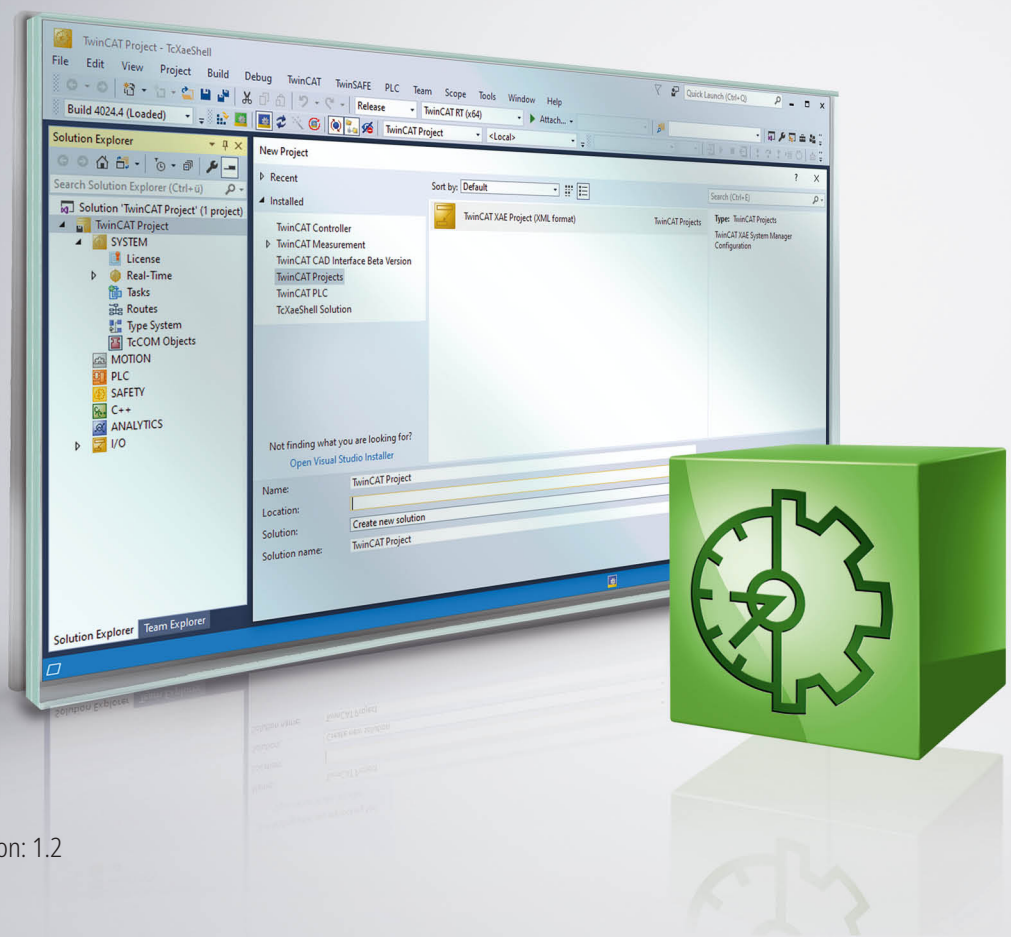
BECKHOFF New Automation Technology

Handbuch | DE

TE1000

TwinCAT 3 Secure ADS



09.11.2020 | Version: 1.2

Inhaltsverzeichnis

1 Vorwort

1.1 Hinweise zur Dokumentation

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs- und Automatisierungstechnik, das mit den geltenden nationalen Normen vertraut ist.
Zur Installation und Inbetriebnahme der Komponenten ist die Beachtung der Dokumentation und der nachfolgenden Hinweise und Erklärungen unbedingt notwendig.
Das Fachpersonal ist verpflichtet, für jede Installation und Inbetriebnahme die zu dem betreffenden Zeitpunkt veröffentliche Dokumentation zu verwenden.

