COMPROTware:Testtool - Introduction to TLS-secured connections

Introduction and examples for TLS-secured connections

First, simple test with CPTT

Test

- Get the IPv4 address for www.google.de
  - Linux: ping -4 www.google.de
  - MS Windows: ping.exe -4 www.google.de
- Start CPTT. Change to protocol family "Raw data", to protocol "Raw date net"
- Set the IPv4 address to www.google.de (September, 2021: 142.251.36.195).
- Set Transport layer to TCP/IP, set Port no. 443
- Enable "Use TLS-secured connection".
- Disable "Verify peer certificate".
- All other fields in section "TLS-secured connection" should be empty.
- Press Accept.
- Prepare a HTTP request:
  - In Main View, press right mouse button, select "Send Message ..."
  - Set Hexstring to ""GET /index.html HTTP/1.1" 0x0a 0x0a" (the innermost quotes are part of the Hexstring).
- Establish a connection: Action, Simulate Master.
- In window "Send Message": Press "Send".

```
     14:02:41.041
         Raw data net protocol profile:
           Max.Frame length = 15
           Delay after client connection estab. failed = 30sec
           TLS-secured connection enabled
             Local host leaf certificate - certificate file: <not specified>
             Local host leaf certificate - private key file: <not specified>
             Verify remote host leaf certificate: no
             Remote host leaf cert. - certificate chain file: <not specified>
             Remote host - Common name (CN): <not specified>
           Delay after connection closed = 10sec
           Log communication errors = YES
     14:02:41.043
         Trying to connect to Slave at TCP:142.251.36.195:443 ...
     14:02:42.015
         Connection established to remote 142.251.36.195:443 via 127.0.0.1:60367
             TLS-secured connection: TLSv1.2, TLS-ECDHE-RSA-WITH-AES-128-GCM-SHA256
M>>  14:02:44.216
M>>    0x47 0x45 0x54 0x20  0x2f 0x69 0x6e 0x64  0x65 0x78 0x2e 0x68  0x74 0x6d 0x6c 0x20    | GET /index.html
M>>    0x48 0x54 0x54 0x50  0x2f 0x31 0x2e 0x31                                              | HTTP/1.1
```

Explanation

- A TLSv1.2 connection is established.
- Used Cipher: TLS-ECDHE-RSA-WITH-AES-128-GCM-SHA256
- For the CPTT side (this is the Master resp. client side) neither certificate nor a private key is specified.
- Google has a certificate. See Help, Support, View Support Info:

```
 CPLBOsCommMbedtlsSslHandshakeTest():   Peer certificate info:
     cert. version     : 3
     serial number     : 9C:70:86:A1:03:E0:22:73:0A:00:00:00:00:FC:F9:25
     issuer name       : C=US, O=Google Trust Services LLC, CN=GTS CA 1C3
     subject name      : CN=google.com
     issued  on        : 2021-08-30 04:14:41
     expires on        : 2021-11-22 04:14:40
     signed using      : RSA with SHA-256
     RSA key size      : 2048 bits
     basic constraints : CA=false
     subject alt name  : google.com, *.appengine.google.com, *.bdn.dev, *.cloud.google.com, *.crowdsource.google.com,
*.datacompute.google.com, *.google ...
```

- If "Verify peer certificate" is enabled, connection establishment will fail because we do not have the Certificate chain file to verify Google’s certificate.
- For TLS-secured connections to a web server, the client (CPTT side) never needs a certificate or a key. If you specify in CPTT a local host certificate and a local host key, these information is ignored by the web server.

CPTT to CPTT TLS-secured connection

Test

- Start CPTT. Change to protocol family "IEC 60870-5" and protocol "IEC 60870-5-104".
- Set Controlled Station IP address to 127.0.0.1 (is localhost), port no. to 19998 (is standard port no for TLS-secured IEC 60870-5-104).
- Enable "Use TLS-secured connection".
- Set "Leaf certificate certificate file" to C:\Program Files (x86)\Real Thoughts GmbH\COMPROTware\Testtool\doc\2.22\Demo_Certificate\ReaTho-Demo-For_CPTT_1-Partner_A-leaf_cert-cert.pem
- Set "Leaf certificate private key file" to C:\Program Files (x86)\Real Thoughts GmbH\COMPROTware\Testtool\doc\2.22\Demo_Certificate\ReaTho-Demo-For_CPTT_1-Partner_A-leaf_cert-privkey.pem
- Disable "Verify peer certificate".
- Clear "Certificate chain file" and "Common name (CN)".
- Press Accept.

