Studying TLS Usage in Android Apps

Abbas Razaghpanah, Arian Akhavan Niaki, Narseo Vallina-Rodriguez, Srikanth Sundaresan, Johanna Amann, Phillipa Gill

Paper: ACM CoNext 2017
Encryption is Everywhere

It’s Time to Encrypt the Entire Internet

Encrypted Web Traffic More Than Doubles After NSA Revelations
However...

- TLS is also an important component of mobile applications
  - 88% of Android applications use TLS
- Unlike Web browsers and servers...
  - ...many application developers implementing TLS
- ...many opportunities to make errors!
Understanding TLS on Android

- Understanding of TLS on Android has been limited …
- Static analysis: Explores all code paths, but not necessarily those taken in practice
- Dynamic analysis: May not cover all code paths

Our Solution: Lumen

- User space traffic monitoring on Android
- Crowd source measurements of application behavior
- Collect anonymized TLS handshake data between apps and servers
...Wait a minute

- Our study is deemed to be non-human-subject research by UC Berkeley’s IRB

- We collect no private information of traffic (encrypted or unencrypted)

- All web browser traffic is excluded

- We are studying software, not people

- We have a comprehensive consent process in place
Wait a minute

- Our study is deemed to be non-human-subject research by UC Berkeley's IRB.
- We collect no private information of traffic (encrypted or unencrypted).
- All web browser traffic is excluded.
- We are studying software, not people.
- We have a comprehensive consent process in place.
...Wait a minute

- Our study is deemed to be non-human-subject research by UC Berkeley's IRB.
- We collect no private information of traffic (encrypted or unencrypted).
- All web browser traffic is excluded.
- We are studying software, not people.
- We have a comprehensive consent process in place.
…Wait a minute

• Our study is deemed to be non-human-subject research by UC Berkeley's IRB.
• We collect no private information (encrypted or unencrypted).
• All web browser traffic is excluded.
• We are studying software, not people.
• We have a comprehensive consent process in place.
...Wait a minute

- Our study is deemed to be non-human-subject research by UC Berkeley’s IRB.
- We collect no private information of traffic (encrypted or unencrypted).
- All web browser traffic is excluded.
- We are studying software, not people.
- We have a comprehensive consent process in place.

Lumen is a tool that helps mobile users to understand their app traffic. With Lumen, we can observe the traffic in the wild by snoopers and adversary platforms. We would love to create more privacy and security for mobile users.

Lumen is brought to you by the International Computer Science Institute, USA and IMDEA Networks, Spain, part of an academic NSF and Data Transport.

Install Root Certificate

If you want Lumen to be able to analyze encrypted traffic, you need to install its root certificate. Please, select install to proceed.

Cancel    Install
...Wait a minute

- Our study is deemed to be non-human-subject research by UC Berkeley's IRB.
- We collect no private information (encrypted or unencrypted).
- All web browser traffic is excluded.
- We are studying software, not people.
- We have a comprehensive consent process in place.
What do we collect?

- Three key items:
  - Client Hello
  - Server Hello (with certificates)
  - Failures of our TLS proxy (reveals pinning)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Users</td>
<td>&gt;5,000 from &gt;100 countries</td>
</tr>
<tr>
<td>Connections (11/15—6/17)</td>
<td>1,486,082</td>
</tr>
<tr>
<td>Apps</td>
<td>7,258</td>
</tr>
<tr>
<td>Domains (unique SNIs)</td>
<td>34,176</td>
</tr>
<tr>
<td>TCP ports</td>
<td>250</td>
</tr>
<tr>
<td>Unique device/OS combos</td>
<td>891</td>
</tr>
<tr>
<td>TLS proxy failures</td>
<td>684,209 (4,268 apps and 10,753 domains)</td>
</tr>
</tbody>
</table>
TLS Library Usage

• 84% of application versions in our dataset use OS-default libraries with default settings
Why do Apps not use defaults?