Das Fachpersonal hat sicherzustellen, dass die Anwendung bzw. der Einsatz der beschriebenen Produkte alle Sicherheitsanforderungen, einschließlich sämtlicher anwendbaren Gesetze, Vorschriften, Bestimmungen und Normen erfüllt.

Disclaimer

Diese Dokumentation wurde sorgfältig erstellt. Die beschriebenen Produkte werden jedoch ständig weiter entwickelt.
Wir behalten uns das Recht vor, die Dokumentation jederzeit und ohne Ankündigung zu überarbeiten und zu ändern.
Aus den Angaben, Abbildungen und Beschreibungen in dieser Dokumentation können keine Ansprüche auf Änderung bereits gelieferter Produkte geltend gemacht werden.

Marken

Beckhoff®, TwinCAT®, EtherCAT®, EtherCAT G®, EtherCAT G10®, EtherCAT P®, Safety over EtherCAT®, TwinSAFE®, XFC®, XTS® und XPlanar® sind eingetragene und lizenzierte Marken der Beckhoff Automation GmbH.
Die Verwendung anderer in dieser Dokumentation enthaltenen Marken oder Kennzeichen durch Dritte kann zu einer Verletzung von Rechten der Inhaber der entsprechenden Bezeichnungen führen.

Patente

Die EtherCAT-Technologie ist patentrechtlich geschützt, insbesondere durch folgende Anmeldungen und Patente:
EP1590927, EP1789857, EP1456722, EP2137893, DE102015105702
mit den entsprechenden Anmeldungen und Eintragungen in verschiedenen anderen Ländern.

EtherCAT® ist eine eingetragene Marke und patentierte Technologie lizenziert durch die Beckhoff Automation GmbH, Deutschland

Copyright

1.2 Sicherheitshinweise

Sicherheitsbestimmungen

Beachten Sie die folgenden Sicherheitshinweise und Erklärungen!
Produktspezifische Sicherheitshinweise finden Sie auf den folgenden Seiten oder in den Bereichen Montage, Verdrahtung, Inbetriebnahme usw.

Haftungsausschluss

Die gesamten Komponenten werden je nach Anwendungsbestimmungen in bestimmten Hard- und Software-Konfigurationen ausgeliefert. Änderungen der Hard- oder Software-Konfiguration, die über die dokumentierten Möglichkeiten hinausgehen, sind unzulässig und bewirken den Haftungsausschluss der Beckhoff Automation GmbH & Co. KG.

Qualifikation des Personals

Diese Beschreibung wendet sich ausschließlich an ausgebildetes Fachpersonal der Steuerungs-, Automatisierungs- und Antriebstechnik, das mit den geltenden Normen vertraut ist.

Erklärung der Symbole

In der vorliegenden Dokumentation werden die folgenden Symbole mit einem nebenstehenden Sicherheitshinweis oder Hinweistext verwendet. Die Sicherheitshinweise sind aufmerksam zu lesen und unbedingt zu befolgen!

⚠ GEFAHR

Akute Verletzungsgefahr!

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, besteht unmittelbare Gefahr für Leben und Gesundheit von Personen!

⚠ WARNUNG

Verletzungsgefahr!

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, besteht Gefahr für Leben und Gesundheit von Personen!

⚠ VORSICHT

Schädigung von Personen!

Wenn der Sicherheitshinweis neben diesem Symbol nicht beachtet wird, können Personen geschädigt werden!

HINWEIS

Schädigung von Umwelt oder Geräten

Wenn der Hinweis neben diesem Symbol nicht beachtet wird, können Umwelt oder Geräte geschädigt werden.

Tipp oder Fingerzeig

Dieses Symbol kennzeichnet Informationen, die zum besseren Verständnis beitragen.

