TLS Secured Connections in CANoe

Practical Approaches to Analyzing Data That Are Not Intended For Analysis
Agenda

- **Introduction**
  - Software Development Impact
  - End-Point Simulation
  - Not End-Point Simulation
  - Conclusion
TLS stands for Transport Layer Security

It secures the TCP communication above the Transport Layer (ISO/OSI Model Layer 4)
- It is a pure point-to-point communication, broadcast is not possible

Typically TLS allows Authentication of the Server and the Client and encrypts the payload content
- The connection can still be traced

Well known application domains are HTTPs and
- SmartCharge communication in automotive domains
How Does TLS Work?

**Introduction**

Alice

Certification

Client Hello (Supported ciphers, random number C and session ID)

Server Hello (chosen cipher, random number S, session ID)

Server certificate
How Does TLS Work?

Introduction

Client Hello (Supported ciphers, random number C and session ID)
Server Hello (chosen cipher, random number S, session ID)

Certificate

Client key exchange (encrypted pre-master secret)

Pre-master secret

Pre-master secret
How Does TLS Work?

Introduction

Client Hello (Supported ciphers, random number C and session ID)

Server Hello (chosen cipher, random number S, session ID)

Server certificate

Client key exchange (encrypted pre-master secret)

Pre-master secret

Generate key

Pre-master secret

Generate key

Transmit/receive encrypted data
Agenda

Introduction

- **Software Development Impact**
  - End-Point Simulation
  - Not End-Point Simulation
  - Conclusion
Consequence of Security for the Development Environment

- Handling of security material like keys and certificates
  - This has practically the highest impact, but out of scope for this presentation
  - Tools like the Security Manager of Vector may help to overcome certain issues while development phase
- Unable to observe communication
Example: TLS secured smart charge communication
# Trace-Fenster

## Software Development Impact

### Example: TLS secured smart charge communication

<table>
<thead>
<tr>
<th>Time</th>
<th>Protocol</th>
<th>Name</th>
<th>Protocol Info</th>
<th>Source IP</th>
<th>Destination IP</th>
<th>Packet Length</th>
<th>Payload</th>
</tr>
</thead>
<tbody>
<tr>
<td>10:03:4036</td>
<td>sdp</td>
<td>SECC Discovery Request</td>
<td>TP version: 1</td>
<td>FE80::2</td>
<td>FF02::1</td>
<td>10</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TLS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transport Layer</td>
<td>TCP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:03:4043</td>
<td>ndp</td>
<td>Neighbor Solicitation</td>
<td></td>
<td>FE80::11</td>
<td>FF02::1:FF00::2</td>
<td>86</td>
<td>0</td>
</tr>
<tr>
<td>10:03:4049</td>
<td>ndp</td>
<td>Neighbor Advertisement</td>
<td></td>
<td>FE80::2</td>
<td>FE80::11</td>
<td>86</td>
<td>0</td>
</tr>
<tr>
<td>10:03:4057</td>
<td>sdp</td>
<td>SECC Discovery Response</td>
<td>TP version: 1</td>
<td>FE80::11</td>
<td>FE80::2</td>
<td>28</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TLS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Transport Layer</td>
<td>TCP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>EVSE IP</td>
<td>FE80::0:0:0:0:0:11</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>EVSE TCP Port</td>
<td>51111</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10:03:4063</td>
<td>tcp</td>
<td>C866 -&gt; C7A7 [Syn] Seq=32FC4660 Win=FFFF</td>
<td>FE80::2</td>
<td>FE80::11</td>
<td>78</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>10:03:4069</td>
<td>tcp</td>
<td>C7A7 -&gt; CF96 [ACK, SYN] Seq=143D3060 Ack=32FC4661 Win=FFFF</td>
<td>FE80::2</td>
<td>FE80::11</td>
<td>78</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>10:03:4075</td>
<td>tcp</td>
<td>CF96 -&gt; C7A7 [ACK] Seq=33FC4661 Ack=143D5C0E Win=FFFF</td>
<td>FE80::2</td>
<td>FE80::11</td>
<td>74</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>10:03:4113</td>
<td>tls</td>
<td>TLS 1.0: Handshake (Client Hello)</td>
<td>FE80::2</td>
<td>FE80::11</td>
<td>472</td>
<td>398</td>
<td></td>
</tr>
<tr>
<td>10:03:4126</td>
<td>tls</td>
<td>TLS 1.2: Handshake (Server Hello)</td>
<td>FE80::2</td>
<td>FE80::11</td>
<td>170</td>
<td>96</td>
<td></td>
</tr>
<tr>
<td>10:03:4248</td>
<td>tls</td>
<td>TLS 1.2: Handshake</td>
<td>FE80::2</td>
<td>FE80::11</td>
<td>1514</td>
<td>1440</td>
<td></td>
</tr>
<tr>
<td>10:03:4253</td>
<td>tcp</td>
<td>C8F6 -&gt; C7A7 [ACK] Seq=33FC47EF Ack=143DBFDE Win=FFFF</td>
<td>FE80::2</td>
<td>FE80::11</td>
<td>74</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>10:03:4323</td>
<td>tls</td>
<td>TLS 46.68: 51111</td>
<td>FE80::11</td>
<td>FE80::30</td>
<td>870</td>
<td>796</td>
<td></td>
</tr>
<tr>
<td>10:03:4340</td>
<td>tls</td>
<td>TLS 1.2: Handshake (Client Key Exchange)</td>
<td>FE80::2</td>
<td>FE80::11</td>
<td>217</td>
<td>143</td>
<td></td>
</tr>
<tr>
<td>10:03:5006</td>
<td>tcp</td>
<td>C7A7 -&gt; CF96 [ACK] Seq=143DEEA Ack=32FC487E Win=FFFF</td>
<td>FE80::11</td>
<td>FE80::2</td>
<td>74</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>10:03:50015</td>
<td>tls</td>
<td>TLS 1.2: Change Cipher Spec, Handshake</td>
<td>FE80::2</td>
<td>FE80::11</td>
<td>117</td>
<td>43</td>
<td></td>
</tr>
<tr>
<td>10:03:50022</td>
<td>tls</td>
<td>TLS 1.2: Change Cipher Spec</td>
<td>FE80::2</td>
<td>FE80::11</td>
<td>80</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>10:03:60006</td>
<td>tcp</td>
<td>CF96 -&gt; C7A7 [ACK] Sec=33FC48A9 Ack=143DEDF0 Win=FFFF</td>
<td>FE80::2</td>
<td>FE80::11</td>
<td>74</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>10:03:60015</td>
<td>tls</td>
<td>TLS 1.2: Handshake</td>
<td>FE80::2</td>
<td>FE80::11</td>
<td>111</td>
<td>37</td>
<td></td>
</tr>
<tr>
<td>10:03:90028</td>
<td>tls</td>
<td>TLS 1.2: Application Data</td>
<td>FE80::2</td>
<td>FE80::11</td>
<td>74</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>11:03:0006</td>
<td>tcp</td>
<td>C7A7 -&gt; CF96 [ACK] Seq=143DF15 Ack=32FC490A Win=FFFF</td>
<td>FE80::11</td>
<td>FE80::2</td>
<td>74</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>11:03:20037</td>
<td>tls</td>
<td>TLS 1.2: Application Data</td>
<td>FE80::2</td>
<td>FE80::11</td>
<td>107</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>11:03:30006</td>
<td>tcp</td>
<td>CF96 -&gt; C7A7 [ACK] Seq=33FC490A Ack=143DF36 Win=FFFF</td>
<td>FE80::2</td>
<td>FE80::11</td>
<td>74</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
Agenda