• To improve security:
  • Facebook uses OpenSSL and removes weaker cipher suites from the list (e.g. RC4 and 3DES ciphers); it also uses Facebook-specific ALPN
  • Twitter uses OS-provided libraries with a reordered cipher suite list

• Some do it wrong:
  • Some private messaging and VoIP apps use their own short cipher suite lists that do not have any forward-secret ciphers

• Others use third party libraries instead of the default:
  • Firefox uses NSS, VLC & SoundCloud GnuTLS (some versions)
Weak/outdated primitives

- **SSLv3:**
  - Supported by any app running on Android 5.0 and below (more than 61% of phones)
  - EA Games apps (FIFA Mobile, Madden NFL Mobile, etc.) with 100s of millions of installs, even when running on versions of Android that do not support it by default

- Null and Anonymous ciphers
  - Apps like TuneIn Radio with hundreds of millions of installs
  - Multiple EA games

- Export-grade ciphers:
  - Android 4.0 and below
  - Tiffany Alvord Dream World, a children’s game that has over one million installs

- Most apps with weak ciphers use poorly-configured OpenSSL
Solutions?

• De-couple TLS updates from OS updates!

  • TLS should be able to updated independent of the rest of the firmware, making it easier to update without manufacturer/vendor cooperation

  • Google is already doing this with Google Play Services (which bundle their own TLS library and certificate stores), so why not do the same with the OS-provided TLS library?

• Give more configuration options to developers

  • This way apps that need extra configuration options (e.g. setting ALPNs) are not forced to use something else
Certificates and Trust

- Android root stores often have “impurities” [Vallina-Rodriguez et al.]
- Some apps do not trust these trust stores and bundle their own CA certificates, pin server certificates, or use self-signed certificates
  - E.g. Firefox (bundles CA cert. store), Uber, Google, Paypal, Facebook (certificate pinning), Yandex (bundles unofficial Yandex root CA), Samsung apps (self-signed certs.) etc.
- Implemented poorly, these can open up apps to MITM attacks
What does this all mean?

- Most apps (98%) trust OS-provided CA stores, and are vulnerable to MITM attacks when those are polluted.

- Some apps pin certificates to mitigate the problem of polluted CA stores.
  - It is not very prevalent (less than 2% of apps).
  - This can be problematic when done poorly: major system recovery app (with root access and ability to flash system firmware, bootloader, recovery, etc.) downloads CA bundle from the cloud in the clear.

```
Outbound connection contents for 47824->52.84.245.50-80(dns:d2to8[.]cloudfront.net)(app:)
GET /truststores/20041/truststore.zip HTTP/1.1
If-Modified-Since: Tue, 21 Mar 2017 15:32:28 GMT+00:00
Connection: close
User-Agent: Dalvik/2.1.0 (Linux; U; Android 6.0.1; Nexus 5 Build/M4B30Z)
Host: d2to8[.]cloudfront.net
Accept-Encoding: gzip
```
How do we fix it?

- What do we do with all the polluted CA certificate stores?
  - Google needs to ensure (e.g. through Android’s licensing terms) that vendors can not surreptitiously inject their own CA certificates in trust stores
  - CA certificates also need to be able to be updated independently

- But some will still use their own libraries and pin certificates…
  - Make sure developers are properly educated about TLS
  - Detect and prevent poor implementations
  - Google has done something similar in the past: they implemented a tool that prevented developers from uploading apps that used a vulnerable version of GnuTLS and informed them about the issue
Summary

• First study of TLS usage in Android apps at scale
• Majority of apps (84%) use OS-provided libraries with default settings
• Apps using OS-defaults are vulnerable when the OS is outdated
• Apps using 3rd-party libraries and configurations are prone to misconfiguration and are therefore vulnerable
• Found low use of certificate pinning and CA bundling (less than 2%)
• Provided insights and potential solutions to the problems we found