2 Allgemeine Beschreibung

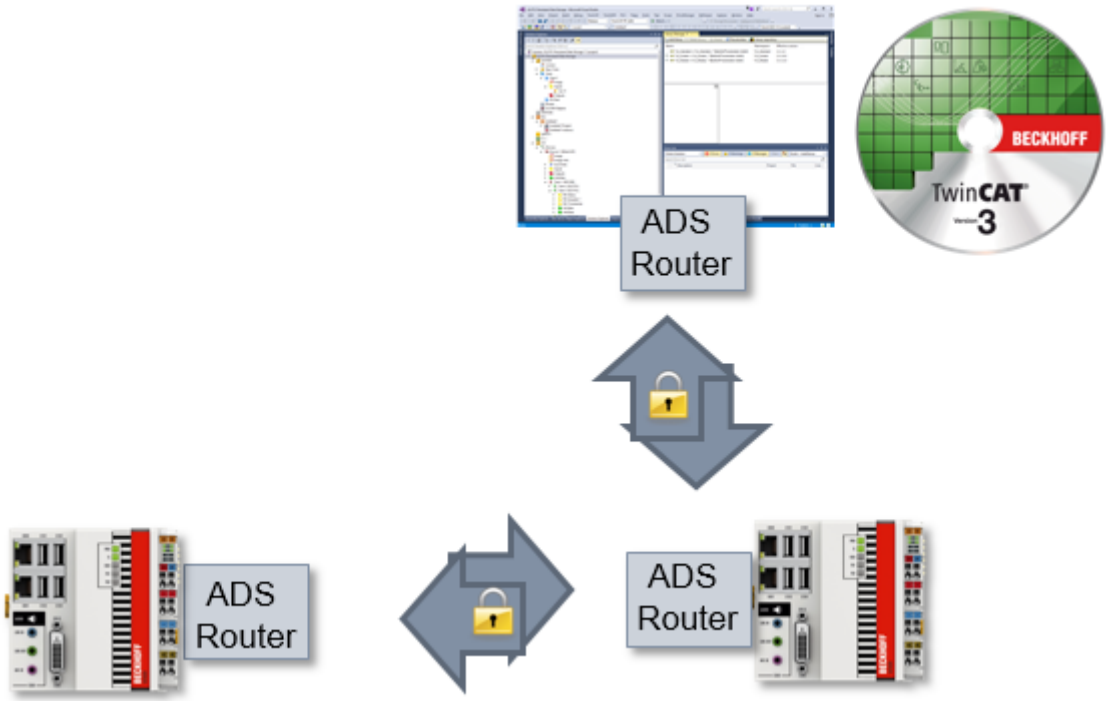
Ab TwinCAT 3.1. Build 4024.0

Die hier beschriebene Funktionalität ist ab TwinCAT 3.1. 4024.0 verfügbar.

Secure ADS ist aus Sicht des ADS Protokolls ein weiterer Transportkanal. Es werden exakt die gleichen ADS Kommandos über eine sichere Verbindung übertragen, wie auch über andere Kommunikationsprotokolle.

Hierfür wird eine mittels TLSv1.2 verschlüsselte Verbindung von einem TwinCAT Router zu einem anderen TwinCAT Router aufgebaut.

Durch die Implementierung innerhalb des TwinCAT Routers müssen existierende Anwendungen nicht modifiziert werden. Sie können durch einfache Parametrierung der verwendeten Route dazu gebracht werden, die verschlüsselte Verbindung zu verwenden.



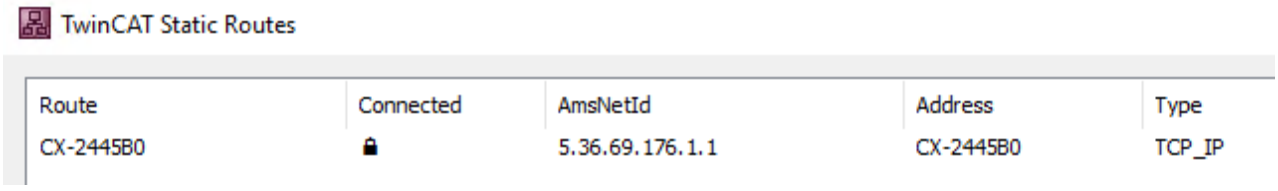
Diese Dokumentation stellt die unterschiedlichen Möglichkeiten von Secure ADS insbesondere im Blick auf die Bereitstellung der Schlüssel dar.

