Securing Network Communications: An Investigation Into CAs On Mobile

Andrew Blaich | October 18, 2016
Overview

• Background info on CAs and trusted communications
• CAs on Mobile
• CA Classifications
• How are the top mobile apps verifying trust?
• How are mobile operating system making advancements?
About Me

- **Andrew Blaich**
  - Lookout Security (previously at Bluebox Security)
    - Manager of Vulnerability Research
  - Ph.D. from University of Notre Dame
  - Reverser of Android, iOS, and IoT things
What is a CA?

- Certificate Authorities are the providers of trust for our communications over the Internet.
  - The Internet’s security is built on top of trusted secure transactions
  - CAs provide assurance of the identity of a web server (trust chain)
  - Self-signed certs may not provide this assurance
Trust Store

- Your devices have a “Trust Store” that contains root certificates it knows about and trusts.
Trusted Certificate Chain

Verified == Trusted Chain

The root CA to verify this chain is installed on the device, thus making the chain verifiable and trusted.
Site Verification - Success

Are you Google.com?
Yes, CA-A says I am.

TRUSTED CONNECTION
Un-trusted Certificate Chain

The root CA to verify this chain is missing from the device making the trust chain un-verifiable.

The same happens for self-signed certs.
Site Verification - Failure

Are you Google.com?

Yes, CA-M says I am.

Trust Store
CA-A

"Google"

NOT A TRUSTED CONNECTION
Un-trusted Certificate Chain in the Wild

hey @Gogo, why are you issuing *.google.com certificates on your planes?
Un-trusted Certificate Chain in the Wild 2
Certificate Authorities on Mobile
Certificate Authorities

• How do CAs make it to my device?
• What certificate authorities are on my device?
  • System installed vs. user installed
• How many are there?
• What security concerns are there?
Root CA Approval Process

Root Certificate Programs

Mozilla  Microsoft  Apple  others
Mozilla Root CA Approval Process

How a CA gets included into Firefox
https://wiki.mozilla.org/CA:How_to_apply#Timeline

The whole process can take approximately 11 months or more.

Linux and Android are strongly tied to the Mozilla process.
CA Trust Infrastructure

- The effectiveness of the global PKI trust infrastructure relies on keeping the designated roots of trust fully secure and operating correctly. A CA can tell a device a site is who it says it is.

**Trusted Root CAs**

- CA-A: Issue cert for *.google.com - No
- CA-B: Issue cert for *.google.com - Yes

- Compromised CA
How many root certs?

<table>
<thead>
<tr>
<th>Mobile OS</th>
<th>Cert Count</th>
<th>Mobile OS</th>
<th>Cert Count</th>
<th>Mobile OS</th>
<th>Cert Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Android 5.*</td>
<td>162</td>
<td>Android 6.*</td>
<td>158</td>
<td>Android 7.*</td>
<td>148</td>
</tr>
<tr>
<td>iOS 8.*</td>
<td>223</td>
<td>iOS 9.*</td>
<td>200</td>
<td>iOS 10.*</td>
<td>177</td>
</tr>
<tr>
<td>- iOS 8.* Trusted</td>
<td>210</td>
<td>- iOS 9.* Trusted</td>
<td>187</td>
<td>- iOS 10.* Trusted</td>
<td>165</td>
</tr>
<tr>
<td>- iOS 8.* Always Ask</td>
<td>13</td>
<td>- iOS 9.* Always Ask</td>
<td>13</td>
<td>- iOS 10.* Always Ask</td>
<td>12</td>
</tr>
<tr>
<td>- iOS 8.* Blocked</td>
<td>17</td>
<td>- iOS 9.* Blocked</td>
<td>18</td>
<td>- iOS 10.* Blocked</td>
<td>26</td>
</tr>
</tbody>
</table>
OS Version Distribution - Root Cert Changes

iOS Versions
- iOS 10: 54%
- iOS 9: 38%
- Earlier: 8%

Android Versions
- Android 6: 36.00%
- Android 5.*: 27.70%
- Android 4.4.*: 18.70%
- Android 4.1.* - 4.3.*: 15.60%
- Android 4.0.*: 1.60%
- Android 2: 1.40%
Root CA Reference Links

