

Google I/O 2018

What's new in Android Security

ABC Spring 2018
2018/6/9(土) 16:00~16:50
246講義室



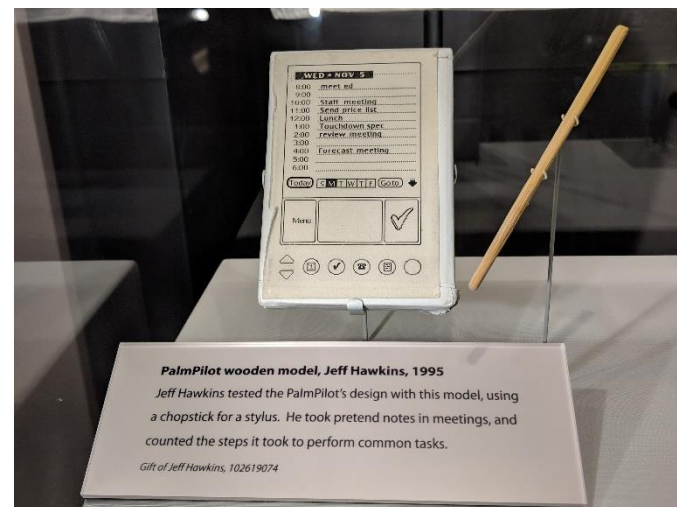
タオソフトウェア株式会社
代表取締役
谷口岳
@tao_gaku





本セッションについて

Google I/O 2018で「What new in Android Security」セッションがありました。どのような事が話されたのかを解説を踏まえながら説明致します。





自己紹介

タオソフトウェア株式会社

- 代表取締役 谷口岳
- 2005年創業、Android関連システム受託開発
- Google Playにアプリを多数公開
- ブログにて開発者向け情報を発信（昔）
- 雑誌執筆、講演
- セキュリティ色々





自己紹介

3DVRがスマホで 手軽に楽しめる!

話題の3Dバーチャルリアリティを体験しませんか?
TaoVisorにAndroidスマホをセットすれば
すぐに手軽に3DVRの世界が体験できます!



3DVR ゴーグル タオバイザー TaoVisor

3DVRは体験しないと、その面白さがわかりません。
様々な3Dアプリを多くの人に手軽に観てもらいたい、
作ってもらいたいとの想いから、スマートフォンをセットして、
3DVRを体験できるゴーグル「タオバイザー」を作りました。

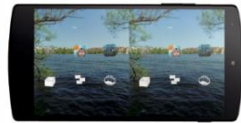
ご支援ありがとうございました!

タオバイザーはクラウドファンディング
を利用して制作されました。
583人の方から、目標金額250%以上の
1,374,500円のご支援を頂きました。

眼鏡をかけたままでも使える
子供も楽しい覗み立てキット
軽くて壊れにくい素材
持ち運びに便利
目の間隔調整ができる
Google Cardboard互換

YouTube等の
side by side形式の
3D動画も楽しめます

▲ 本製品の対象年齢は13歳以上です。対象年齢以下のお子様を使用の場合は、必ず保護者監督のもとで行ってください。



TaoVisor ホームアプリ

タオバイザーホームアプリは、タオバイザーと一緒に
使用する事で、より便利にタオバイザーを使用できる
ようにするアプリです。
Google CardboardやDive など他の3DVR用ゴーグル
でも使用できます。

■ サンプルコンテンツが楽しい!
■ アプリランチャーでカンタン!

Android アプリ Google play Google Playにて無料でダウンロードできます
動作環境: Android OS 4.1.2以上

タオバイザーについての詳しい情報はホームページをご覧ください。
<http://www.taovisor.com/>

Tao software タオソフトウェア株式会社
〒110-0015 東京都台東区東上野 2-1-1 フリーアクセスビル8F
TEL.03-6802-8247 FAX.03-6802-8347 <http://www.taosoftware.co.jp>

タオバイザー

3DVRゴーグル

- クラウドファンディングで資金調達
- <http://taovisor.com>





Android Security 安全なアプリケーションを作成するために

2012年1月1日発刊

開発者向け

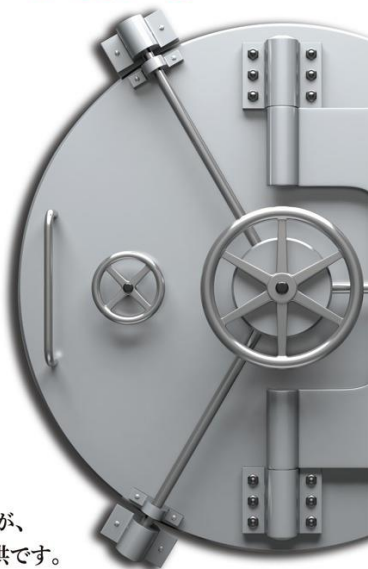
出版社：インプレスジャパン

Android Security

安全なアプリケーションを
作成するために

タオソフトウェア株式会社[著]

谷口 岳／井澤 正道／境原 永典／唐鎌 千里／北村 久雄
岡山 美幸／宮城 善雪／梶山 拓哉／島野 英司



新しいネットワーク市場の活性化を図る、
新しい枠組みの確立が求められています。
こうした取り組みの最も基本的な部分の一つが、
マーケットへの安全なアンドロイドアプリの提供です。
セキュリティに焦点を合わせた本書は、
アンドロイドのコミュニティに歓迎されることでしょう。
日本Androidの会 会長 丸山不二夫

インプレスジャパン



RiskFinder アンドロイドアプリの脆弱性発見ツール

APKファイルをアップロードするだけで脆弱性レポートが作成されます。

講演をする中で、
「気を付ける事が沢山あるのは分かった。
でも全てのプログラマが理解するのは
難しい何かいい方法はないか？」
という声があったので作ってみました。

1. プログラマでなくても使える
2. ソースコード不要
3. ウェブサービス型
4. 脆弱性以外も検出

The screenshot displays the RiskFinder web application interface. The top navigation bar includes 'Analyze', 'Results', and 'Help' links, along with a user profile 'ユーザ1'. The main content area is titled 'Summary' and shows the analysis results for 'VariousRisks1'. It features two large icons: a red octagon with an exclamation mark and the word 'ERROR' next to the number '28', and a yellow triangle with an exclamation mark and the word 'WARNING' next to the number '27'. Below these, there is an 'Analyze' section with a table of metadata:

Analyze	
RiskFinder Version	10
Analyzed Date	2013/04/24 21:46
Filename	VariousRisks1.apk
Size	1,130,608byte
SHA1	3124f29b5edbb4fe89f26823e20f4c8fc73762da
MD5	651444a864678885b61e4b60f1647fff

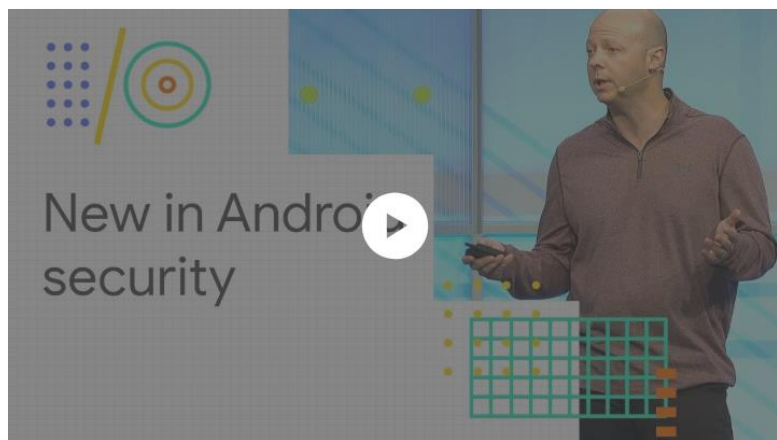
Below the metadata table is a 'Risk Summary' section with a table of detected risks:

No.	Level	Message
1	ERROR	デバッグモードのアプリケーション
2	ERROR	アプリケーション設定誤り (persistent=true)
3	ERROR	デバッグ経路書による署名

On the left side of the interface, there is a sidebar menu with categories like 'Information', 'Manifest Risk', 'Permission', 'File Access', 'String Resources', 'Logs', 'Network', 'Module', 'Cryptography', 'Secure Logic', 'Privacy Policy', 'Lint', 'Google Play', and 'Application Structure'. The 'Application Structure' category is currently selected.



What's new in Android Security Session



[Session] What's new in Android security

Thu. May 10, 9:30AM - 10:30AM

Stage 1

Beginner

Attend this session to learn about security features in Android and how they affect your apps. It will cover new APIs and best practices for protecting the integrity of your app and the privacy of your data.

Android & Play



Google I/O 2018

5月10日(木) 9:30~



Youtube:

https://www.youtube.com/watch?time_continue=12&v=r54roADX2MI

Androidのセキュリティ戦略



Android Security Strategy

Android Security Strategy

Google Play Protect

Defend against Internet-borne threats

User experience that offers security CCC (comprehension, control, confidence)

Platform Engineering

Feature dev

OS hardening, leverage HW

SDLC

Vulnerability management

Full cycle, e.g. fuzzing and security reviews, for AOSP and partners





Androidの3つのセキュリティ戦略

1. Google Play Protect

1. マルウェアからの保護
2. モバイルセキュリティサービス

2. Platform Engineering

1. OSのセキュリティ強化、ハードウェア活用
2. SELinux,セキュアブート、暗号化...

3. SDLC (Software Development Life Cycle)

1. セキュリティテスト
2. セキュリティパッチ（毎月）



セキュリティパッチOEM義務付け

毎月セキュリティパッチを出しているが、OEM端末等は総てが適用されるわけではない



Androidの定例セキュリティパッチをOEMに義務付け
今後は多くの端末でセキュリティパッチが適用される
事となる

Project Treble

- デバイスベンダーがOSを効率的にアップデートできるようにする仕組み（Android Oから）
- <https://source.android.com/devices/architecture/treble>



Androidは安全である





**Androidのセキュリティは
他のモバイルプラットフォーム
と同じレベルである。**

by Google



3つの証明？

1 Transparency（透明性）

2 Measurability（測定可能性）

- 1. PHAをインストールした発生率を見る
- 2. エクスプロイトコード価格を見る



1. Transparency





AndroidOSが安全である理由1

Transparency（透明性）＝オープン

Androidプラットフォーム

- オープンプラットフォームはセキュリティに貢献する
- 防御側は、数千人のGoole社員＋アーム＋インテル＋クアルコム＋ブロードコム＋Linuxコミュニティ＋学術研究コミュニティ

クローズドプラットフォーム

- 防御側は1社の従業員

OpenSSL
(´・ω・`)



2. PHAインストール率





AndroidOSが安全である理由2

Installs of potentially harmful apps



Source: Android Security Year in Review 2017, Google

PHA (Potentially Harmful Applications) :
マルウェアまたは潜在的に有害なアプリケーション

緑 : Google Play以外からのPHAインストール

青 : Google PlayからのPHAインストール



Google Play Protect

端末やアプリに不正な動作がないかを定期的に確認するアプリ。

- https://www.android.com/intl/ja_jp/play-protect/
- Google I/O 2017発表
(2017/7/22頃提供)
- 設定→Google→セキュリティ
→Google Play Protect→ON





PHAアプリのインストール率

GooglePlayからPHAアプリがインストールされた割合

- 0.2から0.1に減っている 2 倍減っていることになる。
(グラフをもっと減っているように見えるように加工すればいいのに)

Google Play以外からのPHAアプリインストール率

- Google Protectがクライアントで動作するようになったので明らかに減少している

機械学習を使ってマルウェアの60%が検知されている

PHAをインストールする確率は雷にあたるのと同じ確率(°D°)

Tip:Android Developer Blog: 2018/5/24

- Keeping 2 billion Android devices safe with machine learning
- 機械学習してるとか言ってるだけで特に有益な情報が出ているわけではない
- <https://android-developers.googleblog.com/2018/05/keeping-2-billion-android-devices-safe.html>