Erkennen einer SecureADS Route

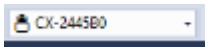
TwinCAT stellt eine SecureADS Route durch ein Schloss dar.

Dieses wird an den entsprechenden Stellen angezeigt:

- Routen-Übersicht eines Systems



- Bei der Auswahl des Zielsystems bei der Engineering-Umgebung XAE:

- 
CX-2445B0

3 Limitierungen

Ab TwinCAT 3.1. Build 4024.0

Die hier beschriebene Funktionalität ist ab TwinCAT 3.1. 4024.0 verfügbar.

- Secure ADS ist nur zwischen ADS Routern verfügbar.
- Für Secure ADS Verbindungen gilt, wie für alle anderen ADS Verbindungen, dass sie einen Vollzugriff für die verbundenen Systeme darstellen, wie es auch im Security Advisory 2017-01 beschrieben ist. Durch Unidirektionale [▶ 12] ADS Routen ist dieser Zugriff pro System konfigurierbar.

4 Voraussetzungen

Ab TwinCAT 3.1. Build 4024.0

Die hier beschriebene Funktionalität ist ab TwinCAT 3.1. 4024.0 verfügbar.

- Secure ADS ist Bestandteil von TC1000 und kann ohne Lizenzkosten genutzt werden.
- Die verwendeten Geräte benötigen eine Netzwerkkommunikation. Secure ADS wird über den TCP Port 8016 eingehend kommuniziert.
- Zur TLS-Verschlüsselung müssen ggf. entsprechende Zertifikate generiert und signiert werden.

5 Technische Einführung

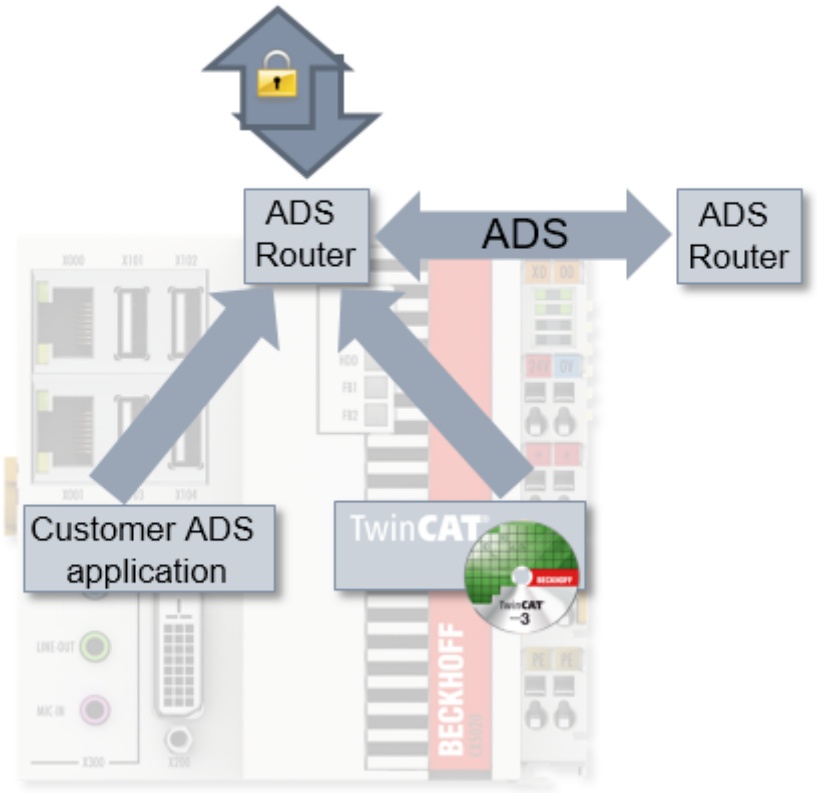
In diesem Abschnitt wird die grundsätzliche Funktionsweise unabhängig von der konkreten Konfiguration beschrieben.