- **iOS:**

- **Android:**
  - [https://android.googlesource.com/platform/system/ca-certificates/+/master](https://android.googlesource.com/platform/system/ca-certificates/+/master)
“...it will likely take years to reduce the number of users and devices at risk from certificates issued by Symantec from this root...” - Ryan Sleevi
Root Cert Changes - OEMs

- **BlackBerry Priv (Android)**
  - 17 Supplemental Certificates:
    - att_suplcert1_v0.der
  - **SHA-1 Hash:**
SHA-1 Deprecation

• Community Controversy
• Among the mobile platforms, Android added SHA256 support in version 2.3. Earlier versions—“still used in large numbers”—support only SHA1.
  • Firefox 43 did not validate against new SHA-1 CAs from 1/1/2016 or after (reverted); Now fixed.
• CAs caught issuing SHA-1 certificates are running into trouble:
  • WoSign
  • StartCom
Root Cert Changes - Let’s Encrypt

How does your browser or device already trust Let’s Encrypt?

ISRG Root X1 is not yet trusted in most browsers (or devices), e.g.
https://bugzilla.mozilla.org/show_bug.cgi?id=1204656
https://bugs.chromium.org/p/chromium/issues/detail?id=531672
https://code.google.com/p/android/issues/detail?id=186312

Source: https://letsencrypt.org/certificates/
CA Classifications
CA Classifications

• Known Failures in Keeping Trust
• Government-Based Roots of Trust
• Cause for Concern
• Artificial Constraints
• Everything else
Known Failures with CAs

• **Highlights in failures of trust:**
  - GlobalSign [2016]
  - StatCom/WoSign [2016]
  - Symantec [2015]
  - CNNIC/MCS Holdings [2015]
  - Comodo [2011]
  - DigiNotar [2011]
  - GlobalSign [2011]
  - India CCA [2014]
  - RapidSSL (indirect) [2008]
WoSign/StartCom Controversy

Mozilla wants woeful WoSign certs off the list
Backdating SHA-1 certs is just not on

SHA-1 backdating, acquisition of StartCom, and miss issuance of certs
http://www.theregister.co.uk/2016/09/27/mozilla_wants_woeful_wosign_certs_off_the_list/
https://docs.google.com/document/d/1C6BlmbeQfn4a9zydVi2UvjBGv6szuSB4sMYUcVrR8vQ/preview#
http://thehackernews.com/2016/08/github-ssl-certificate.html
https://groups.google.com/forum/?nomobile=true#!topic/mozilla.dev.security.policy/BV5XyFJLnQM
WoSign CA Blocking

Blocking Trust for WoSign CA Free SSL Certificate G2

Certificate Authority WoSign experienced multiple control failures in their certificate issuance processes for the WoSign CA Free SSL Certificate G2 intermediate CA. Although no WoSign root is in the list of Apple trusted roots, this intermediate CA used cross-signed certificate relationships with StartCom and Comodo to establish trust on Apple products.

In light of these findings, we are taking action to protect users in an upcoming security update. Apple products will no longer trust the WoSign CA Free SSL Certificate G2 intermediate CA.

To avoid disruption to existing WoSign certificate holders and to allow their transition to trusted roots, Apple products will trust individual existing certificates issued from this intermediate CA and published to public Certificate Transparency log servers by 2016-09-19. They will continue to be trusted until they expire, are revoked, or are untrusted at Apple’s discretion.