- Save the current configuration: File, Save Configuration as user default.
- Start a second CPTT instance (it will overlap with the original instance, move the new window to another position).
- One CPTT instance: Action, Simulate Controlled Station.
- The other CPTT instance: Action, Simulate Controlling Station.
- A TLS-secured connection should be established, IEC 60870-5-104 Frames should be transmitted.
- Controlling Station log should look alike:

```
     13:34:33.237
         IEC 60870-5-104 protocol profile:
           Link Layer:
             Max.Frame length net = 15, gross = 17
             Timeout: t0 = 30sec, t1 = 15sec, t2 = 10sec, t3 = 30sec
             Parameter: k = 12, w = 8
             Delay after connection closed = 10sec
             Log communication errors = YES
           Delay after client connection estab. failed = 30sec
           TLS-secured connection enabled
             Local host leaf certificate - certificate file: "ReaTho-Demo-For_CPTT_1-Partner_A-leaf_cert-cert.pem"
             Local host leaf certificate - private key file: "ReaTho-Demo-For_CPTT_1-Partner_A-leaf_cert-privkey.pem"
             Verify remote host leaf certificate: no
             Remote host leaf cert. - certificate chain file: <not specified>
             Remote host - Common name (CN): <not specified>
     13:34:33.261
         Trying to connect to Controlled Station at TCP:127.0.0.1:19998 ...
     13:34:33.530
         Connection established to remote 127.0.0.1:19998 via 127.0.0.1:60067
             TLS-secured connection: TLSv1.2, TLS-ECDHE-RSA-WITH-AES-256-GCM-SHA384
C>>  13:34:33.530
C>>      U: STARTDT act
<<D  13:34:33.775
<<D      U: STARTDT con
```

- Controlled station log should look alike:

```
     13:34:32.478
         IEC 60870-5-104 protocol profile:
           Application Layer:
             Invoke qualified Message List to respond = no
           Link Layer:
             Timeout: t0 = 30sec, t1 = 15sec, t2 = 10sec, t3 = 30sec
             Parameter: k = 12, w = 8
           TLS-secured connection enabled
             Local host leaf certificate - certificate file: "ReaTho-Demo-For_CPTT_1-Partner_A-leaf_cert-cert.pem"
             Local host leaf certificate - private key file: "ReaTho-Demo-For_CPTT_1-Partner_A-leaf_cert-privkey.pem"
             Verify remote host leaf certificate: no
             Remote host leaf cert. - certificate chain file: <not specified>
             Remote host - Common name (CN): <not specified>
     13:34:32.529
         Waiting for connection from Controlling Station to TCP:0.0.0.0:19998 ...
     13:34:33.508
         Connection accepted from remote 127.0.0.1:60067 to 127.0.0.1:19998
             TLS-secured connection: TLSv1.2, TLS-ECDHE-RSA-WITH-AES-256-GCM-SHA384
C>>  13:34:33.774
C>>      U: STARTDT act
<<D  13:34:33.775
<<D      U: STARTDT con
```

Explanation

- In this example for both client and server side (Controlling Station and Controlled Station side) a certificate and a private key is specified. Untypically both have the same certificate and private key.
- Typically only the client will request and check the certificate.
- The CPTT instance which simulated the client will show the certificate returned by the server (See Help, Support, View Support Info):

```
: CPLBOsCommMbedtlsSslHandshakeTest():   Peer certificate info:
:     cert. version     : 1
:     serial number     : A9:F4:76:02:55:8B:E2:A2
:     issuer name       : C=DE, ST=BaWue, L=Karlsruhe, O=Real Thoughts GmbH, OU=Sales, CN=Real Thoughts GmbH (COMPROTware demo),
emailAddress=info@realth ...
:     subject name      : C=DE, ST=BaWue, L=Karlsruhe, O=Real Thoughts GmbH, OU=Sales, CN=Real Thoughts GmbH (COMPROTware demo),
emailAddress=info@realth ...
:     issued  on        : 2019-02-03 07:07:56
:     expires on        : 2024-02-03 07:07:56
:     signed using      : RSA with SHA-512
:     RSA key size      : 2048 bits
```

- In the profile of the CPTT instance which simulated the client you can clear the fields "Leaf certificate certificate file" and "Leaf certificate private key file". This will not change the behavior.

