


# Trust Your Android

## Authentication in Marshmallow

Panayiotis “Pete” Varvarezis  
Capital One  
Android Platform Architect



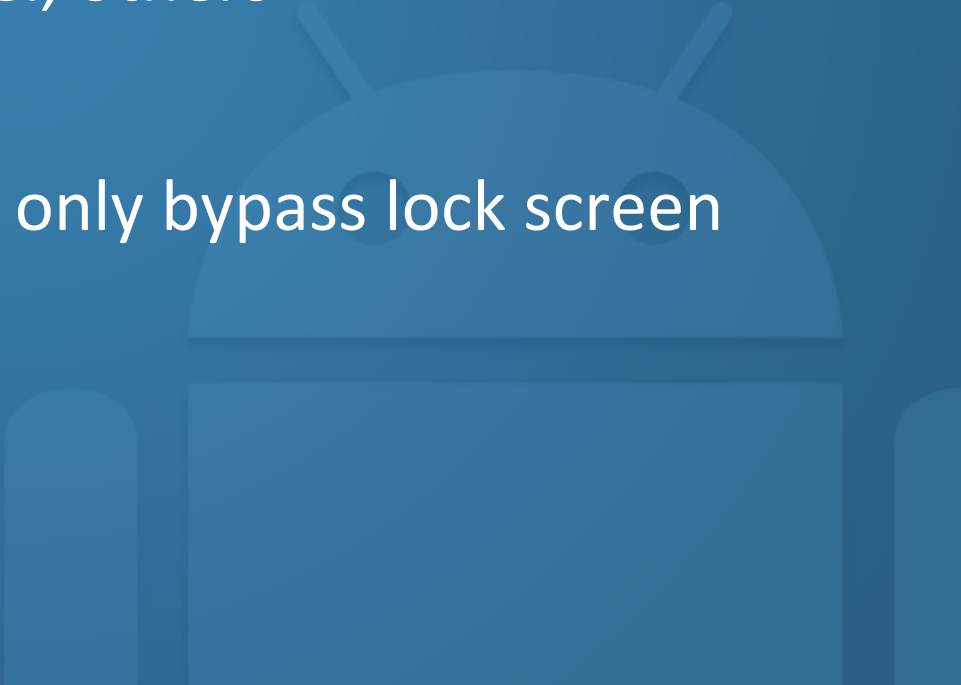
# Overview

- What's New in Authentication
  - How It Works
  - Fingerprint / Confirm Credentials
  - Demo
  - Recap / Wrap-up
- 

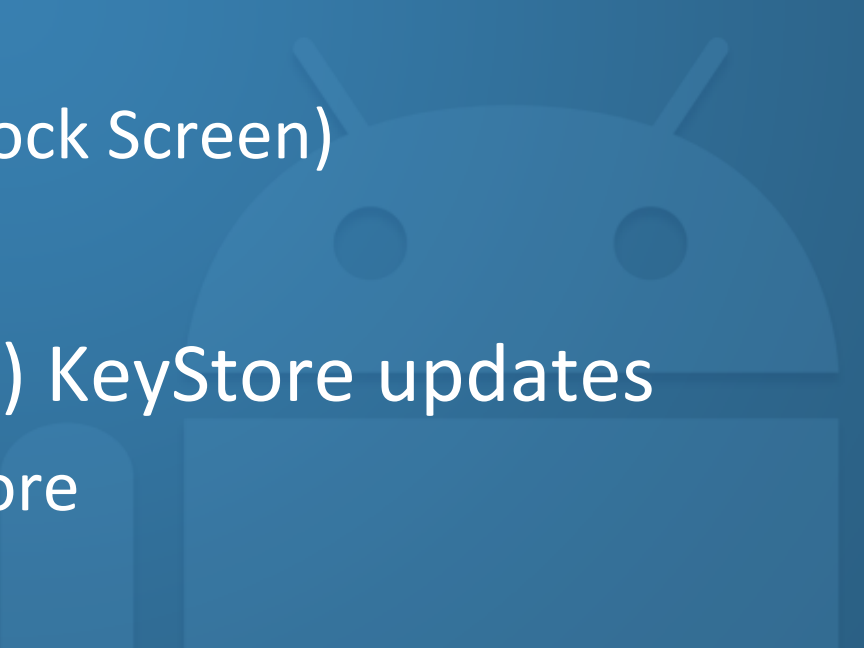
But First  
A History Lesson



# Android + Fingerprint

- History
    - 2011: Motorola Atrix
    - 2013: HTC One Max
    - 2014: Samsung, Huawei, others
  - Issues
    - Early implementations only bypass lock screen
    - No/Proprietary SDK
    - Fragmentation
- 

# What's New in Authentication

- Fingerprint APIs now part of Android SDK
  - Leveraging OS security in your app
    - Fingerprint
    - Confirm Credentials (Lock Screen)
  - Hardware-Backed(HW) KeyStore updates
    - Not just for RSA anymore
- 

# What You'll Need

- Marshmallow / API 23
- Secure Device
  - Pin, Password or Pattern enabled
- Fingerprint Scanner (for Fingerprint)
  - Android Emulator for testing



# How It Works

- Setup requires generating a key with authentication enabled
  - Key is bound to requested authentication model
  - Stored in HW backed KeyStore
  - Generate a key when
    - Key does not exist
    - Previous key is unrecoverable
- Key is retrieved from HW Backed KeyStore
- Authentication is validated when the key is used
  - Obtaining the key is not enough, you need a Cipher
  - Keys will be invalidated when security profile has been negatively impacted

# Fingerprint

- New manifest permission
  - android.permission.USE\_FINGERPRINT
- FingerprintManager
  - <http://developer.android.com/reference/android/hardware/fingerprint/FingerprintManager.html>
- You control the UI
  - ... with one *minor* exception
    - Must use standard Android fingerprint icon
- Authentication required each time key is retrieved for use
  - Authenticate before using Cipher





# Confirm Credentials

- Access to secure keys is time-based
  - Since last confirmation, not last screen unlock
- KeyguardManager
  - <http://developer.android.com/reference/android/app/KeyguardManager.html>
- System-provided Intent
  - Part of Android SDK since API 21
  - User can authenticate with any valid lock screen mechanism, including fingerprint


Demo



# Flow

- Secret/PrivateKey is retrieved
- Initialize Cipher from Key
  - KeyPermanentlyInvalidatedException
    - Device is not as secure as it once was
  - UserNotAuthenticatedException
    - Request authentication
  - Both extend InvalidKeyException
    - Check instanceof
    - Catch seperately

# Magician's Secret

- `createConfirmCredentialsIntent`
    - `KeyGuardManager`
    - Returns intent that you should launch if authorization is required
  - `authenticate`
    - `FingerprintManager`
    - `CryptoObjects`, `Callback` and `Cancellation`
      - Work required `onPause/onResume`
- 

# Useful Bits

- Simulating Fingerprint on Emulator
  - `adb emu finger touch <#>`
- Power Button on Emulator
  - F7 (Press twice to get to lock screen)
- Marshmallow API Overview
  - <http://developer.android.com/preview/api-overview.html>
- Android KeyStore information
  - <https://developer.android.com/training/articles/keystore.html>
- Google Samples
  - Fingerprint:  
<https://github.com/googlesamples/android-FingerprintDialog>
  - Confirm Credentials:  
<https://github.com/googlesamples/android-ConfirmCredential>

Thank You