As the investigation progresses, we will take further action on WoSign/StartCom trust anchors in Apple products as needed to protect users.
WoSign/StartCom Revocation

- Mozilla’s Actions:
  1) Distrust certificates chaining up to Affected Roots with a notBefore date after October 21, 2016. If additional back-dating is discovered (by any means) to circumvent this control, then Mozilla will immediately and permanently revoke trust in the Affected Roots.
  2) Add the previously identified backdated SHA-1 certs chaining up to the Affected Roots to OneCRL.
  3) No longer accept audits carried out by Ernst & Young Hong Kong.
  4) Remove the Affected Roots from NSS after the SSL certificates issued before October 1, 2016, have expired or have been replaced.

https://groups.google.com/forum/?nomobile=true#!topic/mozilla.dev.security.policy/BV5XyFJLnQM
GlobalSign Controversy

GlobalSign screw-up cancels top websites' HTTPS certificates

Revoked certs may linger for days, locking people out of sites

4 days to correct itself

Dear Valued GlobalSign Customer,

As most of you are aware, we are experiencing an internal process issue (details below) that is impacting your business. While we have identified the root cause, we deeply apologize for the problems this is causing you and wanted to ensure you that we are actively resolving the issue.

GlobalSign manages several root certificates and for compatibility and browser ubiquity reasons provides several cross-certificates between those roots to maximize the effectiveness across a variety of platforms. As part of a planned exercise to remove some of those links, a cross-certificate linking two roots together was revoked. CRL responses had been operational for 1 week, however an unexpected consequence of providing OCSP responses became apparent this morning, in that some browsers incorrectly inferred that the cross-signed root had revoked intermediates, which was not the case.

GlobalSign has since removed the cross-certificate from the OCSP database and cleared all caches. However, the global nature of CDNs and effectiveness of caching continued to push some of those responses out as far as end users. End users cannot always easily clear their caches, either through lack of knowledge or lack of permission. New users (visitors) are not affected as they will now receive good responses.

The problem will correct itself in 4 days as the cached responses expire, which we know is not ideal. However, in the meantime, GlobalSign will be providing an alternative issuing CA for customers to use instead, issued by a different root which was not affected by the cross that was revoked, but offering the same ubiquity and does not require to reissue the certificate itself.

We are currently working on the detailed instructions to help you resolve the issue and will communicate those instructions to you shortly.

Thank you for your patience.

Lisa Xue
Chief Product Officer
GMO GlobalSign

US +1 803-570-7060 | UK +44 1622 766 786 | EU +32 16 86 1900

www.globalsign.com/en

https://downloads.globalsign.com/acton/fs/blocks/showLandingPage/a/2674/p/p-008f/t/page/fm/0
Symantec Controversy

Don't panic, says Blue Coat, we're not using CA cert to snoop on you
Symantec and partner say HTTPS certificate-issuing powers used only for testing

Symantec employees fired for issuing rogue HTTPS certificate for Google
Unauthorized credential was trusted by all browsers, but Google never authorized it.

What's a certificate? Photo source: Shutterstock

Discontinued Use of VeriSign G1 Roots

Printer Friendly

Alerts ID: ALERT1941

Updated: 12/14/2015
CNNIC/MCS Controversy

Google to drop China’s CNNIC Root Certificate Authority after trust breach

CNNIC Root was removed from Chrome in April 2015

CNNIC Root became partially trusted in iOS in June 2015

CNNIC Root was removed from Android in the source March 2016, was finally removed in Android 7.0.
https://android.googlesource.com/platform/system/ca-certificates/+/c36d8eb8071d73528993e024e73d40c6977b1d0d
Government CAs
Government Related CAs

Government CAs
- Govt. of Spain
  - Izenpe S.A.
- Govt. of Spain
  - ACCV
- Govt. of The Netherlands
  - Staat der Nederlanden
- Govt. of Hong Kong
  - Hong Kong Post Root
- Govt. of South Korea
  - KISA
- Govt. of Japan
- Spanish FNMT
- Govt. of France
  - DCSSI
- Govt. of Taiwan
  - Government Root CA
- Govt. of Turkey
  - Kamu SM