3. エクスプロイト コード価格





AndroidOSが安全である理由3

Exploit（エクスプロイト）コード

- 脆弱性検証するための実証コード
- 脆弱性を利用した悪意ある行為のために書かれたコード

エクスプロイトコード価格

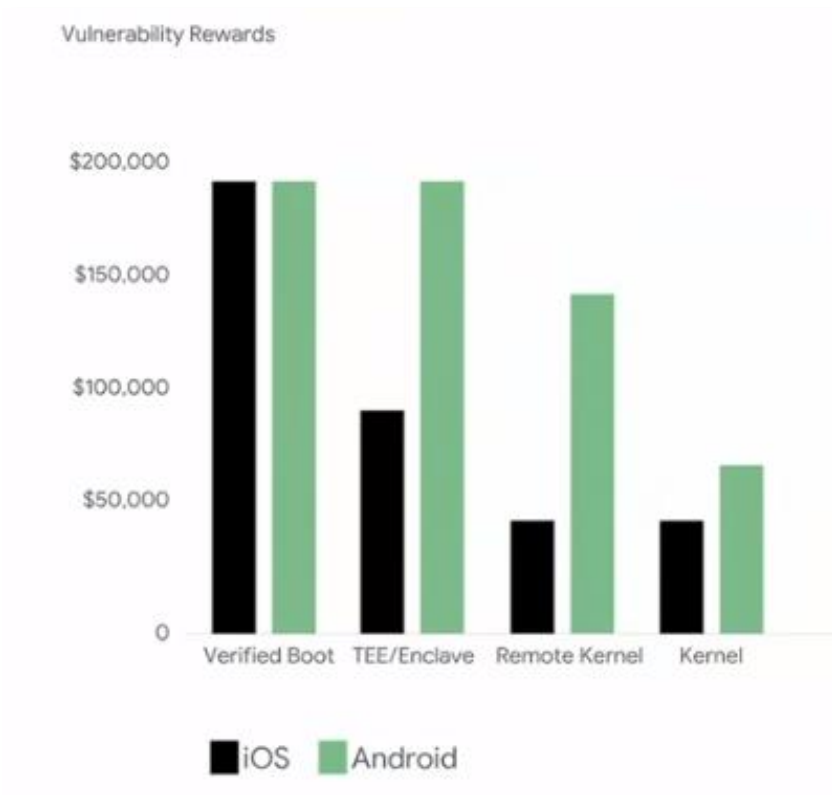
- 売買されている。
- 攻撃性が高いものほど高価
- 攻撃が困難な物ほど高価

※Androidのエクスプロイトコードは高価→安全であるとGoogleは言っている。

- 言いたいことはわかるけど、ちょっと強引かなという気もする。



企業のバグ発見報酬プログラム



緑 : Android

黒 : iOS

Androidの方が高額を支払っている

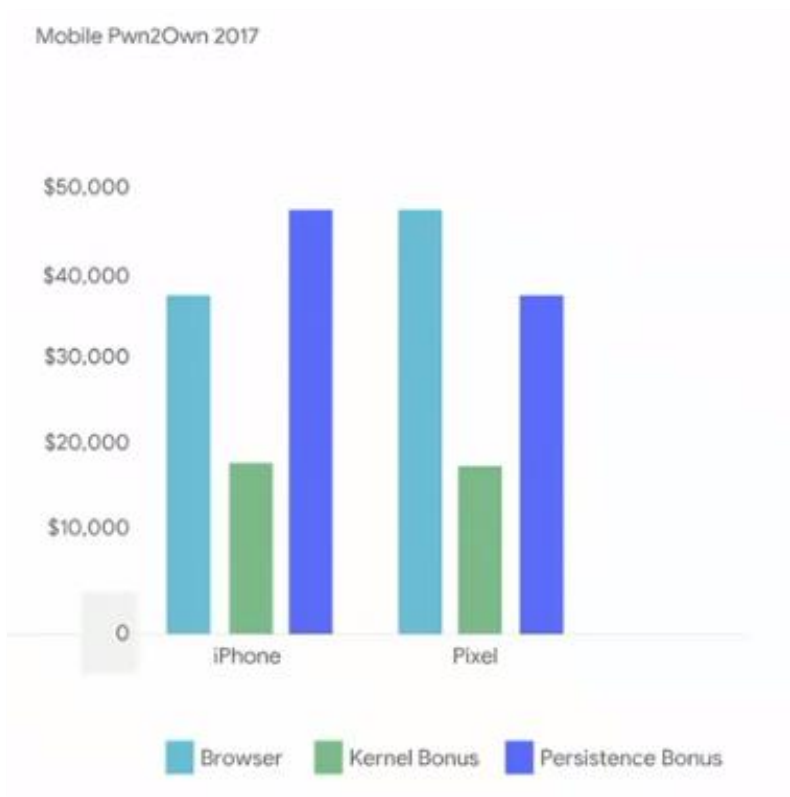
- Verified Boot
- TEE/Enclave
- Remote Kernel
- Kernel

Android Security Rewards

- セキュリティバグを報告すると賞金が貰える
- <https://www.google.com/about/appsecurity/android-rewards/>



ハッキングコンテスト



**Mobile Pwn2Own
2017(賞金付き脆弱性発見コ
ンテスト)**

AndroidとiPhoneはほぼ同額



アンダーグラウンドマーケット

アンダーグラウンドマーケットでの価格も、他のプラットフォームと同じかそれ以上の価格が観測されている。



よって安全である！

1. Transparency
2. PHAをインストールした発生率
3. エクスプロイトコード価格



Androidのセキュリティは他のモバイルプラットフォームと同じレベルにある。by Google

Android P





Android Protected Confirmation





Protected Confirmation

Why stronger confirmation?



Medical

Confirm medical device operations (e.g. inject insulin)



Financial

Confirm large money transfers



Enterprise

Confirm signing in using phone as second factor

スマートフォンのセキュリティは良くなっているが

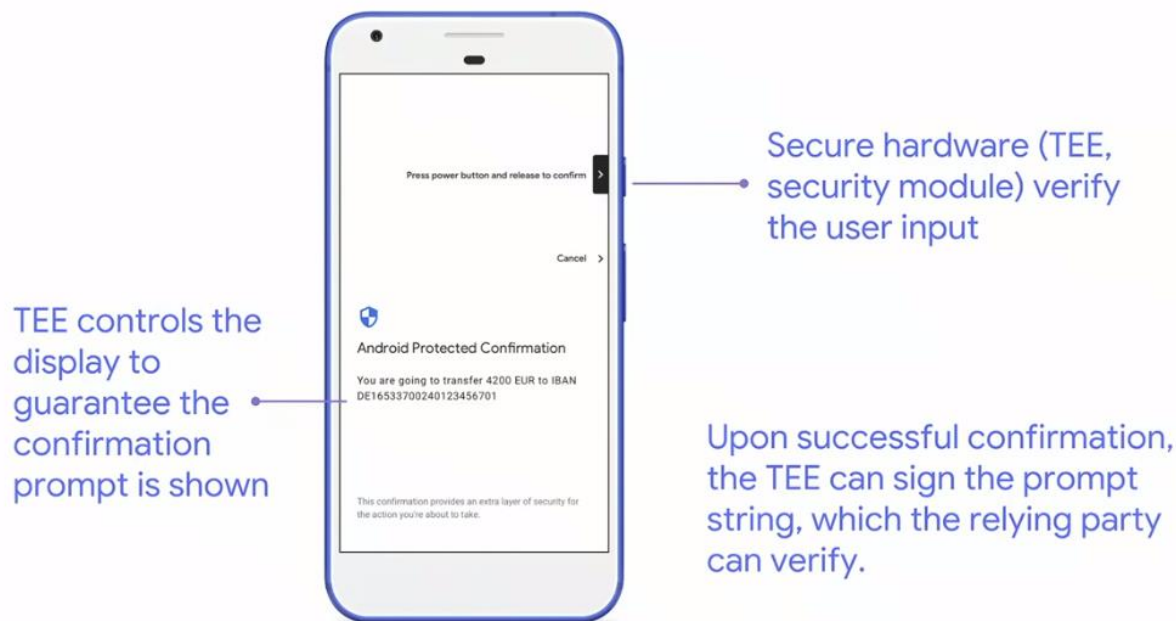
- スマートフォンから投票はしない
- インシュリンのような生命に重要な医療機器をプログラムしない
- パスポート機能もない

