

Developing secure Android for Work apps

Kristian Monsen

Software Engineer
Google



Google for Work | Android

Android Security 2015 Year In Review

April 2016

<http://tinyurl.com/AndroidSecurity2015>

android



Developing Secure Applications for Android



Tips for developing
secure Android apps



Overview of Google Play
Services for secure app
development



Introduce the Application
Security Improvement
Program

Security best practices

For developers of Android applications



SAMSUNG
DEVELOPER
CONFERENCE™

Networking

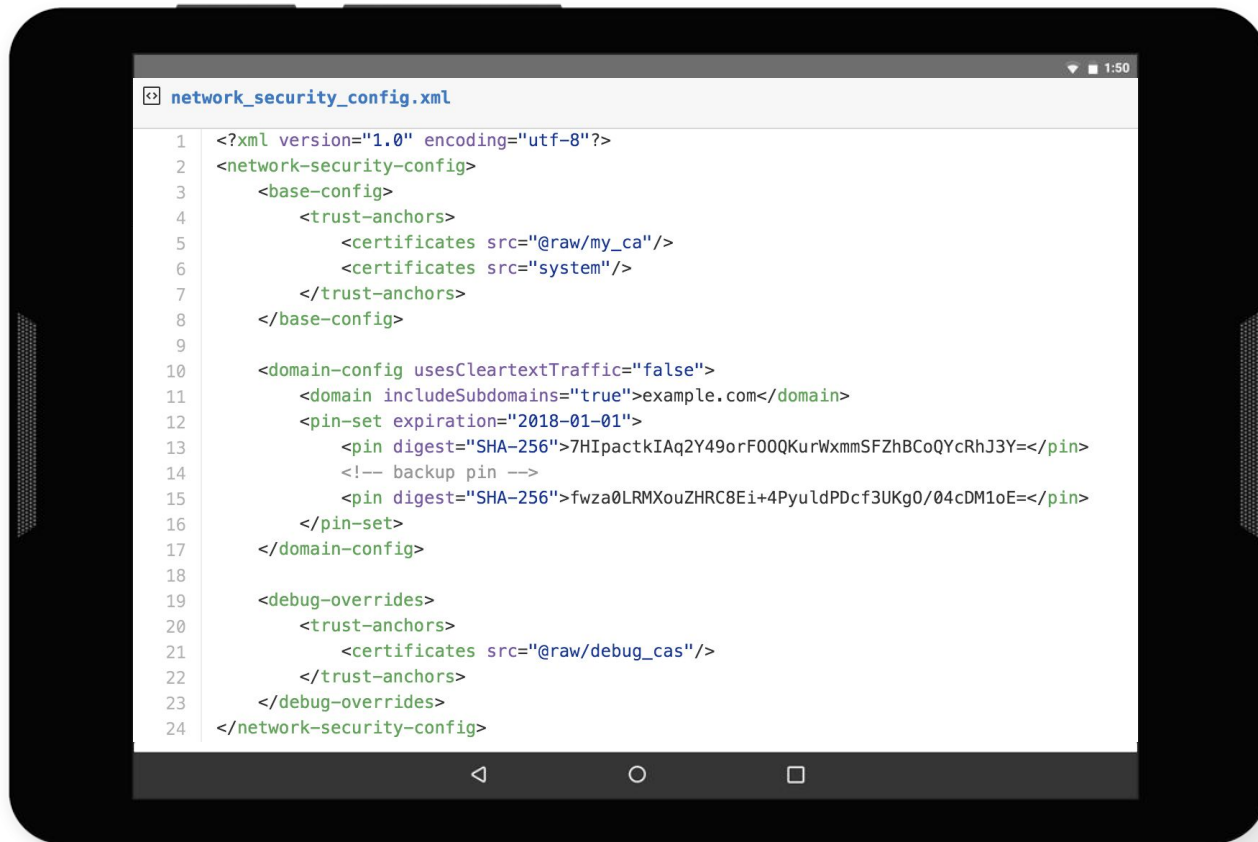
Always use HTTPS

- Should always do this for all network traffic
- Even more important for mobile, devices are often on untrusted networks

