DEFENDING (MORE THAN) ANDROID APPS WITH TLS CERTIFICATE PINNING AND NOTARY SERVICES

Damjan Buhov, MSc
Is my communication secure?
Is my communication secure?
Is my communication secure?
Is my communication secure?
Is my communication secure?
Is my communication secure?
Is my communication secure?
Is my communication secure?
Motivation

• 2 billion smartphones & 2,2+ million apps
• **MUST** meet certain criteria
• Process sensitive user information
• Present not so obvious threats for end users
• Users rely on developer‘s capability and knowledge
Motivation

• Network utilization
  o (In) Secure transfer of sensitive personal data
  o (In) Correct use of SSL/TLS protocols
    – Custom implementations
      ➢ Tested 13,000 apps, 1,000+ broken
  o Lack of centralized body that will control the implementation of specific security mechanisms

---

TLS connection setup

- Handshake protocol
- Four phases
- In phase 2:
  - The client checks the validity of the server certificate
  - Uses CA information available on the trust store
Certificate Validation

• Is the Certificate authentic?
  o Is it signed by a trusted Certificate Authority?
  o Is it expired?
  o Does the hostname matches the subjectAltname or CN?
  o Is the certificate revoked?
SSL/TLS & Android apps

- Most common protection mechanism
- Allows the client (app) to authenticate the legitimate server
- Provides secure network communication
- 150 trusted certificates
  - By whom?
SSL/TLS & Android apps

• Default HTTPS API in Android implements proper certificate validation
SSL/TLS & Android apps

• Default HTTPS API in Android implements proper certificate validation

What would be the problem?
The central role of CAs

• Any CA can issue/sign a certificate for any address
  o Modern OS come pre-shipped with hundreds of CAs which are inherently trusted
• What if a CA is compromised?
  o e.g., Comodo, Diginotar
• What if a CA issues fake cert “for security”?
  o e.g., WoSign/github
• What if the trust store is manipulated?
  o e.g., Superfish and eDellRoot
The central role of CAs

- Any CA can issue/sign a certificate for any address
- Modern OS come pre-shipped with hundreds of CAs which are inherently trusted

What if a CA is compromised?
- e.g., Comodo, DigiNotar

What if a CA issues fake cert “for security”?
- e.g., WoSign / github

What if the trust store is manipulated?
- e.g., Superfish and eDellRoot
The central role of CAs

- Any CA can issue/sign a certificate for any address
- Modern OS come pre-shipped with hundreds of CAs which are inherently trusted

What if a CA is compromised?
- e.g., Comodo, DigiNotar

What if a CA issues fake cert “for security”?
- e.g., WoSign, GitHub

What if the trust store is manipulated?
- e.g., Superfish and DellRoot

The central role of CAs
Custom handling of certificates

• “Custom” logic
  o Usually copied from popular tech forums

• Using self-signed certificates
  o For testing and sometimes even for production

• Implementation of advanced security concepts
  o e.g., certificate pinning
Problems with custom code handling

- Common problem
  - Broken implementations
- Lack of knowledge
- Not easy to detect
- No requirement that regulates its use
- Update interval

---

Static analysis

- Indicative experiment on 50,000 apps\(^3\)
  - Top 25,000 from Q4/2013
  - Top 25,000 from Q4/2014
- Determine the current situation
  - Not tracking evolution of apps
- Focused explicitly on **custom TrustManager** implementations
- **Mallodroid** script

Results

Apps 2013

- Trust Manager: 17% Broken, 6% Possibly Broken
- Hostname Verifier: 7% Broken, 1% Possibly Broken
- TLS Error Handling: 15% Broken
Results

Apps 2014

- Trust Manager: 23% Broken, 10% Possibly Broken
- Hostname Verifier: 13% Broken, 4% Possibly Broken
- TLS Error Handling: 29% Broken, 0.05% Possibly Broken
Proposed fix

• Protects against developer’s errors in cert handling
• Combination of dynamic instrumentation techniques and certificate pinning
• Brings the control back to the user
• No modification of the \texttt{apk} file
• Works at runtime
• Dynamic patching of Android applications
• User is alerted upon every suspicious certificate change
Certificate Pinning

• Current solutions – Application based
  o AndroidPinning
  o Cert-pinner
  o …

• Our proof-of-concept⁴ – Device based
  o Installs as any other app
  o Can be integrated in the Android OS as default setting
  o No application limitation

⁴ PinningTrustManager - https://github.com/dbuhov/pinningTrustManager
TLS Notary Service

- Principle of multi-path probing
- Do we see the same certificate?
Implementation

• Prerequisites
  o Cydia Substrate
  o Android device with root access
  o Knowledge about the instrumented part

• Java Code
  o Hooking class
  o Custom Trust Manager class
Implementation

• Prerequisites
  o Cydia Substrate
  o Android device with root access
  o Knowledge about the instrumented part

• Java Code
  o Hooking class
  o Custom Trust Manager class

Diagram:
- Applications
- Application Framework
- Libraries
- Android Runtime
- Linux Kernel
  - Notary assisted pinning
Check stored pins
Cert. handling in Android 7.0 Nougat

- New approach – config file
- Much easier implementation/integration
- Happy developers = No more custom logic
- Happy users?
Security Implications

• Update interval
  o What happens when the cert expires?
• Possibility to hook and modify the configuration file
• Affected Devices – Previous OS versions?

Changes to Trusted Certificate Authorities in Android Nougat

Posted by Chad Brubaker, Android Security team

In Android Nougat, we've changed how Android handles trusted certificate authorities (CAs) to provide safer defaults for secure app traffic. Most apps and users should not be affected by these changes or need to take any action. The changes include:

• Safe and easy APIs to trust custom CAs.
• Apps that target API Level 24 and above no longer trust user or admin-added CAs for secure connections, by default.
• All devices running Android Nougat offer the same standardized set of system CAs—no device-specific customizations.
Conclusion

Dynamic patching of SSL/TLS issue in Android apps

Works for all apps

Brings control back to the users
Thank you!

Damjan Buhov, MSc

dbuhov@sba-research.org
damjan.buhov@fhstp.ac.at