医療、金融、企業なので安心して使えるようにさらなるセキュリティ機能として

Protected Confirmation機能を作成した



Confirmation Prompt動作



Android P には、Trusted Execution Environment (TEE) を使用して任意のプロンプト文字列がユーザーに確実に表示および確認されるようにする Android Protected Confirmation が導入されています。TEE では、ユーザーが正常に確認した場合にのみ、アプリで検証できるプロンプト文字列が署名されます。



その他

- Android Protection ConfirmationがサポートされているAndroidP端末が必要
- ソフトウェアではなくハードウェアボタンによる確認
- 主要OSでは初めてAPIを実現
- ConfirmationDialogはConfirmationPromptに名前変更(Dp1→Dp2)

https://developer.android.com/sdk/api_diff/p-dp2-incr/changes/pkg_android.security#ConfirmationDialog



Confirmation Prompt

```
// Create and use a key that requires confirmation

keyPairGenerator.initialize(KeyGenParameterSpec.Builder(
    alias,
    KeyProperties.PURPOSE_SIGN or KeyProperties.PURPOSE_VERIFY)
    .setDigests(KeyProperties.DIGEST_SHA512)
    .setUserConfirmationRequired(true)
    .build())
val keyPair = keyPairGenerator.generateKeyPair()

val dialog = ConfirmationDialog.Builder()
    .setPromptText("Send $10,000 to Greg?")
    .setExtraData(myExtraData)
    .build(context)
dialog.presentPrompt(threadReceivingCallback, callback)
```

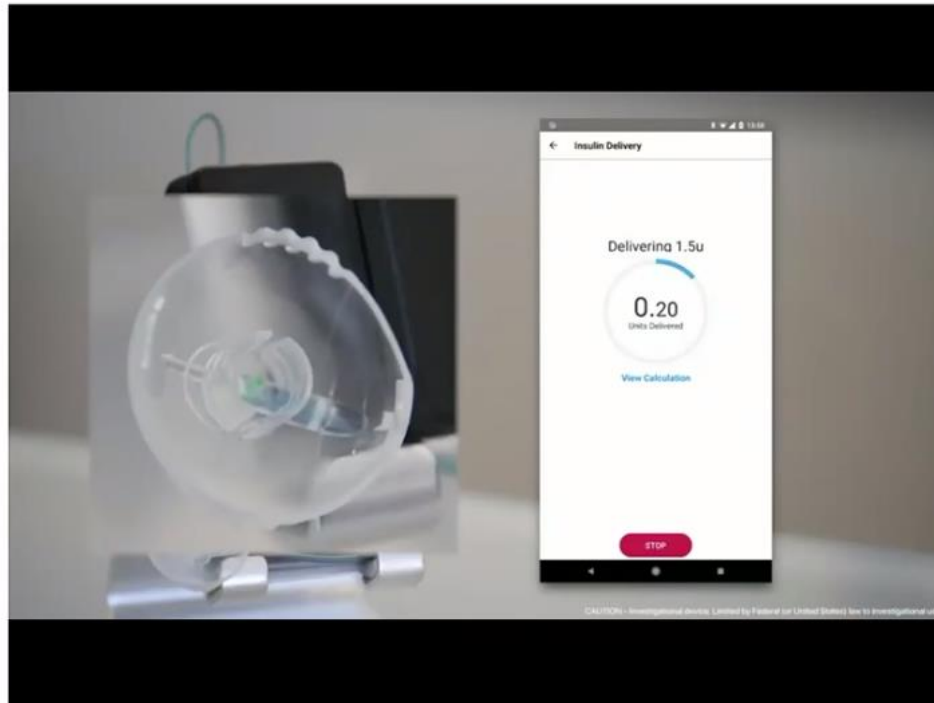
Require user confirmation
before key can be used