Advanced CPTT to CPTT TLS-secured connection

Test

- IN BOTH CPTT INSTANCES, DO THE FOLLOWING:
- Use profile from previous test.
- Enabled "Verify peer certificate".
- Set "Certificate chain file" to C:\Program Files (x86)\Real Thoughts GmbH\COMPROTware\Testtool\doc\2.22\Demo_Certificate\ReaTho-Demo-For_CPTT_2-Partner_A-cert_chain-cert.pem
- Set "Common name (CN)" to "Partner_A".
- Press Accept.
- One CPTT instance: Action, Simulate Controlled Station.
- The other CPTT instance: Action, Simulate Controlling Station.
- A TLS-secured connection should be established, IEC 60870-5-104 Frames should be transmitted.
- Controlling Station log should look alike:

```
     14:41:56.025
         IEC 60870-5-104 protocol profile:
           Link Layer:
             Max.Frame length net = 15, gross = 17
             Timeout: t0 = 30sec, t1 = 15sec, t2 = 10sec, t3 = 30sec
             Parameter: k = 12, w = 8
             Delay after connection closed = 10sec
             Log communication errors = YES
           Delay after client connection estab. failed = 30sec
           TLS-secured connection enabled
             Local host leaf certificate - certificate file: "ReaTho-Demo-For_CPTT_1-Partner_A-leaf_cert-cert.pem"
             Local host leaf certificate - private key file: "ReaTho-Demo-For_CPTT_1-Partner_A-leaf_cert-privkey.pem"
             Verify remote host leaf certificate: YES
             Remote host leaf cert. - certificate chain file: "ReaTho-Demo-For_CPTT_2-Partner_A-cert_chain-cert.pem"
             Remote host - Common name (CN): "Partner_A"
     14:41:56.088
         Trying to connect to Controlled Station at TCP:127.0.0.1:19998 ...
     14:41:56.390
         Connection established to remote 127.0.0.1:19998 via 127.0.0.1:60997
             TLS-secured connection: TLSv1.2, TLS-ECDHE-RSA-WITH-AES-256-GCM-SHA384
C>>  14:41:56.390
C>>      U: STARTDT act
<<D  14:41:56.600
<<D      U: STARTDT con
```

- Controlled station log should look alike:

```
     14:41:54.437
         IEC 60870-5-104 protocol profile:
           Application Layer:
             Invoke qualified Message List to respond = no
           Link Layer:
             Timeout: t0 = 30sec, t1 = 15sec, t2 = 10sec, t3 = 30sec
             Parameter: k = 12, w = 8
           TLS-secured connection enabled
             Local host leaf certificate - certificate file: "ReaTho-Demo-For_CPTT_1-Partner_A-leaf_cert-cert.pem"
             Local host leaf certificate - private key file: "ReaTho-Demo-For_CPTT_1-Partner_A-leaf_cert-privkey.pem"
             Verify remote host leaf certificate: YES
             Remote host leaf cert. - certificate chain file: "ReaTho-Demo-For_CPTT_2-Partner_A-cert_chain-cert.pem"
             Remote host - Common name (CN): "Partner_A"
     14:41:54.493
         Waiting for connection from Controlling Station to TCP:0.0.0.0:19998 ...
     14:41:56.383
         Connection accepted from remote 127.0.0.1:60997 to 127.0.0.1:19998
             TLS-secured connection: TLSv1.2, TLS-ECDHE-RSA-WITH-AES-256-GCM-SHA384
C>>  14:41:56.496
C>>      U: STARTDT act
<<D  14:41:56.497
<<D      U: STARTDT con
```

Explanation

- The "Certificate chain file" specified in CPTT instance 1 contains the Certificate chain to verify the Leaf certificate from CPTT instance 2. And vice versa. A "Certificate chain file" contains a chain of certificate (at least two cerfiticates).
- The "Common name (CN)" needs to be the same value as the item in the certificate of the opposite CPTT.
- Now, with "Verify peer certificate" enabled, both CPTT instances will show the opposite certificate (See Help, Support, View Support Info).

More explanation

- A pem file is a file in ASCII format which may contain a private key, a ceritificate or a certificate chain. You can examine this file. It starts with "-----BEGIN CERTIFICATE-----" or "-----BEGIN RSA PRIVATE KEY-----" or alike.
- In contrast a p7 file typically contains (in binary format) a certificate chain, a p12 file typically contains a certificate and a private key.
- We always have 2 or 3 files which belong together:
    - Leaf certificate certificate file: For CPTT instance 1, send to CPTT instance 2 to identify CPTT instance 1 and to encrypt sent data.
    - Leaf certificate private key file: For CPTT instance 1, used by CPTT instance 1 for decryption of received data.
    - Certificate chain file: For CPTT instance 2 to verify the certificate received from CPTT instance 1.

Contact

Real Thoughts GmbH
Welfenstrasse 35   phone +49 721 627 6730
76137 Karlsruhe   e-mail mailto:info@realthoughts.de
Germany   website https://www.realthoughts.de/