Use Android APIs for IPC communication

- Services (Binder or Messenger)
- Intents
- Broadcast Receiver

New networking APIs in N developer preview



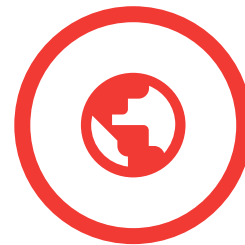
```
network_security_config.xml
1 <?xml version="1.0" encoding="utf-8"?>
2 <network-security-config>
3   <base-config>
4     <trust-anchors>
5       <certificates src="@raw/my_ca"/>
6       <certificates src="system"/>
7     </trust-anchors>
8   </base-config>
9
10  <domain-config usesCleartextTraffic="false">
11    <domain includeSubdomains="true">example.com</domain>
12    <pin-set expiration="2018-01-01">
13      <pin digest="SHA-256">7HIpactkIAq2Y49orF00QKurWxmmSFZhBCoQYcRhJ3Y=</pin>
14      <!-- backup pin -->
15      <pin digest="SHA-256">fwza0LRMXouZHRC8Ei+4PyuIdPDcf3UKg0/04cDM1oE=</pin>
16    </pin-set>
17  </domain-config>
18
19  <debug-overrides>
20    <trust-anchors>
21      <certificates src="@raw/debug_cas"/>
22    </trust-anchors>
23  </debug-overrides>
24 </network-security-config>
```

Storage



Use internal storage provided by Android

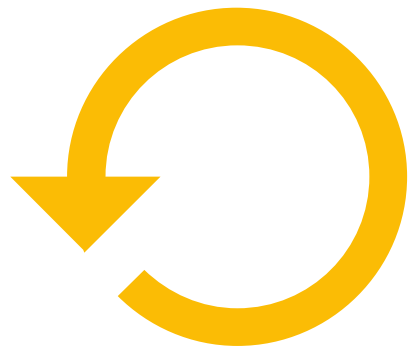
- Only accessible to the current application
- Avoid `MODE_WORLD_(WRITEABLE|READABLE)`
 - ◆ Not fine grained to specific applications
 - ◆ Most used alternative is content provider
- Optional: Encrypt files with key not available on device



External storage is world writeable/readable by default

- Be careful as other apps can read and modify
- This is also true for expansion files (saved to external storage)

Don't dynamically load code



- Large security risk, very difficult to get right
 - ◆ External storage
 - ◆ Insecure network
- Expansion files are in world writeable store and very unsafe
- Adds complexity (testing, version management etc)

Input Validation

- Important on all platforms
- External storage is world writable
- Many issues with native code, but also Java can be vulnerable
- Script injection
- Use well-formatted data formats and verify before using



More information



Links with security tips and best practises for Android

→ Security tips:

<http://developer.android.com/training/articles/security-tips.html>

→ Best practises:

<http://developer.android.com/training/best-security.html>



Play services APIs



SAMSUNG
DEVELOPER
CONFERENCE™

Updating the security provider



Make sure the application is using an updated security provider

There have been various vulnerabilities in the security providers

Example: OpenSSL
(CVE-2014-0224)

Does not work if the developer use
SSLCertificateSocketFactory
directly

It takes up to 350 ms
on older devices

There is an async
and synchronous
method available

Device compatibility attestation

Google API to tell you the CTS compatibility of the device

1

Simple async API

2

Read the response

3

Verify the response

4

Validating
the response
with Google



More information



Links to play services APIs

→ Play services:

https://developers.google.com/android/guides/overview#the_google_play_services_client_library

→ Attestation:

<https://developer.android.com/training/safetynet/index.html>

→ Updating security provider

<http://developer.android.com/training/articles/security-gms-provider.html>



Application Security Improvement Program

How Google can protect users and developers in Google Play



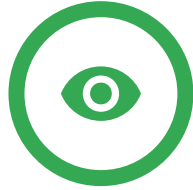
SAMSUNG
DEVELOPER
CONFERENCE™

Program overview

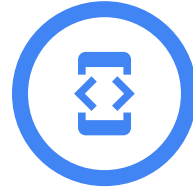


Find vulnerabilities

External reports
Internal research



Scan all apps in the Play store for vulnerability



Notify developer

Dev console
Email to primary contact



Remediation deadline

90 days after notification
No app updates or new apps
with vulnerabilities after this

Example vulnerability

```
OpenSSL Security Advisory [05 Jun 2014]
```

```
=====
```

```
SSL/TLS MITM vulnerability (CVE-2014-0224)
```

```
=====
```

An attacker using a carefully crafted handshake can force the use of weak keying material in OpenSSL SSL/TLS clients and servers. This can be exploited by a Man-in-the-middle (MITM) attack where the attacker can decrypt and modify traffic from the attacked client and server.

The attack can only be performed between a vulnerable client *and* server. OpenSSL clients are vulnerable in all versions of OpenSSL. Servers are only known to be vulnerable in OpenSSL 1.0.1 and 1.0.2-beta1. Users of OpenSSL servers earlier than 1.0.1 are advised to upgrade as a precaution.

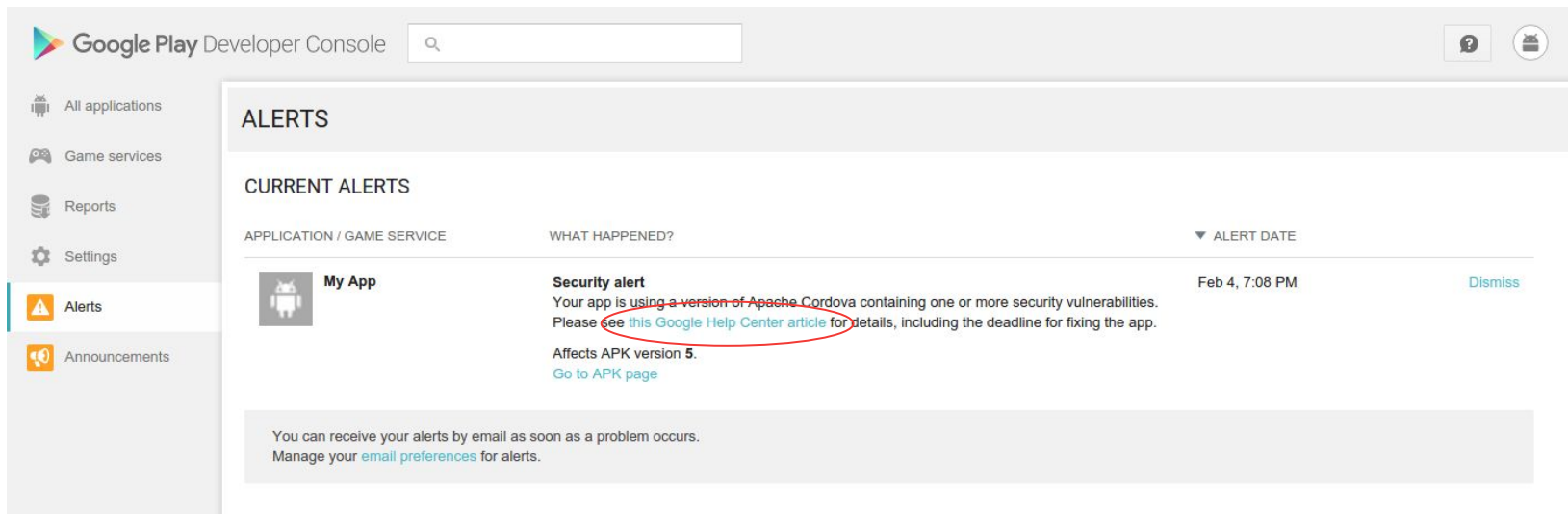
OpenSSL 0.9.8 SSL/TLS users (client and/or server) should upgrade to 0.9.8za.
OpenSSL 1.0.0 SSL/TLS users (client and/or server) should upgrade to 1.0.0m.
OpenSSL 1.0.1 SSL/TLS users (client and/or server) should upgrade to 1.0.1h.