Show confirmation
prompt



適用例: Bigfoot biomedical

Bigfoot Biomedical

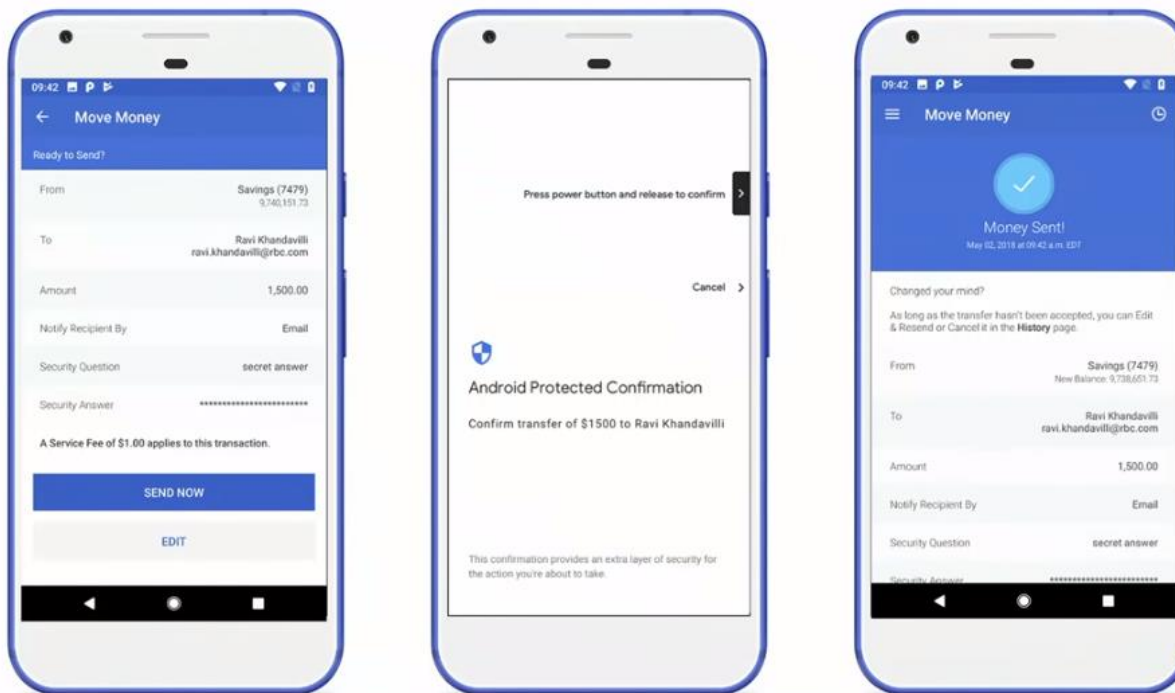


インシュリンの投与（糖尿病）



適用例:RBC

RBC

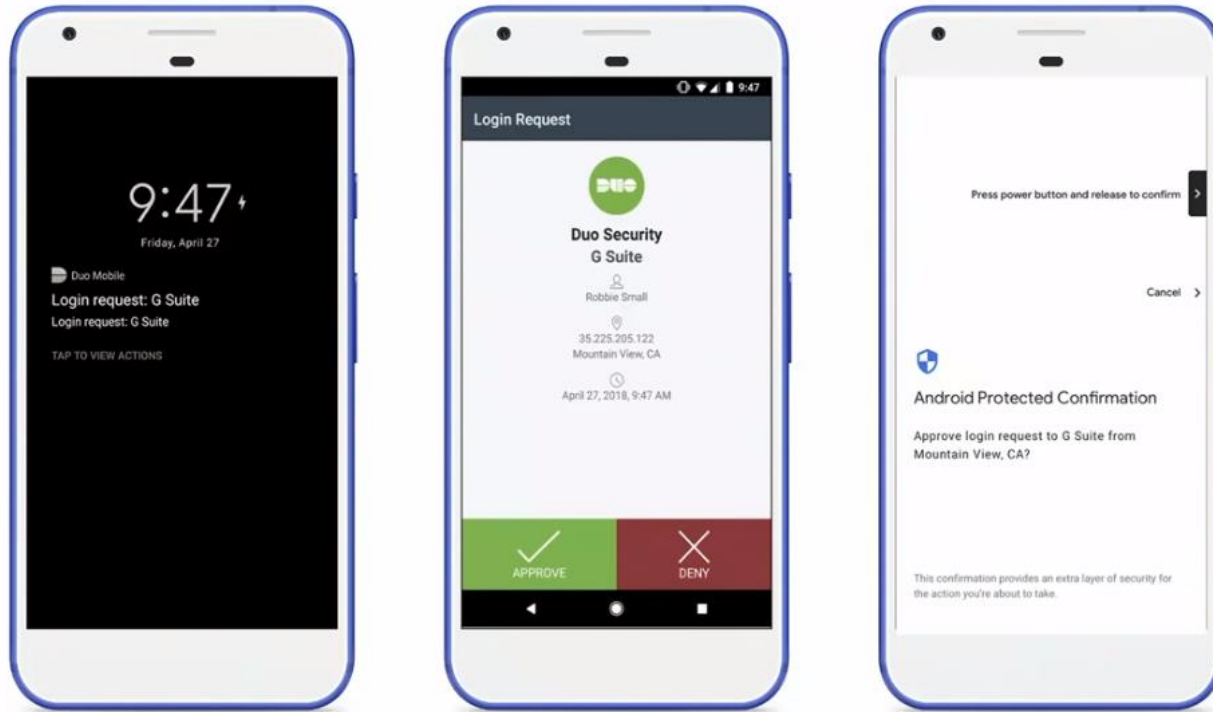


お金を1500ドル転送



適用例: Duo Security

Duo Security



<https://duo.com/>



Launch Partners

Requires Supported
Android P Device



StrongBox





About Android KeyStore

Android KeyStoreとは、暗号で使用する鍵を安全に保存する場所及び機能
鍵は暗号化された上でKeyStoreに格納されるため、Root端末であっても鍵を取り出す事はできない。

Android4.3

- 導入されたが遅いかつ、鍵の生存期間が不明瞭だったため利用に耐えれなかった

Android6.0 リニューアル

- OSに指紋認証機能追加
- ユーザのロック画面知識ファクタ（Lock Screen Knowledge Factor :LSKF）とAndroid KeyStoreの関連機能追加
- AES（共通鍵暗号）が利用可能
- 一般的な利用に耐えられるようになった。



AndroidKeyStoreタイプ

Comparison of KeyStore types

NEW!



StrongBox: OSはP以上で対応端末のみ利用可能



3つのタイプのキーストア

1. Android System

- OSがサポート（ソフトウェア）

2. TEE(Trusted Execution Environment)

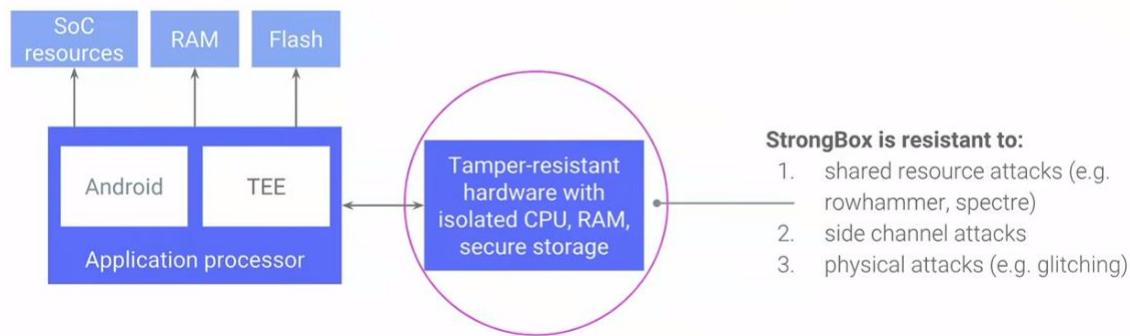
- N以上で全ての端末がサポート

3. Strong Box (New!)

- P以上の端末でサポート（必須ではない）
- 改ざんに強いハードウェアで実現
- `hasSystemFeature(FEATURE_STRONGBOX_KEYSTORE)`



