TLS 1.3 CIP Security Impacts

Jack Visoky, Joakim Wiberg, Nancy Cam-Winget
Rockwell Automation, HMS Industrial Networks, Cisco Systems

October 10, 2018
• CIP Security is built on TLS and DTLS for authentication and data confidentiality/data authenticity
• When the CIP Security specification was released in November 2015, the most current version of TLS was TLS 1.2
• In May 2018 TLS 1.3 was approved by the IETF
• We will explore some of the changes here and the impacts to CIP Security
TLS 1.3: What’s New?

• A few main changes with TLS 1.3
  – Improved privacy with perfect forward secrecy
  – Separation of authentication and key management within a handshake
  – Optimization of the handshake for improved performance

• Note that TLS 1.3 builds on TLS 1.2
  – This is not a complete re-write of TLS, but rather an evolution
Handshake Changes

- TLS 1.3 makes several changes to the handshake mechanism
- Authentication and key establishment have been decoupled
  - Previously in TLS 1.2 this was linked; a party was authenticated by virtue of successful key establishment
  - For TLS 1.3, the decoupling means that any combination of algorithms for these two functions can be used
**Handshake Changes – Session Renegotiation**

- **Session renegotiation is no longer supported in TLS 1.3**
  - This is something that the CIP Security specification talked about specifically
  - It was an optional feature to support sequence count rollover cases
- **KeyUpdate message is supported**
  - This can be used to refresh session keys, but without full renegotiation
  - This can and should be used by CIP Security if/when adopting TLS 1.3
Handshake Changes – 0-RTT

- Zero Round Trip Time (0-RTT) handshakes are potentially the biggest and most impactful change in TLS 1.3
  - When a client and server first establish a TLS connection, they can optionally share a key (referred to as a Pre-Shared Key, or PSK)
  - The next time this client and server connect, the PSK can be used to transmit application data immediately (this is referred to as “Early Data”)
  - In this way application data is sent immediately, without any handshaking
0-RTT vs Full Handshake (TLS 1.2)

**TLS 1.3 0-RTT**

- ClientHello
  + early_data
  + psk_key_exchange_modes
  + key_share

- ApplicationData

- ServerHello
  + pre_shared_key
  + key_share

- EncryptedExtensions
  + early_data

- Finished

- ApplicationData

- EndOfEarlyData

- Finished

- ApplicationData

**TLS 1.2 Full Handshake**

- ClientHello

- ServerHello

- Certificate

- ServerKeyExchange

- CertificateRequest

- ServerHelloDone

- Certificate

- ClientKeyExchange

- CertificateVerify

- ChangeCipherSpec

- Finished

- ChangeCipherSpec

- Finished

- ApplicationData

**Notes:**
- Indicates optional or situation-dependent messages/extension that are not always sent.
- Indicates messages protected using keys derived from client_early_traffic_secret.
- Indicates messages protected using keys derived from [sender]_handshake_traffic_secret.
- Indicates messages protected using keys derived from [sender]_application_traffic_secret_N.
0-RTT Risks

• Although 0-RTT significantly increases the performance of the TLS handshake, it is not without risk
  – Early Data is protected just with the PSK
  – Vulnerable to a replay attack
  – RFC strongly suggests that any early data sent be idempotent, meaning it does not affect the state of the client or server
    • Example would be a read request
  – For HTTP, much of the data sent at the beginning of a connection is idempotent
  – The same cannot be said of CIP
    • It could be a read attribute, or it could be a set attribute or a service which changes state
0-RTT Risks

- Note that 0-RTT is optional, both client and server must agree to using this.
- Since this is optional, the user should be able to configure whether or not to leverage 0-RTT.
  - User should understand the risk and benefit of this.
  - For some CIP connections it might make sense, like listen-only I/O.
  - Others the risk is likely too great.
- Note that CIP Security with TLS 1.2 already supports PSKs through an attribute.
  - This could be extended to support 0-RTT.
Cipher Suites