Suspected of affiliation with a government entity
- JCSI SecureSign
- Agencia Catalana de Certificacio
- CNNIC
- Elektronik Bilgi Guvenligi A.S.
- HARICA
- SOCIEDAD CAMERAL DE CERTIFICACION
- Chunghwa Telecom Co
- e-tugra
- TURKTRUST

Other nationally-operating entities
- French Postal Service
  - Certinomis
- Switzerland Post Office
  - SwissSign

Allowed to use an internal audit for approval.
Causes for Concern - CAs
Causes for Concern

**Issued improper certificates**
- SecureTrust Corporation
- Secure Global CA
- SecureTrust CA
- Government of France (PM/SGDN)
- DCSSI
- TÜRKTRUST
- TÜRKTRUS T .... (c) Aralık 2007

** Deprecated**
- Certicámara S.A.
- AC Raíz Cericámara S.A.
- TDC Internet
- TDC Internet Root CA
Causes for Concern – cont’d.

**Community Controversy**

- Staat der Nederlanden
  - Staat der Nederlanden Root CA
  - Staat der Nederlanden Root CA - G2
  - HARICA
    - Hellenic Academic and Research Institutions RootCA 2011

- StartCom Ltd.
  - StartCom Certification Authority
  - StartCom Certification Authority G2
  - CNNIC
    - CNNIC ROOT
Causes for Concern – cont’d.

Certificate Authorities using a 1024 bit key

- FNMT Class 2 CA
- CyberTrust Global Root
- Equifax Secure Certificate Authority
- GTE CyberTrust Global Root
- Netlock Uzleti (Class B)
- DigiCert Class 2 Public Primary Certification Authority
- ValiCert Class 1 Policy Validation Authority
- VeriSign Class 3 Public Primary Certification Authority
- Thawte Consulting cc Thawte Premium Server CA
- Thawte Consulting cc Thawte Server CA
- Entrust.net Secure Server Certification Authority

The diagram shows a red X over the certificate authorities that use a 1024-bit key, indicating a concern.
<table>
<thead>
<tr>
<th>Cert Subject</th>
<th>Reason For Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>CN=IGC/A,OU=DCSSI,O=PM/SGDN,L=Paris,ST=France,C=FR</td>
<td>Issued several un-authorized certificates for Google domains. TLD restrictions: .fr (France), .gp (Guadeloupe), .gf (Guyane), .mq (Martinique), .re (Réunion), .yt (Mayotte), .pm (Saint-Pierre et Miquelon), .bl (Saint Barthélemy), .mf (Saint Martin), .wf (Wallis et Futuna), .pf (Polynésie française), .nc (Nouvelle Calédonie), .tf (Terres australes et antarctiques françaises)</td>
</tr>
</tbody>
</table>
Artificial Constraints - cont’d.

https://bugzilla.mozilla.org/show_bug.cgi?id=948175
CA Cryptography Analysis
## Public Key-Size Stats

<table>
<thead>
<tr>
<th>Key Type/Size</th>
<th>5.X</th>
<th>6.X</th>
<th>7.X</th>
<th>Notable Entities for Latest</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSA / 1024 bit</td>
<td>15</td>
<td>5</td>
<td>0</td>
<td>none</td>
</tr>
<tr>
<td>RSA / 2048 bit</td>
<td>101</td>
<td>96</td>
<td>89</td>
<td>N/A</td>
</tr>
<tr>
<td>RSA/ 4096 bit</td>
<td>14</td>
<td>46</td>
<td>47</td>
<td>N/A</td>
</tr>
<tr>
<td>Elliptic Curve</td>
<td>6</td>
<td>11</td>
<td>12</td>
<td>GeoTrust, VeriSign, COMODO, Thawte, Entrust, AffirmTrust</td>
</tr>
</tbody>
</table>