StrongBox: additional KeyStore type



<https://developer.android.com/preview/features/security#hardware-security-module>

Android P がインストールされたサポート対象の端末では、ハードウェアのセキュリティ モジュール内にある Keymaster HAL の実装である *StrongBox Keymaster* を使用できます。このモジュールには、独自の CPU、安全なストレージ、真性乱数ジェネレータのほか、パッケージの改ざんやアプリの不正なサイドローディングを防ぐ追加のメカニズムが含まれています。システムでは、StrongBox Keymaster に格納されている鍵をチェックするときに、信頼できる実行環境（TEE）で鍵の整合性を保証します。



StrongBoxの利用するソースコード

```
// Set StrongBox when generating the key

val kpg = KeyPairGenerator.getInstance(
    KeyProperties.KEY_ALGORITHM_EC, "AndroidKeyStore")

kpg.initialize(KeyGenParameterSpec.Builder(
    alias,
    KeyProperties.PURPOSE_SIGN or KeyProperties.PURPOSE_VERIFY)
    .setDigests(KeyProperties.DIGEST_SHA512)
    .setIsStrongBoxBacked(true)
    .build())

val kp = kpg.generateKeyPair()
```

Use StrongBox

setIsStrongBoxBacked(true)を設定

端末がStrongBox非対応の場合はStrongBoxUnavailableExceptionとなる



Keyguard-bound keys





Keyguard-bound keys

Keyguard-bound keys

All Android P Devices

Require unlocked device



- **encrypt** at any time
- **decrypt** only when device is **unlocked**

Protect private data:



- enterprise
- health
- passwords
- auth tokens
- IM msgs
- ...

端末がロックされていない状態では復号化できなくする。

ロック画面のライフサイクルに関連付けられる

- 暗号化はいつでもできる
- 復号化はアンロックした時のみ可能
- 端末を紛失したり盗まれた時に有効



Keyguard-bound keys ソース

```
// Using keyguard-bound keys
```

```
val kg = KeyGenerator.getInstance(KEY_ALGORITHM_AES, ...)
kg.init(
    KeyGenParameterSpec.Builder("k", PURPOSE_ENCRYPT or PURPOSE_DECRYPT)
        .setBlockModes(BLOCK_MODE_GCM)
        .setEncryptionPaddings(ENCRYPTION_PADDING_NONE)
        .setUnlockedDeviceRequired(true)
        .build())
val key = kg.generateKey()
...
cipher.init(Cipher.ENCRYPT_MODE, key)
...
cipher.init(Cipher.DECRYPT_MODE, key)
```

The key can only be used to decrypt
when the device is unlocked

Initialize a cipher to
encrypt sensitive data

Initialize a cipher to decrypt
sensitive data when unlocked

setUnlockedDeviceRequired (true)を設定

Biometric Prompt

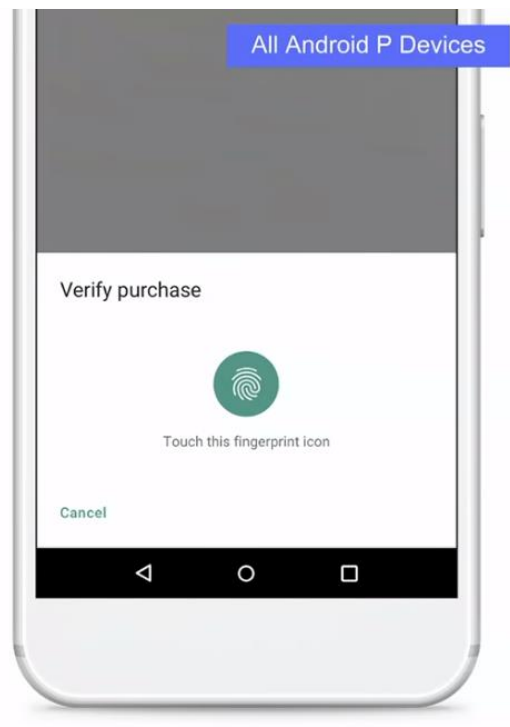




Biometric Prompt

BiometricPrompt

- Replacement for FingerprintManager
- Chooses the most appropriate biometric
 - fingerprint, face, iris
- System-wide consistent experience
- Support library version coming



FingerprintManagerに代わる物

- 指紋認証だけでなく、顔認証、虹彩認証、他利用
- `USE_FINGERPRINT` パミッション非推奨、`USE_BIOMETRIC` パミッション新規追加
- 指紋認証UIがサードパーティアプリによってバラバラで指紋認証処理に一貫性がなくユーザの利便性が悪かったので、標準UIを用意
- PとP以下と一つのコードで記載できるように、サポートライブラリが用意される予定



FingerPrint系クラス

FingerPrintDialogクラス

- Android P DP 1 で追加
https://developer.android.com/sdk/api_diff/p-dp1/changes/classes_index_additions
- Android P DP2で削除
https://developer.android.com/sdk/api_diff/p-dp2-incr/changes/classes_index_removals

FingerPrintManagerクラス

- API Level23 (Android 6.0:M)で追加
- API Level28 (Android P)で非推奨

BiometricPrompt

- 生体認証機能が存在するかの確認
- `hasSystemFeature( FEATURE_FINGERPRINT )`
- `hasSystemFeature( FEATURE_IRIS )`
- `hasSystemFeature( FEATURE_FACE )`



BiometricPrompt ソースコード

```
// Using BiometricPrompt

val prompt = BiometricPrompt.Builder(this)
    .setTitle("Verify purchase")
    .setSubtitle("...")
    .setDescription("...")
    .setNegativeButton("Cancel", executor, listener)
    .build()

prompt.authenticate(
    cryptoObject, cancellationSignal, executor, callback);
```

*Use a crypto object so user auth
properly gates data access*



USE_BIOMETRICパミッション追加

USE_BIOMETRIC

- 端末が持っている総ての生体認証に対応したパミッション
- API Level28(AndroidP)で追加
- protection Level Normal

USE_FINGERPRINT

- 非推奨
- API Level23で追加、28で非推奨
- Protection Level Normal



WebAuthn and FIDO2





About Web Authn(Web Authentication)

All Android O+ Devices
Expected: Q4 2018

WebAuthn and FIDO2 (coming Q4)

