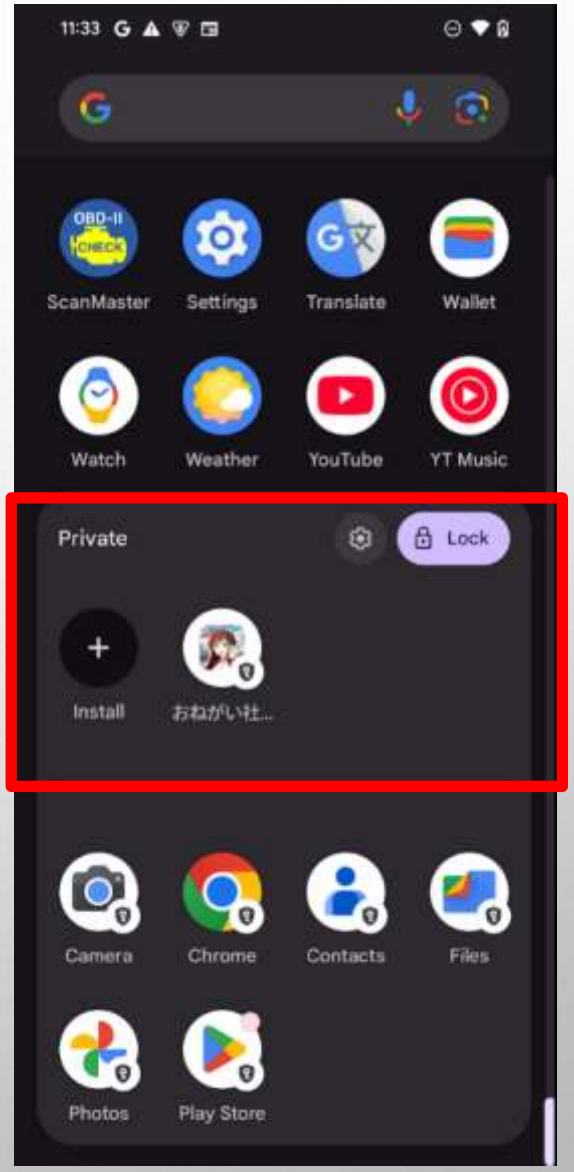
プライベートスペースにインストールしたアプリはプライベートスペースでしか見えない。

