INTERNATIONAL CRYPTOGRAPHIC MODULE CONFERENCE 2017

May 16-19 | Westin Arlington Gateway | Washington, DC

# TLS Panel Discussion


Tim Hudson – Cryptsoft –(Moderator)

Steve Marquess – OpenSSL

David Hook – Bouncy Castle

Kenn White – Open Crypto Audit

Nicko van Someren – Linux Foundation




ICMC17

# Session Overview

- TLS in an Open Source Context

# Discussion Topics

# Discussion Topics - General

- How many TLS implementations are there?
  - Why are there so many (or few)?
  - What motivates creation (or destruction) of implementations?
  - Is this a good (or bad) thing?
  - Do we need more (or fewer) implementations?

# Discussion Topics – (Background Info)

| Implementation | Developed by | Open source | Written in | Origin |
|---|---|---|---|---|
| Botan | Jack Lloyd | Yes | C++ | US (Vermont) |
| BoringSSL | OpenSSL project | Yes | C, assembly | Australia/EU |
| Bouncy Castle | The Legion of the Bouncy Castle Inc. | Yes | Java, C# | Australia |
| cryptlib | Peter Gutmann | Yes | C | NZ |
| GnuTLS | GnuTLS project | Yes | C | EU (Greece and Sweden) |
| Java Secure Socket Extension (JSSE) | Oracle | Yes | Java | US |
| LibreSSL | OpenBSD Project | Yes | C, assembly | Canada |
| MatrixSSL[7] | PeerSec Networks | Yes | C | US |
| mbed TLS (previously PolarSSL) | ARM | Yes | C | EU (Netherlands) |
| Network Security Services (NSS) | Mozilla, AOL, Red Hat, Sun, Oracle, Google and others | Yes | C, assembly | US |
| OpenSSL | OpenSSL project | Yes | C, assembly | Australia/EU |
| S2n | Amazon | Yes | C | US |
| SChannel | Microsoft | No | | US |
| Secure Transport | Apple Inc. | Yes | | US |
| wolfSSL (previously CyaSSL) | wolfSSL[14] | Yes | C | US |

https://en.wikipedia.org/wiki/Comparison_of_TLS_implementations

# Discussion Topics - General

- What is the typical development time of a TLS implementation?
  - Why so long (so short) a time period?
  - How can we make it faster (or slower)?
  - Is it fundamentally different in a commercial closed source context?

# Discussion Topics – (Background Info)

| name | language | role(s) | version | features/limitations |
|---|---|---|---|---|
| NSS | C | C/S | -18 | Almost everything, except post-handshake auth and key update |
| Mint | Go | C/S | -18 | PSK resumption, 0-RTT, HRR |
| nqsb | OCaml | C/S | -11 | PSK/DHE-PSK, no EC*, no client auth, no 0RTT -- live server at tls13test.nqsb.io port 4433, records traces, ping @hannesm, contains a static PSK/DHE_PSK token: id: 0x0000 secret: 0x000102030405060708090a0b0c0d0e0f101112131415161718191a1b1c1d1e1f |
| ProtoTLS | JavaScript | C/S | -13 | EC/DHE/PSK, no HelloRetryRequest |
| miTLS | F* | C/S | -13 | EC/DHE/PSK, no HelloRetryRequest |
| Tris | Go | S | -18 | ECDHE/PSK/0-RTT, no HelloRetryRequest |
| BoringSSL | C | C/S | -18 | P-256, X25519, HelloRetryRequest, resumption |
| Wireshark | C | other | -20 | Full decryption and -19 dissection support since v2.3.0rc0-2802-gf6e04681fc (keylog format proposal), -20 since v2.3.0rc0-3395-g0807e50f69. Missing ChaCha20-Poly1305 decryption, missing 0RTT trial decryption. Tracking bug |
| picotls | C | C/S | -18 | P-256, X25519, HelloRetryRequest, resumption, 0-RTT |
| rustls | Rust | C/S | -20 | P-256/P-384/curve25519, HRR, resumption |
| Haskell tls | Haskell | C/S | -19 | (EC)DHE w/ P* and X*, full, HRR, PSK, 0RTT |
| Leto | C# | S | -18 | DHE, X25519, AES, no PSK no 0RTT. Tested against NSS |
| OpenSSL | C | C/S | -20 | P-256, X25519, HelloRetryRequest, resumption, 0-RTT |
| wolfSSL | C | C/S | -18 | P-256, P-384, HelloRetryRequest, resumption, KeyUpdate |

https://github.com/tlswg/tls13-spec/wiki/Implementations

# Discussion Topics - Standards

- What level of participation do those working on open source TLS implementations have on the actual TLS specification development process?
  - Why so much (so little) participation?
  - Is this is a good (or bad) thing?

# Discussion Topics – Implementation Issues

- What level of cooperation is there between the open source TLS implementations?
  - Between Forks?
  - Across entirely different code bases?
  - Why so much (so little) cooperation?
  - Is this is a good (bad) thing?

# Discussion Topics – Implementation Issues

- Are there common implementation issues?
  - Between Forks?
  - Across entirely different code bases?
  - What can we learn from this?
  - Should we be addressing this in a more holistic manner?

# Discussion Topics – Defects

- What happens with protocol defects?
  - Between Forks?
  - Across entirely different code bases?
  - How quickly (or slowly) are protocol issues fixed?
  - What can we learn from this?

# Discussion Topics – Open/Closed Source

- Is it fundamentally different in a commercial closed source context?
  - Is development longer (or shorter)?
  - Are there fewer (or more) defects?
  - How can we make it faster (or slower)?

# Discussion Topics – Open/Closed Source

- What benefits are there in an open source context?
  - Development?
  - Support?
  - Defect remediation?
  - Documentation?
  - Communication?

# Discussion Topics – Open/Closed Source

- What benefits are there for **validation** in an open source context?
  - Is it harder (or easier)?
  - Is it faster (or slower)?
  - Is it cheaper (or more expensive)?
  - Is it more (or less) fun?

# Discussion Topics – Collaboration

- Is there enough collaboration?
  - Do we have people contributing across more than one project?
    - How frequent (or infrequent) is this?
  - Is that across languages (or just forks)?

# Discussion Topics – Contribution

- Is there enough contribution?
  - Are those benefiting from open source contributing?
  - How much commercial feedback occurs?

# Questions

Tim Hudson – Cryptsoft – (Moderator)

Steve Marquess – OpenSSL

David Hook – Bouncy Castle

Kenn White – Open Crypto Audit

Nicko van Someren – Linux Foundation

T

# OpenSSL

Cryptography and SSL/TLS Toolkit