Biometric
re-auth via
Javascript web
APIs



- 現在Webサービス上で認証にパスワードを使用しているが、フィッシング詐欺、パスワードの使いまわし問題、脆弱性によるパスワード流出問題がある。
- FIDOアライアンスとW3CがWEB認証の新しいプロトコルとなる、「WebAuthn」を作成（2018/4/10:現在勧告候補）
- ウェブサイトにログインする時、ロックスクリーンや生体認証を利用してウェブサイトへの認証を行う。
- 対応にはWebサイトとブラウザが対応している必要がある。ブラウザは、Google (Q4), Mozilla, Microsoftサポートを表明



パスワードなしプロトコルイメージ

ユーザ登録時

- クライアントはサーバ毎に公開鍵と秘密鍵を作成する
- 公開鍵をサーバーに送り、サーバーは公開鍵を保存する

ログイン時

- サーバはクライアントに、適当なメッセージを送る
- クライアントはメッセージを秘密鍵で暗号化しサーバに送る
- サーバは公開鍵で復号化して、最初にしたメッセージと同じならログインとする。



TLS by Default



TLS by Default

TLS by default: integrity of data-in-transit



<https://developer.android.com/training/articles/security-config.html>

- TargetSDKがP以上の時Network Security Configuration ファイルで指定するuseCleartextTrafficのデフォルトがtrueからfalseに変更
- 平文通信を禁止するので、通信をしているライブラリを利用している場合は対応済か確認が必要



TLS by default: integrity of data-in-transit



<https://developer.android.com/training/articles/security-config.html>

- **FIPS (フィップス) : Federal Information Processing Standard)**アメリカ政府が取り扱う情報機器や情報技術に求められる基準
- **Boring SSL**がNISTからCAVP署名を受け取ったのでFIPS 準拠となった。



KeyAttestation

All Android Oreo+
Devices

Key Attestation: to check device integrity

KeyStore key

attestation gives you a signed statement from the secure hardware attesting to device and key properties

- Passed Verified Boot
- Has recent security patch
- Key is protected by TEE or **StrongBox**
- Key requires user authentication within X seconds
- **Firmware hash** (compatible P+ devices)
- ...

Sample code: <https://github.com/googlesamples/android-key-attestation>

- デバイスが改竄されていないかを判断
- Root化の判断に使用可能

バックグラウンド動作

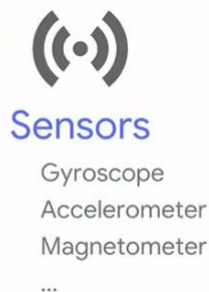




バックグラウンド盗聴禁止

All Android P Devices

Sensors access only in foreground



- AndroidP端末上では、APIレベルに関係なく、アプリがバックグラウンドでアイドル状態の時に、カメラやマイク、フォン、センサー（加速度センサー、ジャイロセンサー他）からデータを取得できなくなる。
- targetSDK等関係しない。



フォアグラウンドサービス ソース

```
// Foreground service

val notifIntent = Intent(this, ExampleActivity::java.class)
val pendingIntent = PendingIntent.getActivity(this, 0, notifIntent, 0)
val notification =
    Notification.Builder(this, CHANNEL_DEFAULT_IMPORTANCE)
        .setContentTitle("...")
        .setContentText("...")
        .setSmallIcon(icon)
        .setContentIntent(pendingIntent)
        .setTicker("...")
        .build()
startForeground(ONGOING_NOTIFICATION_ID, notification)
```

Start a foreground service with a persistent notification to use sensors

アクセスする場合は、フォアグラウンドサービスにする。

- setForgtound()永続的なNotificationが表示される。
- AndroidPでFORGROUND_SERVICEパMISSIONが必要（プロテクションレベル Normal)

一時的（1分）有効なホワイトリストへのアプリ登録が可能

- AlarmManager.setAndAllowWhiteIdele()
- AlarmManager.setExactAndAllowWhiteIdle()



DNS over
TLS

DNS over TLS





セキュアなDNS

DNSとは、ドメイン名からIPアドレスを調べる仕組み

DNSは平文通信を行うため（UDP）、中間者攻撃が可能であり盗聴や改ざんが可能。

対策

- DNS over TLS
- DNS over DTLS
- DNS over HTTP
- DNSSEC(改竄のみ対応)



RFC7858, RFC8310

Google Security Blog

- DNS over TLS support in Android P Developer Preview(April 17, 2018)
- <https://developers-jp.googleblog.com/2018/05/dns-over-tls-support-in-android-p.html> (日本語)



DNS over TLS動作

「ネットワークとインターネット」 → 「詳細設定」 → 「プライベート DNS」
(B1→B2で変更)

デフォルトON (Automatic)

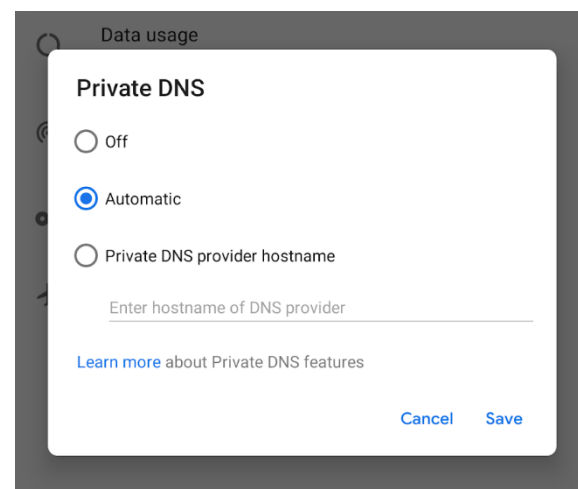
通信するDNSサーバーが、DNS over TLSをサポートしているならDNS over TLSで通信、それ以外なら今でどおり通信する。