人に見られたくないアプリをプライベートスペースに集約するということが可能

デフォルト



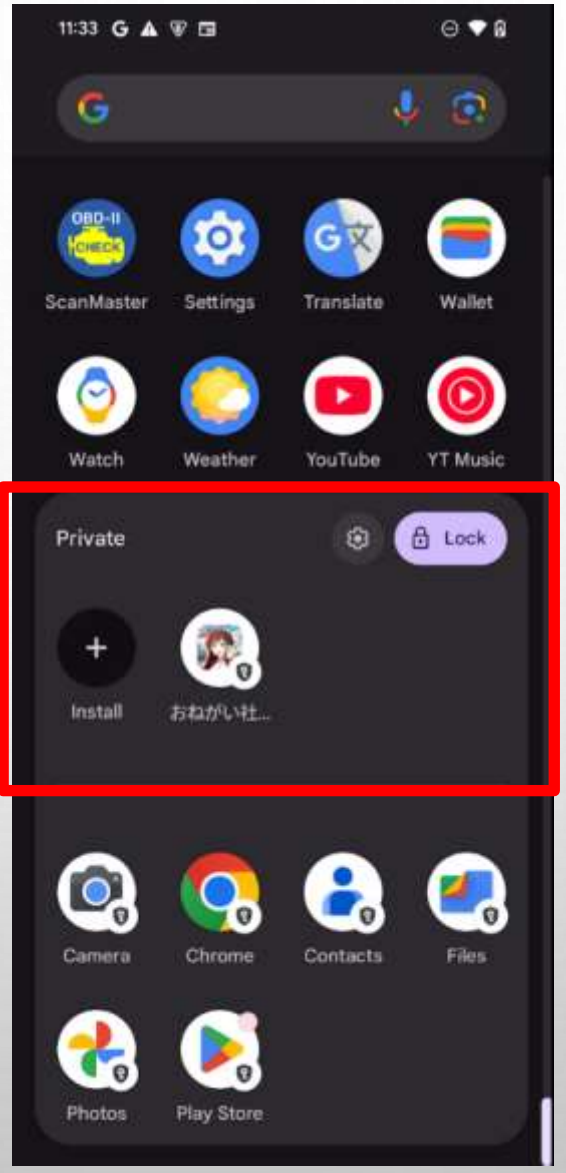
インストールしたアプリ



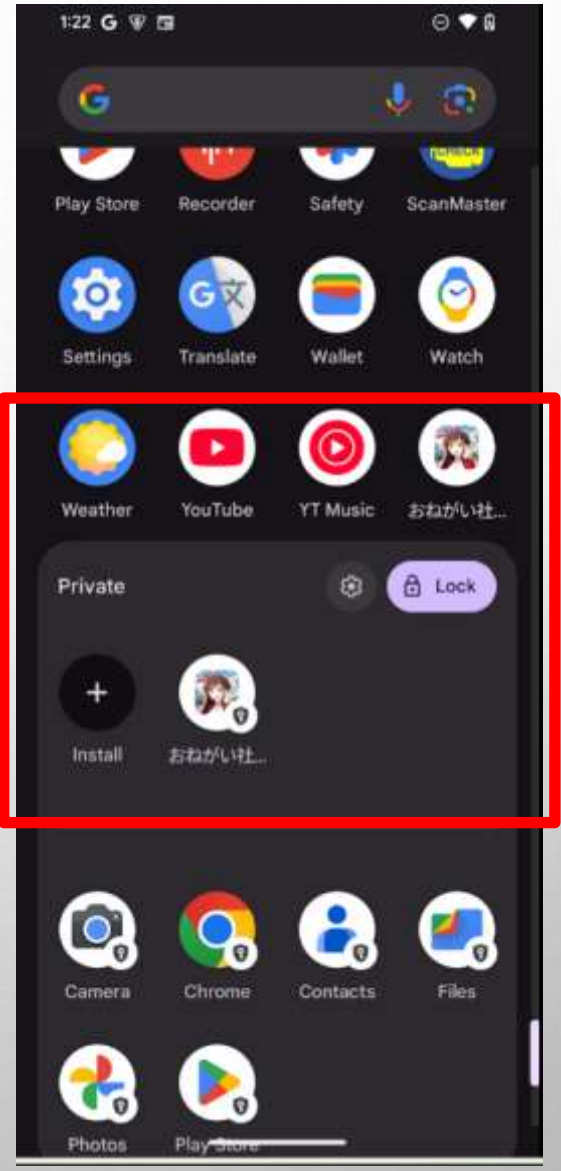
同一のアプリをメインスペースとプライベートスペースの両方にインストールすることもできる。