Secure ADS führt einen zusätzlichen Kommunikationskanal für das bekannte ADS Protokoll ein. Diesen können Programme nutzen, ohne dass sie für den neuen Kommunikationskanal angepasst werden müssen.

Aus Security-Sicht handelt es sich also um eine Transportverschlüsselung aber keine Ende-zu-Ende-Verschlüsselung zwischen den Komponenten, denn alle lokal auf einem Gerät laufenden Anwendungen können diese verschlüsselte Verbindung gemeinsam nutzen – genau, wie es auch für ADS-Routen ist.

Lokale Realisierung

Secure ADS ist Bestandteil des ADS-Routers und wird auch hier konfiguriert. Der ADS Router baut eine verschlüsselte Verbindung zu einem anderen TwinCAT Router auf und stellt diese den Anwendungen bereit. Es ist also darauf zu achten, dass die ADS Geräte Anwendungen nicht selber verschlüsselt kommunizieren, sondern dies zwischen den Routern erfolgt.



Transparente Nachrüstung

Durch die Realisierung von Secure ADS innerhalb des TwinCAT Routers wird ein nachträgliches „Retrofitting“ von Anwendungen ermöglicht. Alle ADS Anwendungen (Client und Server), dazu zählen auch Anwendungen die von Kunden geschrieben wurden, müssen nicht neu übersetzt werden.

Die ADS Anwendungen nutzen ADS Routen, um den Kommunikationspartner zu identifizieren. Diese ADS Route ist unabhängig vom Transportkanal und wird im TwinCAT Router beschrieben.

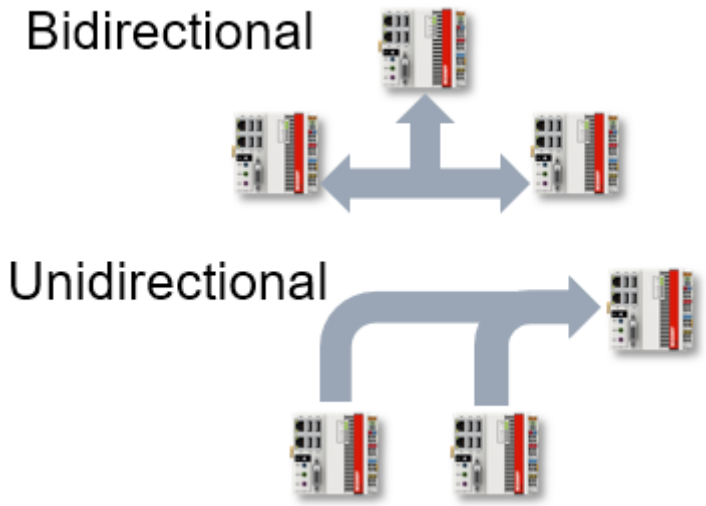
Wenn die genutzte Route auf eine Secure ADS Verbindung umgestellt wird, wird der ADS-Verkehr verschlüsselt transportiert.

5.1 Gerichtete ADS Kommunikation

Eine Eigenschaft von ADS Routen ist, dass diese gerichtet sein können. Diese Eigenschaft wurde im Rahmen von Secure ADS ergänzt, ist jedoch allgemein für Routen verfügbar.