Private DNS provider hostname

- 常にDNS over TLSを使いたい時に、対応サーバーを指定する使い方か？

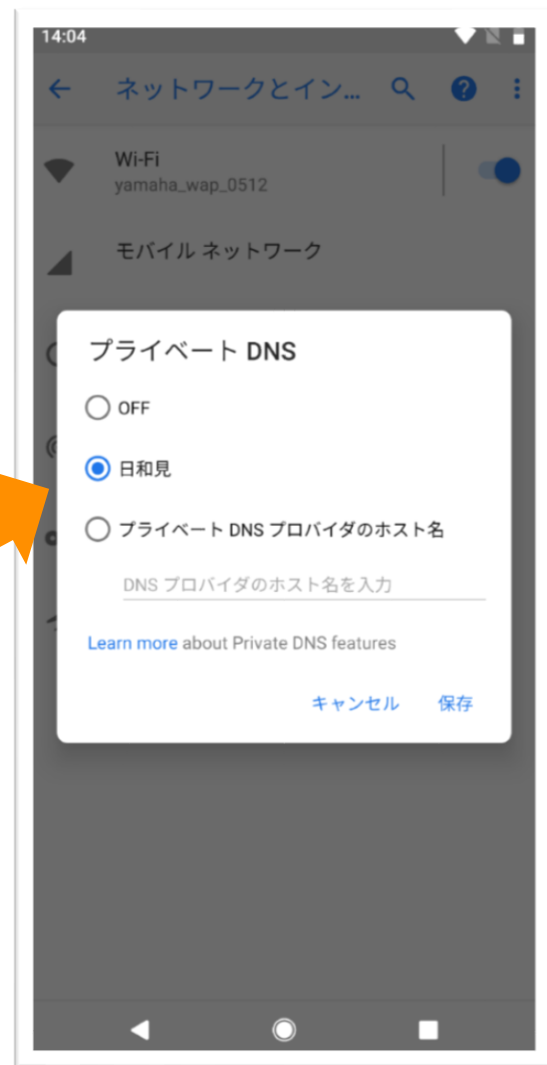
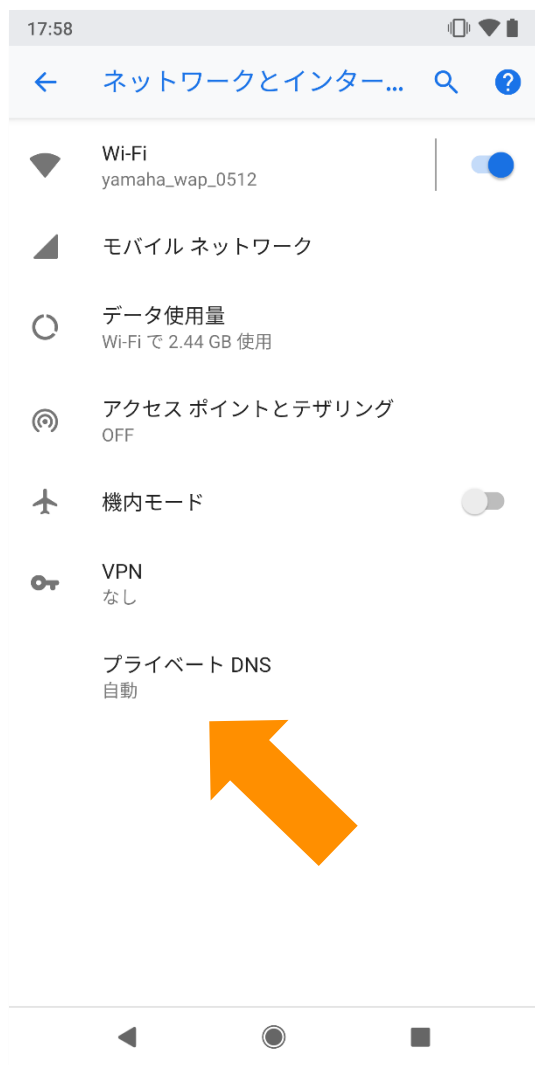
追加API

- `LinkProperties.isPrivateDnsActive()`
- `LinkProperties. getPrivateDnsServerName()`





Android P Beta 日本語表記



β2で「自動」に治ってました



Google Public DNS(8.8.8.8)

- DNS-over-HTTPS API
- DNS over TLSとは違う
- <https://developers.google.com/speed/public-dns/docs/dns-over-https>



Lock down
mode

Lock Down mode





Lock Down Mode

一時的にデバイスにアクセスできなくするモード

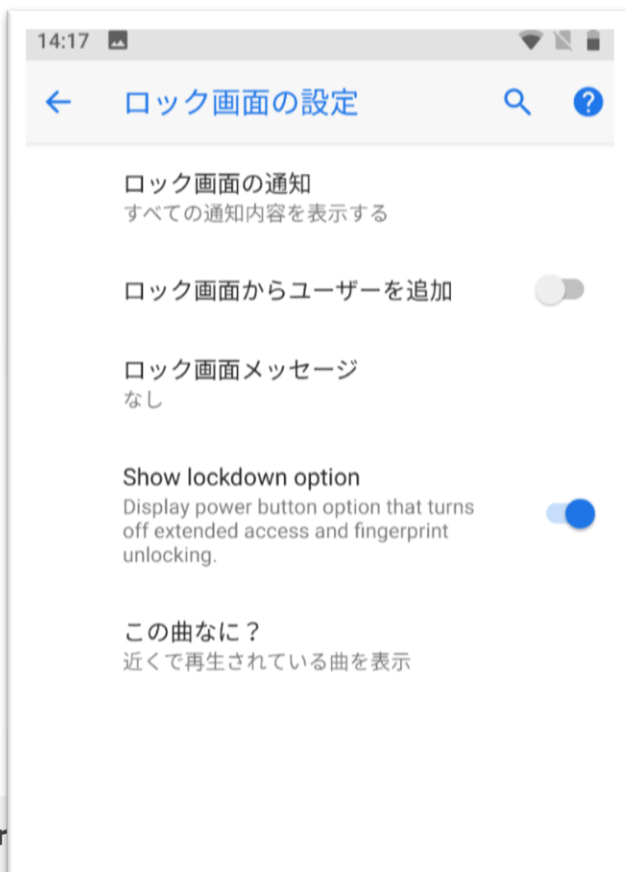
セキュリティゲートで端末を人に渡したりする時、警察に端末を押収された、盗難等レアケースではあるが、生体認証、他によって端末にアクセス可能にできるケースがある。（就寝時にも有効かもしれない）

- パスワード,PIN,パターン以外使用できなくなる。
- Smart Lock(身に着けている物、場所、顔認識、音声認識等でロックを外す機能)が無効
- 指紋認証（生体認証）が無効



Lock Down Mode動作

1. 「設定」 → 「セキュリティと現在地情報」 → 「ロック画面の設定」 → 「ロックダウンオプションの表示」
2. 電源長押しでメニューを出して、ロックダウンを入力でロックダウンモード





AndroidOS危険危険と言うほどではなくなってきました。

AndroidPのセキュリティ変更点については、まだまだ色々ありますが、大体細かい所です。

タオソフトウェア株式会社 谷口岳

ありがとうございました。