こうすると同一のアプリを複数アカウントで使い分けることが可能。

プライベートスペースのみインストール



メインスペースにもインストール



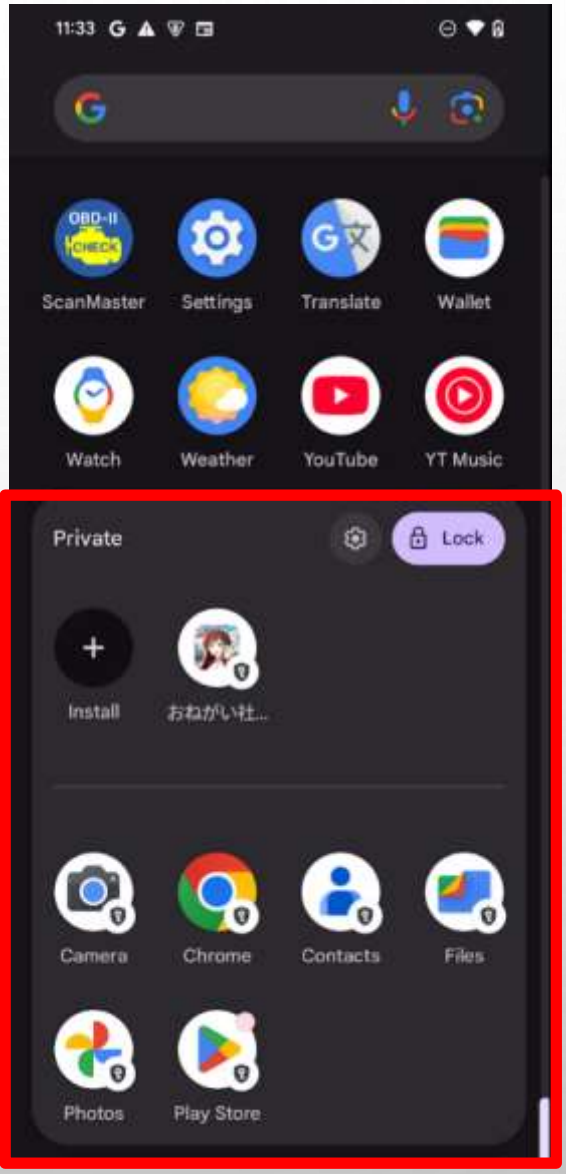
Lockをタップするとプライベートスペース全体が非表示になる。

ロックした場合、プライベートスペースにインストールしたアプリは以下の場所でも非表示になる。

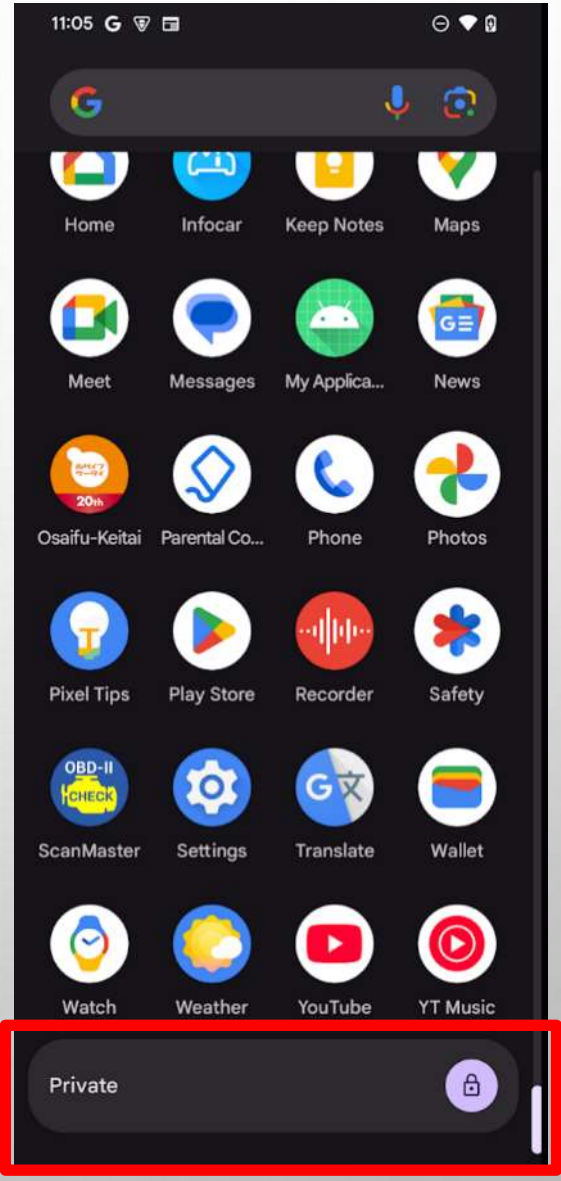
- 最近のアプリ一覧
- 通知
- 設定
- 他のアプリ

ロックの解除には認証が必要

表示状態



ロック状態



プライベートスペースに感じる闇

1. 隠蔽性：プライベートスペースの存在自体を隠せるため、他人に知られずに秘密のアプリや情報を保管できる。これは個人のプライバシー保護に役立つ反面、不適切な使用の可能性も示唆する。