47
# Hash Algorithm Stats

<table>
<thead>
<tr>
<th>Signature Algorithm</th>
<th>5.X</th>
<th>6.X</th>
<th>7.X</th>
</tr>
</thead>
<tbody>
<tr>
<td>md5WithRSAEncryption</td>
<td>6</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>sha1WithRSAEncryption</td>
<td>115</td>
<td>98</td>
<td>85</td>
</tr>
<tr>
<td>sha256WithRSAEncryption</td>
<td>28</td>
<td>43</td>
<td>46</td>
</tr>
<tr>
<td>sha384WithRSAEncryption</td>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>sha512WithRSAEncryption</td>
<td>--</td>
<td>--</td>
<td>1</td>
</tr>
<tr>
<td>ecdsa-with-SHA256</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>ecdsa-with-SHA384</td>
<td>6</td>
<td>10</td>
<td>11</td>
</tr>
</tbody>
</table>
### CA Consolidation

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>GeoTrust</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>VeriSign</td>
<td>7</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>TC Trust Center</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Thawte</td>
<td>5</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Equifax</td>
<td>3</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td><strong>Total:</strong></td>
<td><strong>15.4%</strong></td>
<td><strong>11.4%</strong></td>
<td><strong>9.5%</strong></td>
</tr>
</tbody>
</table>

Additional CAs

• Some OEMs and carriers add additional certificates into the ROM that are not found in the standard roots of trust (i.e. NSS):
  • Sony Xperia running 4.4.4 includes two root certs for Sony
  • iOS has several additional certificates that Android does not currently have e.g.: Cisco and US Government certs
User installed root CAs

Network monitoring

A third party is capable of monitoring your network activity, including emails, apps, and secure websites.

A trusted credential installed on your device is making this possible.

Check trusted credentials

CONFIGURATION PROFILE

mitmproxy
Case Study: Mobile Apps and Trust
Mobile apps and trust

- Who are the top Android apps trusting?
- Most apps trust the certificates on your device
  - Your browser may distrust the CNNIC root cert, but your mobile app may trust it
  - Some apps even disable hostname verifications (trusting everything)
### Top Android Apps

<table>
<thead>
<tr>
<th>X509HostnameVerifier</th>
<th>% of Apps</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ALLOW_ALL_HOSTNAME</strong></td>
<td>35% -&gt; 29% 1&lt;sup&gt;st&lt;/sup&gt; party code;</td>
</tr>
<tr>
<td></td>
<td>86% 3&lt;sup&gt;rd&lt;/sup&gt; party code</td>
</tr>
<tr>
<td><strong>STRICT_HOSTNAME</strong></td>
<td>30.31%</td>
</tr>
<tr>
<td><strong>BROWSER_COMPATIBLE_HOSTNAME</strong></td>
<td>19.29%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Other Features</strong></th>
<th><strong>Number of Apps</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Custom Trust Store</strong></td>
<td>6.30%</td>
</tr>
<tr>
<td></td>
<td><strong>Average</strong>: 13 CAs; <strong>Min</strong>: 1 CA (8 apps); <strong>Max</strong>: 129 CAs (1 app)</td>
</tr>
</tbody>
</table>

---

54
Protecting Yourself
Protecting yourself

- Beware of government controlled root CAs, example: Kazakhstan (December 2015)
  - User installed (at this time)
  - [https://bugzilla.mozilla.org/show_bug.cgi?id=1232689](https://bugzilla.mozilla.org/show_bug.cgi?id=1232689) [not added to NSS]
  - [https://bugzilla.mozilla.org/show_bug.cgi?id=1229827](https://bugzilla.mozilla.org/show_bug.cgi?id=1229827)

**Kazakhstan will force its citizens to install internet backdoors**

The poorly thought-out and crude surveillance technique could have a devastating effect on the country’s internet security.
Mobile OS Enhancements
iOS Enhancements

- iOS 9 added “App Transport Security”

**HTTPS by default**
**TLS 1.2 w/ Forward Secrecy**

### App Transport Security

App Transport Security (ATS) enforces best practices in the secure connections between an app and its back end. ATS prevents accidental disclosure, provides secure default behavior, and is easy to adopt; it is also on by default in iOS 9 and OS X v10.11. You should adopt ATS as soon as possible, regardless of whether you’re creating a new app or updating an existing one.