Introduction
Software Development Impact

- **End-Point Simulation**
  - Not End-Point Simulation
  - Conclusion
CANoe supports build in TLS layer for simulation
- Simply call function `tlsOpen(socket)` at an existing TCP socket
- Configuration of TLS is supported by means of Security Manager profile
End-Point Simulation

Analysis

CANoe

2000 Bytes

TLS
TCP
IP
Ethernet

2000 Bytes

TLS
TCP
IP
Ethernet

CANoe Protocol Observer

Trace

Ethernet Paket #1
Ethernet Paket #n-1
Ethernet Paket #n
End-Point Simulation

TLS Trace-Window Support

- Smart charge communication TLS secured but enhanced interpretation (hence CANoe 12)
End-Point Simulation

Friendly Reminder on Our Objective

- Alice
- Bob
- TOOL
Playing The Role of Alice And Bob

End-Point Simulation
End-Point Simulation

Playing The Role of Alice And Bob

Pro

▶ Issues on the application layer can be analyzed due to availability of decrypted messages
▶ Even messages/signals can be manipulated
▶ Security layer on real ECU does not need any modification
  ▶ Works only if private certificates are shared with the tool
    ▶ No private development certificates might be used, too
▶ Logging for post data analysis is less complicated

Con

▶ Communication layer is completely replaced by simulation
  ▶ Issues on the communication layer may vanish when using the tool
  ▶ Impact on timing and reaction time
▶ Also non TLS related communication is routed through the proxy
  ▶ Hardware bypass might be an option for some non IP traffic
  ▶ Partially routing of IP and related traffic may confuse the TCP/IP stack
    ▶ Issues if Bob' and Bob are connected to the same network and run in parallel
    ▶ E.g. DHCP, ARP
Agenda

Introduction
Software Development Impact
End-Point Simulation

► **Not End-Point Simulation**

Conclusion
We Are Not an End-Point, Now What?

- Observing methods
  - Null-cipher → result of the encryption is the origin content
  - Weak cipher → allow real-time hacking/decryption
- You may think about modifying Alice and/or Bob to
  - disable encryption at all
  - manipulate the crypto algorithm → weaker cryptography to allow real-time hacking
  - provide a proprietary side channel for the tool → e.g. share master secret, use temporary keys
- Any modification you apply may weaken your cryptography
  - If such “debug” functionality can be enabled in the released product it may become an attack point for hackers, too
  - If the modification will be removed before release, potentially all performed tests are to be repeated
Consider to Use Standards But Weaker Security Mechanisms

Client Hello (Supported ciphers, random number C and session ID)

Server Hello (chosen cipher, random number S, session ID)

Server certificate

Transmit/receive encrypted data
Agenda

Introduction
Software Development Impact
End-Point Simulation
Not End-Point Simulation

► Conclusion
Security in The Development Environment

- Taking into account all discussed aspects, there is no single solution that fits all requirements
- The tools will not solve the problems. Tools can help to address and overcome individual issues
  - e.g. Security Manager can help to handle sensitive materials like certificates, keys to ease the distribution and make them available to users who are not trained to handle sensitive crypto material
- Make sure your measures during the development phase will not have a negative impact on the final product
  - Don’t leave backdoors open, revert any modification permanently and use temporary keys if possible
- Since most likely your security measures can be considered somehow proprietary don’t forget to discuss this with your preferred tools supplier

Roadmap:
- CANoe 11.0 SP3 \(\rightarrow\) supports TLS Stack
- CANoe 12.0 \(\rightarrow\) supports “decryption” of TLS encrypted packets
- CANoe 12.0 SP \(\rightarrow\) will support pre-shared keys
- An easy to use Man-in-the-middle (proxy) is being conceived for a future release
Your questions are welcome!

Author: Fellmeth, Peter
Vector Germany