ADS Routen werden, nachdem sie auf Netzwerkebene geöffnet wurden, beidseitig von den jeweiligen ADS Anwendungen zur Kommunikation genutzt. Dieses Verhalten ist sehr effizient, aber jedoch ggf. unerwünscht. Beispielsweise soll ein Engineering-Rechner (XAE) im Normalfall zwar per ADS Zugriff auf ein Runtime (XAR) System haben, jedoch ist es nicht notwendig, dass ein XAR-System per ADS auf das XAE System zugreift.
Diese Richtung kann deswegen eingeschränkt werden, welches dadurch realisiert wird, dass ein entsprechendes System (im Beispiel das XAE) keine ADS-Request Befehle über die Route akzeptiert.

Das Kapitel Konfiguration [▶ 14] beschreibt das Vorgehen, um die Eigenschaften einzuschränken.



5.2 Server

Eine normale ADS Route wird von beiden Teilnehmern aufgebaut, sobald diese benötigt wird.
Ist eine Route einmal aufgebaut, wird diese in beiden Richtungen genutzt.
Als Erweiterung für Secure ADS wird eine Server-Konfiguration angeboten. Eine solche Konfiguration stellt die Basis dar um konkrete Routen einzurichten.

```
<TcConfig>
  <RemoteConnections>
    <Server>
      …
    </Server>
  </RemoteConnections>
</TcConfig>
```

Für PSK [▶ 16] sowie Kunden-bereitgestellte Zertifikate [▶ 17] wird dieses genutzt, um die initiale Konfiguration auf einer Seite abzulegen.
Beim Einrichten der konkreten Route werden dann die Server-Einträge überprüft, ob eine Berechtigung vorliegt. Wenn dieses der Fall ist, wird eine normale Route eingerichtet.

Sehen Sie dazu auch

- Konfiguration [▶ 14]

5.3 Schlüsselaustausch

Es werden drei Möglichkeiten durch Secure ADS angeboten um die zur Verschlüsselung nötigen Schlüssel bereitzustellen, welche hier mit ihren Vor- und Nachteilen beschrieben werden sollen.

Allen gemein ist, dass das jeweilige Gerät in Bezug auf den Zugriff auf die Geheimnisse (PreSharedKeys, Zertifikate) abgeschottet werden muss. Werden diese Geheimnisse kompromittiert, ist das System neu aufzusetzen um die Integrität des Gesamtsystems wieder herzustellen.

SelfSigned Certificates (SSC)

TwinCAT erzeugt beim ersten Starten (z.B. nach der Installation) ein selbst-signiertes Zertifikat.

Das Nutzen solcher Zertifikate hat den Vorteil, dass sie lokal erzeugt werden und bereitstehen. Um ein Vertrauensverhältnis aufzubauen muss jedoch zwischen allen Kommunikationsteilnehmern jeweils eine Überprüfung der Zertifikate durchgeführt werden.

Diese Zertifikate eignen sich also für die initiale Inbetriebnahme oder auch statische Maschinen, die ohne Dynamik in der Anlagenstruktur oder auch den Zugangsberechtigten auskommen.

Ab TwinCAT 4024.0 werden diese Zertifikate als Standard bei der Nutzung vorgesehen. Im Kapitel Konfiguration [▶ 15] wird beschrieben, wie sie zum Aufbau einer ADS-Route eingesetzt werden.

Laufzeiten der Zertifikate

Die erzeugten Zertifikate haben eine feste Laufzeit von 1.1.2000 bis 1.1.2061. Aus Security-Sicht ist dieses zu lang, sodass organisatorische Maßnahmen getroffen werden müssen um den Security-Ansprüchen zu genügen. Beckhoff stellt durch diese über-lange Laufzeit sicher, dass eine Kommunikation nicht fehlschlägt, auch wenn beispielsweise falsche Zeiten im lokalen System eingestellt sind.

Sollte dieses Verhalten unerwünscht sein, können eigene Zertifikate bereitgestellt und genutzt werden (vgl. Kunden-bereitgestellte Zertifikate).