If you’re developing a new app, you should use HTTPS exclusively. If you have an existing app, you should use HTTPS as much as you can right now, and create a plan for migrating the rest of your app as soon as possible. In addition, your communication through higher-level APIs needs to be encrypted using TLS version 1.2 with forward secrecy. If you try to make a connection that doesn’t follow this requirement, an error is thrown. If your app needs to make a request to an insecure domain, you have to specify this domain in your app's Info.plist file.
iOS Enhancements

- **iOS 10 added “Certificate Transparency” and deprecated older algorithms.**

**Certificate Transparency and Certificate Trust APIs**

Strong encryption for your network connections is not enough. To help ensure your app is connecting to the right server, employ Apple’s certificate trust APIs and Certificate Transparency.

- Certificate, Key, and Trust Services Programming Guide
- Certificate Transparency website
- iOS trusted root certificates.

**Supported Algorithms**

With iOS 10 and macOS v10.12, the RC4 cipher suite is now disabled by default. In addition, Apple recommends that you upgrade your servers to use certificates signed with the SHA-2 cryptographic function.
Android Enhancements

• Android 7.0 (Nougat) added “Network Security Configuration”

The Network Security Configuration feature lets apps customize their network security settings in a safe, declarative configuration file without modifying app code. These settings can be configured for specific domains and for a specific app. The key capabilities of this feature are as follows:

• **Custom trust anchors**: Customize which Certificate Authorities (CA) are trusted for an app’s secure connections. For example, trusting particular self-signed certificates or restricting the set of public CAs that the app trusts.

• **Debug-only overrides**: Safely debug secure connections in an app without added risk to the installed base.

• **Cleartext traffic opt-out**: Protect apps from accidental usage of cleartext traffic.

• **Certificate pinning**: Restrict an app’s secure connection to particular certificates.

https://android-developers.blogspot.com/2016/07/changes-to-trusted-certificate.html
Wrapping Up
Managing your Security

• Traveling or worried about security?
  • Remove or disable any CAs that concern you (government, deprecated, or others)
    • Test that the sites you need access to work with your restricted CA list
  • Be wary of installing **iOS profiles**, **device admin apps**, or **3rd party certificates**
    • Check your Security (Android) and Profile (iOS) settings
    • Some organizations use these to manage your device (check with your IT department)
    • Manage/disable root access when traveling (malware can exploit this)
  • Disable/remove apps you don’t need (they’re chatty and may be using insecure communications)
  • Beware of free / un-secured WiFi
Managing your CAs

- **Android:**
  - System Settings
    - Settings -> Security -> Trusted credentials
    - Disable or Enable each CA
  - Programmatically or via shell
    - (requires root access)

- **iOS:**
  - No manual method on iOS
  - Programmatically or via shell
    - (requires root access)

- **Common:**
  - Certificate Pinning
  - Monitor for MITM attacks
Trust Managers

Protect Yourself

- **Executive / Management Team**
  - Secure the apps and data your company and employees use
  - Always use secure communications and apps; be aware of the trust chain

- **Developer**
  - Apply browser strategies for root certs to your app
  - Check, validate, and enforce the expected chain of trust to your servers
  - OCSP Checks
  - Utilize features of the latest OS platforms

- **Consumer**
  - Review, disable, and remove certificates you don’t trust
  - Keep your trust store up to date
Summary

- CAs provide the trust for the internet’s security model
- CAs are known to have failures in trust
- Mobile operating systems vary with their support for the vast number of CAs
- Mobile apps should not rely on the device (or CAs) to be trustable
- Users can take action to reduce the amount of 3rd parties they trust.
- Protect yourself
Q&A

• Contact Information:
  • Andrew Blaich
    • Twitter: @ablaich
    • Email: andrew.blaich@lookout.com