Thanks to KIKUCHI Masashi (Lepidum Co. Ltd.) for discovering and researching this issue. This issue was reported to OpenSSL on 1st May 2014 via JPCERT/CC.

The fix was developed by Stephen Henson of the OpenSSL core team partly based on an original patch from KIKUCHI Masashi.

Notify Developer


Message on the Play Developer Console



Google Play Developer Console

ALERTS

CURRENT ALERTS

APPLICATION / GAME SERVICE	WHAT HAPPENED?	ALERT DATE	
 My App	Security alert Your app is using a version of Apache Cordova containing one or more security vulnerabilities. Please see this Google Help Center article for details, including the deadline for fixing the app. Affects APK version 5. Go to APK page	Feb 4, 7:08 PM	Dismiss

You can receive your alerts by email as soon as a problem occurs.
Manage your [email preferences](#) for alerts.

Contains a link to a help article explaining more about the vulnerability

Notify developer

Email to primary contact

Vulnerability type

Remediation details

Relevant Play policy

Affected apps

Google Play <googleplay-noreply@google.com> Mar 31 ☆ ↩

to me ▾

Hello Google Play Developer,

Your app(s) listed at the end of this email utilize a version of OpenSSL that contains one or more security vulnerabilities. If you have more than 20 affected apps in your account, please check the [Developer Console](#) for a full list.

Please migrate your app(s) to OpenSSL 1.02f1.01r or higher as soon as possible and increment the version number of the upgraded APK. Beginning July 11, 2016, Google Play will block publishing of any new apps or updates that use older versions of OpenSSL.

The vulnerabilities were addressed in OpenSSL 1.02f1.01r. The latest versions OpenSSL can be downloaded [here](#). To confirm your OpenSSL version, you can do a grep search for (\$ unzip -p YourApp.apk | strings | grep "OpenSSL").

If you're using a 3rd party library that bundles OpenSSL, you'll need to upgrade it to a version that bundles OpenSSL 1.02f1.01r or higher.

To confirm you've upgraded correctly, submit the updated version to the Developer Console and check back after five hours. If the app hasn't been correctly upgraded, we will display a warning.

The vulnerabilities include "logjam" and [CVE-2015-3194](#). The Logjam attack allows a man-in-the-middle attacker to downgrade vulnerable TLS connections to 512-bit export-grade cryptography. This allows the attacker to read and modify any data passed over the connection. Details about other vulnerabilities are available [here](#). For other technical questions, you can post to [Stack Overflow](#) and use the tags "android-security" and "OpenSSL."

While these specific issues may not affect every app that uses OpenSSL, it's best to stay up to date on all security patches. Apps with vulnerabilities that expose users to risk of compromise may be considered in violation of our [Malicious Behavior policy](#) and section 4.4 of the Developer Distribution Agreement.

Apps must also comply with the [Developer Distribution Agreement](#) and [Developer Program Policies](#). If you feel we have sent this warning in error, contact our policy support team through the [Google Play Developer Help Center](#).