2. セキュリティの二重構造：通常のロックとは別の認証を設定できるため、デバイスへのアクセスを許可しても、特定の情報へのアクセスを制限できる。便利だが、同時に複雑なセキュリティ構造を生み出す。

3. データの完全分離：プライベートスペース内のデータは完全に分離され、他のアプリからアクセスできない。セキュリティを高める反面、データの統合や管理を困難にする。

4. 企業管理との衝突：企業が管理するデバイスでは、管理者がプライベートスペースを無効化したり削除したりできる。これはユーザーのプライバシーと企業のセキュリティポリシーの間に緊張関係を生み出す。

5. 検出の可能性：完全に隠蔽することは難しく、特定のユーザーやアプリ、デバイスログなどでは検出される可能性がある。これは、完全なプライバシーの保証が難しいことを示す。

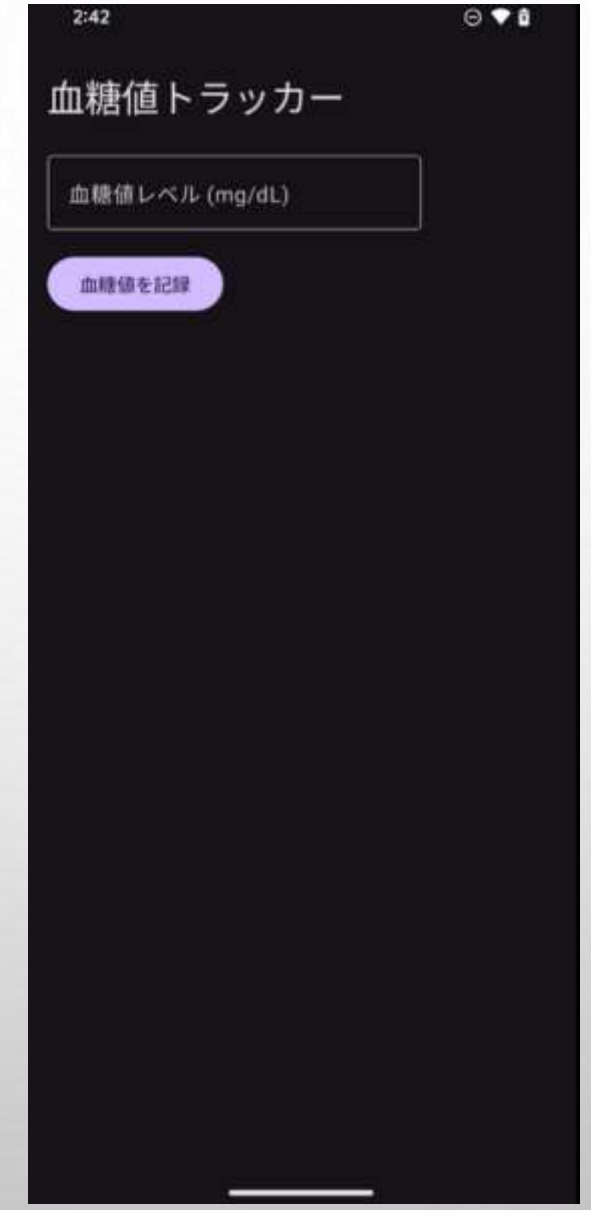
開発者が気を付けるべき点

1. アプリの分離と停止：プライベートスペースのアプリは、メインスペースとは別の独立した環境で動作し、スペースがロックされると完全に停止する。
2. 通知の制御：プライベートスペースがロックされている間、アプリからの通知は一切表示されないため、通知に依存するアプリは注意が必要。
3. データの分離：ユーザーコンテンツやアカウントはメインスペースと完全に分離され、データの共有や移行はできない。
4. 医療アプリへの影響：重要な通知や継続的な機能が必要な医療アプリは、プライベートスペースでの利用に適さない場合がある。ユーザーへの適切な情報提供が求められる。
5. セキュリティとプライバシー：プライベートスペースは追加の認証層を提供し、アプリやデータを保護する。開発者は、このセキュリティ機能を考慮したアプリ設計を行う必要がある。