PreSharedKeys (PSK)

In einem TwinCAT System können Pre-Shared-Keys abgelegt werden. Diese werden zur Autorisierung der eingehenden ADS-Routen beim Aufbau der Verbindung genutzt.

Da die Pre-Shared-Keys konfiguriert werden müssen, eignen sie sich insbesondere um z.B. Wartungspersonal Zugriff zu gewähren. Dabei können die Pre-Shared-Keys Personen-gebunden verwendet werden.

Pre-Shared-Keys haben keine Laufzeit, wie es für Zertifikate vorgesehen ist. Auch werden sie direkt in Dateien abgelegt, sodass sie nicht (wie normalerweise Passwörter) als Hashwert abgelegt werden. Sie sind damit gegen direkte Einsichtnahme nicht geschützt.

Im Kapitel Konfiguration [▶ 16] wird beschrieben, wie Pre-Shared-Keys auf beiden Seiten der Kommunikation verwendet werden.

Kunden-bereitgestellte Zertifikate (CA mit Zertifikaten)

Secure ADS stellt auch die Möglichkeit bereit, dass Kunden eigene Zertifikate erzeugen und verwalten.

Dadurch sind insbesondere dynamische Konstellationen gut abbildbar, da es eine gemeinsame Certificate Authority (CA) geben kann. Alle Teilnehmer, die dieser CA vertrauen, können ohne weitere Konfiguration verschlüsselt untereinander kommunizieren, selbst wenn sie sich zuvor nicht begegnet sind.

Im Kapitel Konfiguration [▶ 17] wird beschrieben, wie diese Zertifikate in TwinCAT eingebunden werden können.

HINWEIS

Ablauf der Zertifikate

Zertifikate haben ein Ablaufdatum. Es müssen organisatorische Maßnahmen getroffen werden, um Zertifikate vor ihrem Ablauf zu ersetzen.

6 Konfiguration

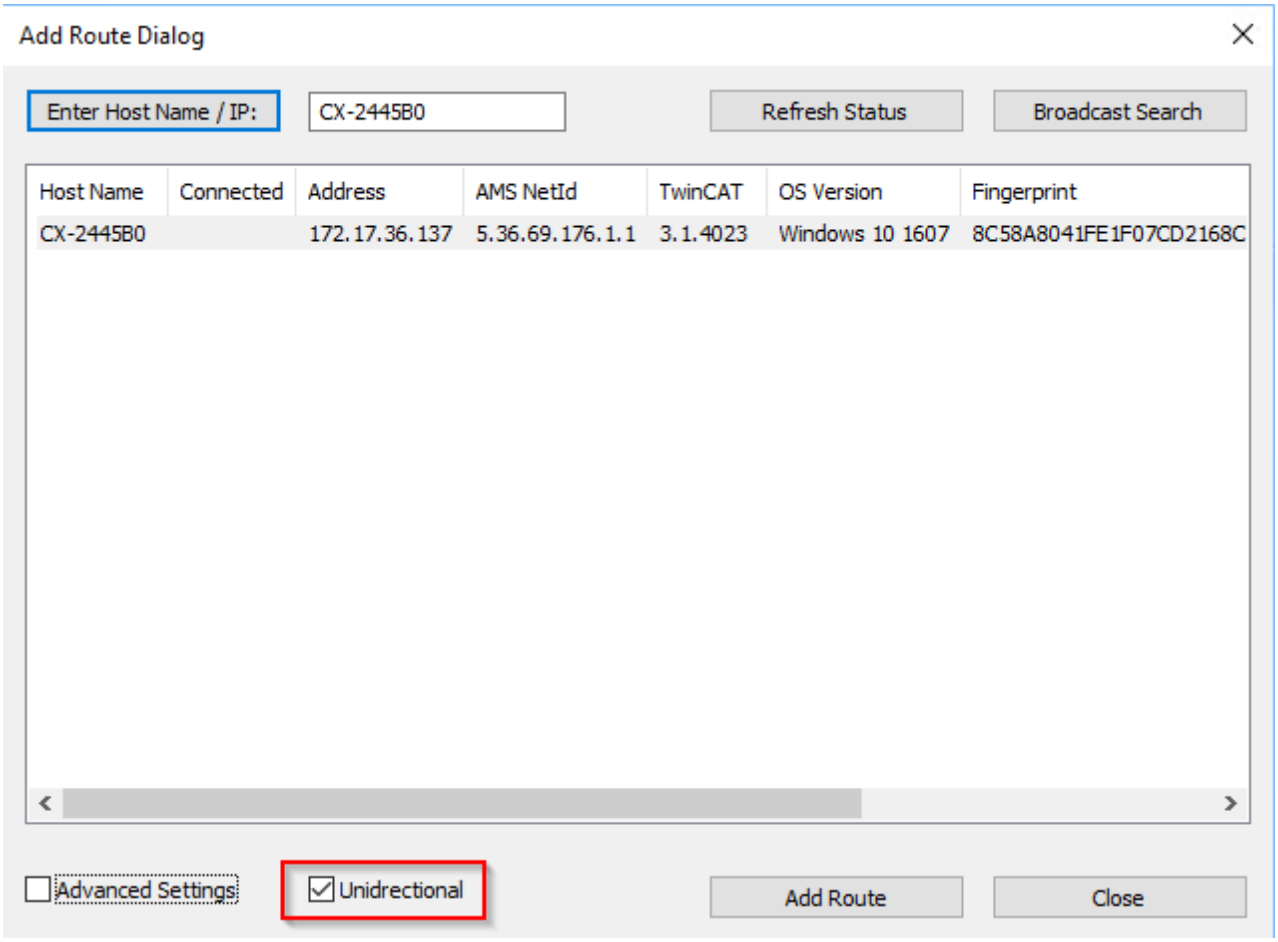
Es werden drei Möglichkeiten durch Secure ADS angeboten, um die zur Verschlüsselung notwendigen Schlüssel bereitzustellen. An dieser Stelle wird die Konfiguration getrennt voneinander beschrieben.