Regards,

The Google Play Team

©2016 Google Inc. 1600 Amphitheatre Parkway, Mountain View, CA 94043

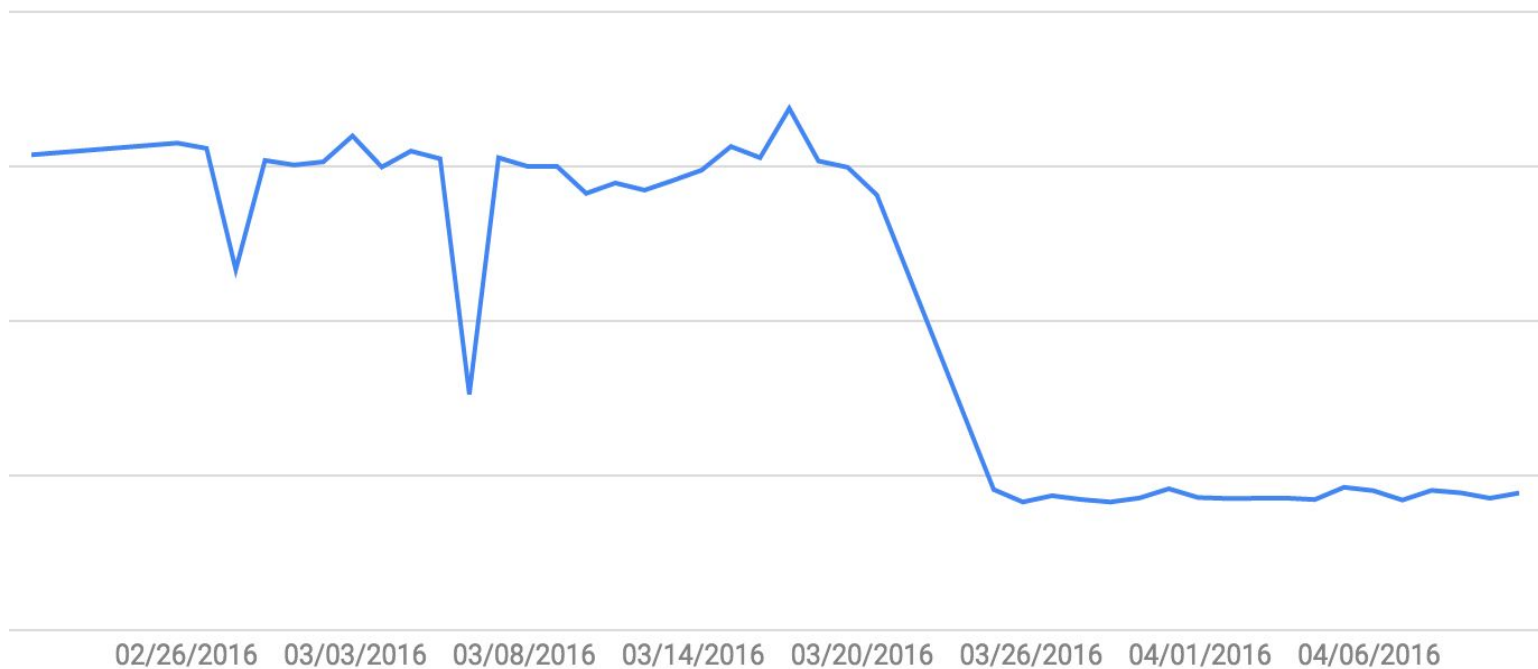
Email preferences: You have received this mandatory email service announcement to update you about important changes to your Google Play Developer account.

Affected App(s) and Version(s):
com.slothplay.A147e175e8b.escapebbasement 1000005



Remediation deadline

Typical campaign progression, user installations

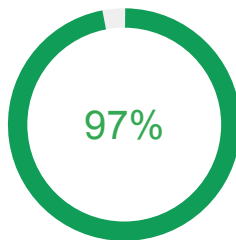


Current results

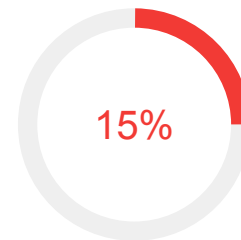
Mostly successful, but not always



15 campaigns
done so far



Best result:
installed apps are fixed



Worst result:
installed apps are fixed, this was a
warning only campaign

*It strongly depends if it is a warning and if it
has a deadline*

Other activities



- Add lint warnings for Android Studio
- Improve APIs so apps are safe by default

Tips about vulnerabilities

If you know of any vulnerability we should scan for we are always interested

Send email to security@android.com

Report security bug:

<https://source.android.com/security/overview/updates-resources.html#report-issues>

Q & A

and **THANK YOU** for your time.

Kristian Mosen

kristian@android.com

www.SDC2016.com