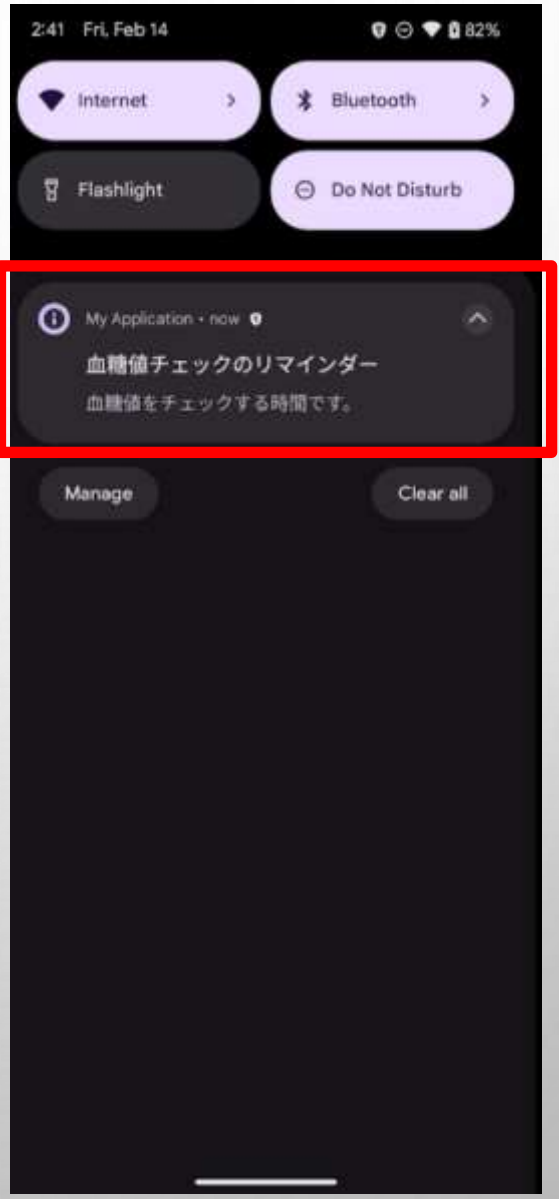
サンプルアプリは血糖値記録アプリ
血糖値を定期的にチェックするよう
にユーザーにリマインダー通知を送
信する。