- List of supported cipher suites cut down significantly
- All of these primitives for cryptography are no longer supported:
  - SHA-1 Hash Function
  - RC4 Steam Cipher
  - DES
  - 3DES
  - AES-CBC
  - MD5 Algorithm
  - Various Diffie-Hellman groups
  - EXPORT-strength ciphers
  - RSA Key Transport
- Supported Cipher suites are:

<table>
<thead>
<tr>
<th>Encryption Cipher</th>
<th>HMAC for key derivation</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>AES-128 GCM</td>
<td>SHA-256</td>
<td>Mandatory</td>
</tr>
<tr>
<td>AES-256 GCM</td>
<td>SHA-384</td>
<td>Optional</td>
</tr>
<tr>
<td>CHACHA20 Poly1305</td>
<td>SHA-256</td>
<td>Optional</td>
</tr>
<tr>
<td>AES-128 CCM</td>
<td>SHA-256</td>
<td>Optional</td>
</tr>
<tr>
<td>AES-128-8</td>
<td>SHA-256</td>
<td>Optional</td>
</tr>
</tbody>
</table>
Cipher Suites – No Null Encryption

- One area of concern for CIP Security is that all of the TLS 1.3 cipher suites include confidentiality
  - For CIP Security and TLS 1.2, Authentication-only (also termed NULL Encryption) cipher suites were important
  - This allows for data authenticity without data confidentiality
  - Useful for:
    - Performance improvements of when confidentiality is not needed (e.g. CIP Motion)
    - Inspection and debugging to ensure system is operating properly
- IETF had a general philosophy to simplify TLS with TLS 1.3, hence removal of these cipher suites which are not often used within Internet communications
Bringing Authentication-only Cipher Suites to TLS 1.3

- There is an effort to include Authentication-only cipher suites within TLS 1.3
  - Authored by Nancy Cam-Winget and Jack Visoky
- This draft RFC has been brought to the TLS 1.3 Working Group
  - Discussion took place within the mailing list
  - Received many comments, currently updating the draft based on the IETF discussion
• There are several extensions defined for TLS 1.3
• A few potentially useful ones for CIP Security are listed below
• Support could be added to control these extensions explicitly through CIP Security objects

<table>
<thead>
<tr>
<th>Extension</th>
<th>Applicability to CIP Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supported Versions</td>
<td>Helpful to know what version of TLS an endpoint is supporting/using</td>
</tr>
<tr>
<td>Cookie</td>
<td>Can be used to prevent some DoS attacks, configuring this could be supported through a CIP attribute</td>
</tr>
<tr>
<td>Signature Algorithms</td>
<td>Closely related to the Allowed Cipher Suites attribute; this attribute could be extended to allow/disallow certain signature algorithms</td>
</tr>
</tbody>
</table>
TLS 1.3 Extensions (continued)

- A few more potentially interesting extensions

<table>
<thead>
<tr>
<th>Extension</th>
<th>Applicability to CIP Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negotiated Groups</td>
<td>Also similar to the existing Allowed Cipher Suites Attribute, this could be enhanced to support various negotiated groups</td>
</tr>
<tr>
<td>Server Name Indicator</td>
<td>Could be useful to support some CIP name here which could link the TLS security with the application layer naming/addressing</td>
</tr>
<tr>
<td>Certificate Authorities</td>
<td>Optimize handshake, potentially support multiple identities on the target</td>
</tr>
</tbody>
</table>
Conclusions

- TLS 1.3 brings many benefits, although there is some risk with the lack of Authentication-only/NULL Encryption support
- Recommended to enhance CIP Security to support TLS 1.3
  - Within 3 years if NULL encryption is included in TLS 1.3
  - Within 6 years if not
- TLS 1.2 will need to continue to be supported for several years
  - User should be able to configure whether TLS 1.3 or TLS 1.2 or both are used within an endpoint
- There are several attribute changes suggested here to support TLS 1.3 new features (such as 0-RTT), it is suggested these be implemented within the CIP Specification for optimal support
THANK YOU