まずは、アプリをメインスペースに
インストールした場合

アプリ起動



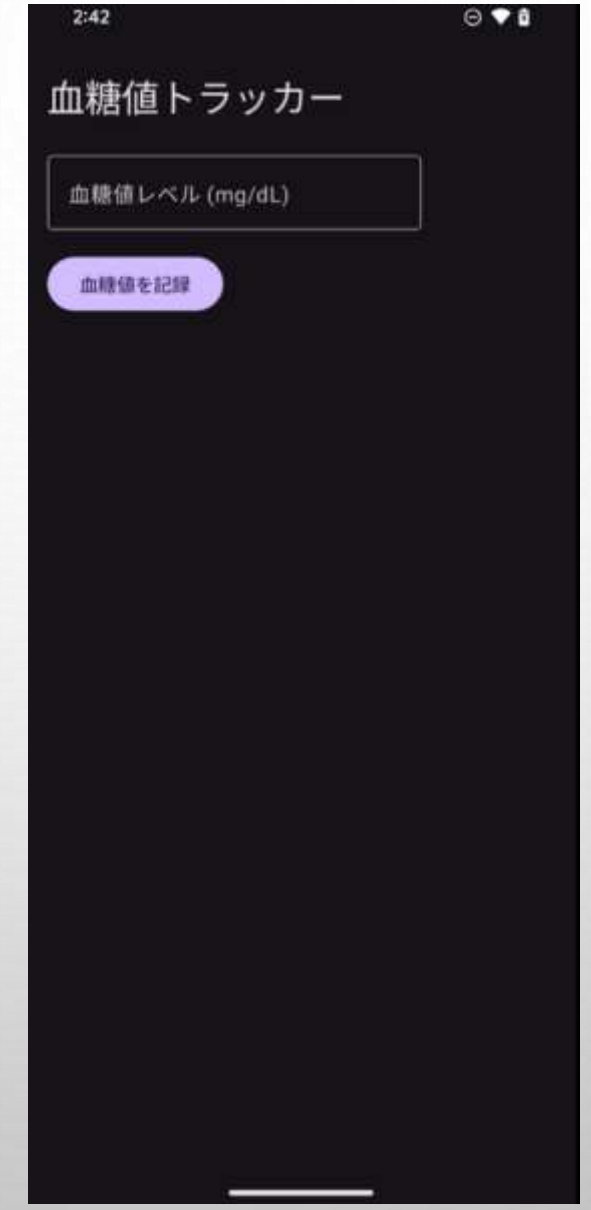
通知を表示



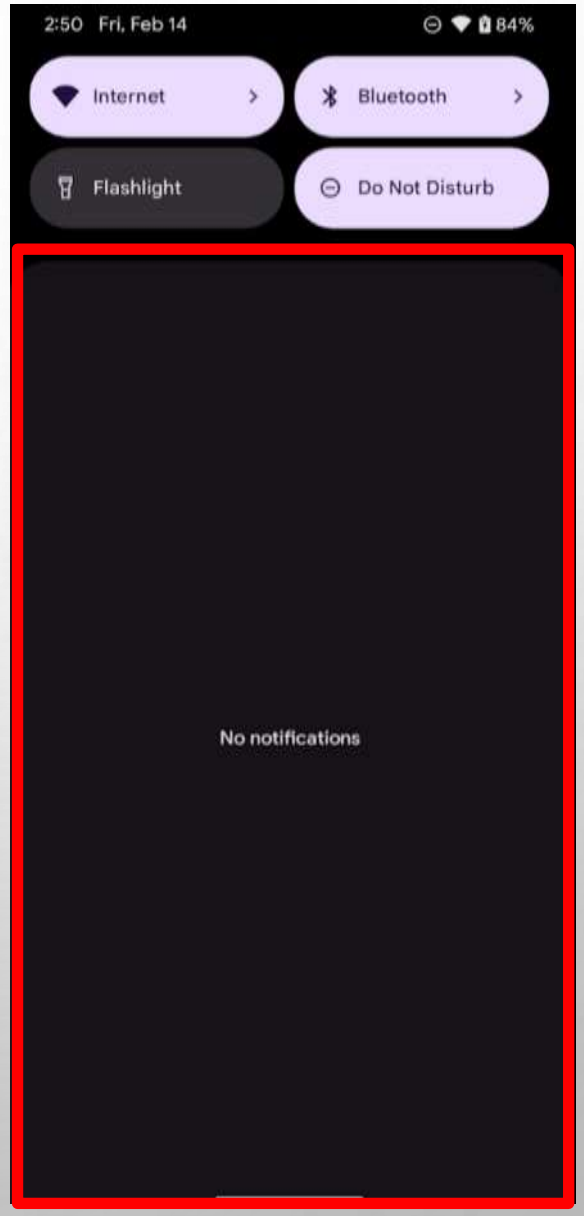
血糖値記録アプリをプライベートスペースにインストールし且つロックした場合

通知は一切表示されない
ロック前に表示されていた通知も表示されなくなる

アプリ起動



通知は表示されない

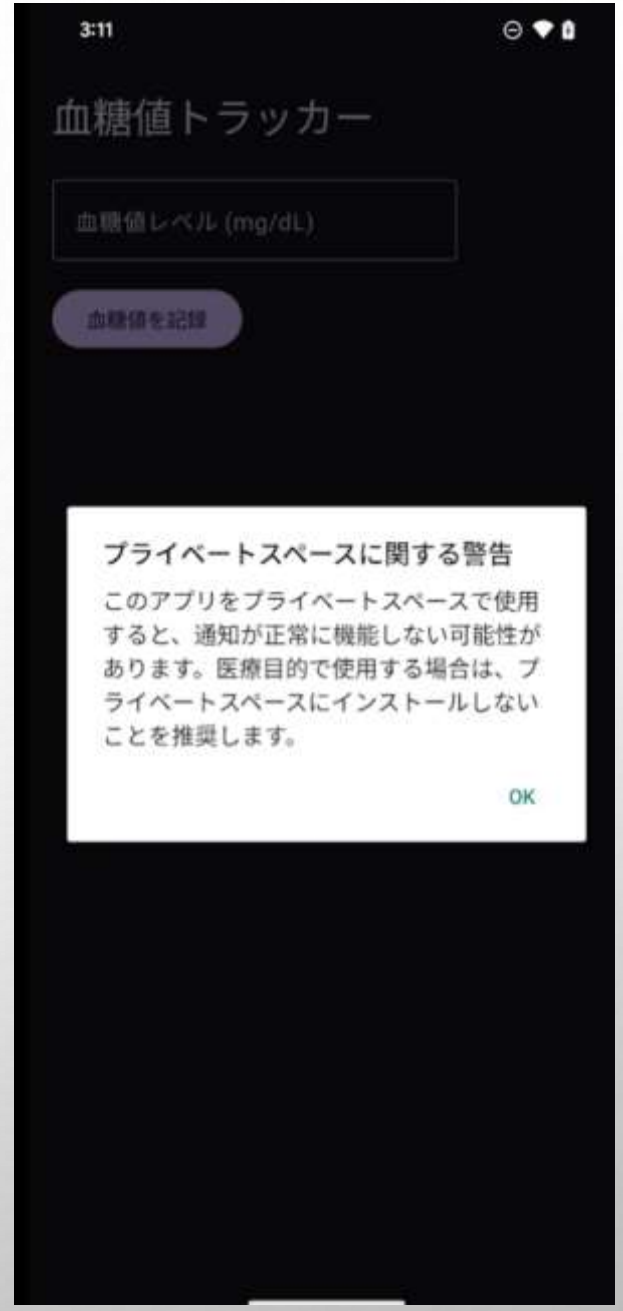


プライベートスペース非推奨の警告：

アプリの初回起動時、または設定画面に、プライベートスペースにインストールすると通知が正常に機能しない可能性がある旨を警告するメッセージを表示する。

医療用途で重要な通知が届かなくなるリスクを強調し、プライベートスペースの使用を非推奨とすることを明確に伝える。

可能であれば、プライベートスペースにインストールされているかどうかを検出し、その場合のみ警告を表示する。



その他

サンプルコードはJavaからKotlinへ

セキュアコーディングガイドのサンプルコードをJavaからKotlinに変更する主な理由は以下。

1. 安全性: KotlinはNull安全機能を強化。NullPointerExceptionのリスクを減らし、より安全なコードを実現。コンパイル時にエラーを検出しやすく、実行時エラーを抑制。

2. 簡潔さ: KotlinはJavaに比べてコード記述量が少ない。可読性が向上し、開発とメンテナンスの負担を軽減。

3. モダンな機能: Kotlinは関数型プログラミングや拡張関数など、現代的な言語機能をサポート。より効率的で柔軟なコーディングが可能。

4. Android公式サポート: GoogleがAndroid開発でKotlinを推奨。Android開発におけるKotlinの重要性が高まっている。

5. Java互換性: KotlinはJavaと互換性があり、既存のJavaコードやライブラリを移行または共存できる。段階的な導入が可能。

実際Kotlinは完結に書ける

Java

```
// Intentデータを取得
Uri uri = intent.getData();
String uriString = (uri != null) ? uri.toString() : "No URI";

// 検証用のUriRelativeFilterGroupを作成
UriRelativeFilterGroup filterGroup = new
UriRelativeFilterGroup(UriRelativeFilterGroup.ACTION_ALLOW);
filterGroup.addUriRelativeFilter(new UriRelativeFilter(UriRelativeFilter.PATH,
PatternMatcher.PATTERN_PREFIX, "/auth"));
filterGroup.addUriRelativeFilter(new UriRelativeFilter(UriRelativeFilter.QUERY,
PatternMatcher.PATTERN_LITERAL, "token=securetoken"));

// データがフィルタにマッチするか検証
boolean isMatch = false;
if (uri != null) {
    isMatch = filterGroup.matchData(uri);
}
```

Kotlin

```
// Intentデータを取得
val uri = intent?.data
val uriString = uri?.toString() ?: "No URI"

// 検証用のUriRelativeFilterGroupを作成
val filterGroup = UriRelativeFilterGroup(UriRelativeFilterGroup.ACTION_ALLOW).apply {
    addUriRelativeFilter(UriRelativeFilter(UriRelativeFilter.PATH,
PatternMatcher.PATTERN_PREFIX, "/auth"))
    addUriRelativeFilter(UriRelativeFilter(UriRelativeFilter.QUERY,
PatternMatcher.PATTERN_LITERAL, "token=securetoken"))
}

// データがフィルタにマッチするか検証
val isMatch = uri?.let { filterGroup.matchData(it) } ?: false
```

SDKバージョンは35へ

サンプルプロジェクトのSDK設定理由は以下。

1. 最新API活用: compileSdkVersionとtargetSdkVersionを35に設定し、最新のAndroid APIや機能を活用。セキュリティ機能やパフォーマンスを最適化。

2. 幅広いデバイスサポート: minSdkVersionを15に設定し、Android 4.0.3以降のデバイスをサポート。幅広いユーザー層をカバー。

3. 将来性: compileSdkVersionとtargetSdkVersionを35に設定し、将来のAndroidバージョンへの対応を見据える。長期的な保守性と互換性を確保。

4. 開発効率: 最新のbuildToolsVersion '35.0.0'を使用し、最新のビルドツールや最適化技術を活用。開発プロセスの効率化とパフォーマンス向上を期待。

5. セキュリティ: 新しいSDKバージョンを採用し、最新のセキュリティパッチや機能を取り入れる。HTTPS通信を扱うアプリにとって重要。

これにより、最新技術を活用しつつ、幅広いデバイスをサポートする柔軟なアプローチをとっている。

build.gradle の例

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 35
    buildToolsVersion '35.0.0'

    defaultConfig {
        applicationId "org.jssec.android.https.imagesearch"
        minSdkVersion 15
        targetSdkVersion 35
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-
rules.txt'
        }
    }
    namespace 'org.jssec.android.https.imagesearch'
}
```

今後

Android 7以前を対象とした記述やサンプルは削除します。

1. 最新のセキュリティ基準への対応: Android 7.0以降で導入された重要なセキュリティ機能や改善点を反映させるため。

2. 開発者の効率向上: 古い記述を削除することで、開発者が最新の推奨プラクティスに集中できる。

3. アプリの品質向上: 最新のAPIやセキュリティ機能を使用することで、より安全で高品質なアプリ開発を促進。

4. Google Playストアの要件遵守: Google Playストアが Android 7.0 未満をターゲットとするアプリのサポートを終了する方針に合わせるため。

5. ユーザー体験の向上: 最新のセキュリティ機能を活用したアプリを提供することで、エンドユーザーの安全性と満足度を高める。

これらの理由により、JSSECは最新のAndroidバージョンに焦点を当てたガイドラインを提供し、業界全体のセキュリティ水準の向上を図っている。

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
- 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16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	仲上 竜太	ニューリジェンセキュリティ株式会社
	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

（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。