Während die Konfiguration Server vs. Route innerhalb der drei Möglichkeiten beschrieben wird, werden gerichtete ADS Verbindungen [▶ 14] unabhängig dargestellt.

6.1 Gerichtete ADS Kommunikation

Die Konfiguration einer gerichteten ADS Kommunikation erfolgt über die Checkbox **Unidirectional** beim Anlegen der Route.

Ist diese Checkbox gesetzt, wird TwinCAT über die zugehörige Route keine ADS Befehl-Aufrufe vom gegenüberliegenden Zielsystem entgegennehmen. Selbst werden ADS Befehl-Aufrufe (Requests) gesendet und auch Antworten (Response) empfangen.



In der XML-Konfiguration erfolgt diese Einstellung über das Attribut `Unidirectional="true"` erfolgen:

```
<RemoteConnections>
<Route Unidirectional="true">
<Name>CX-123456</Name>
<Address>CX-123456</Address>
<NetId>5.36.69.176.1.1</NetId>
<Type>TCP_IP</Type>
<Flags>128</Flags>
<Tls>
…
</Tls>
</Route>
</RemoteConnections>
```

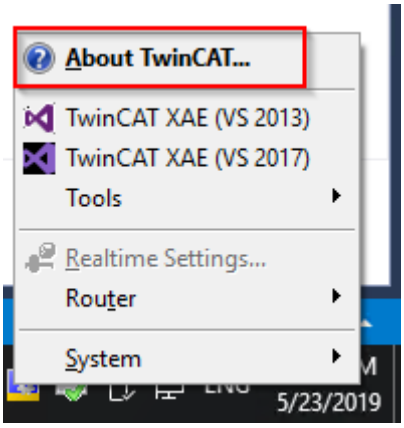
6.2 SelfSigned Zertifikate (SSC)

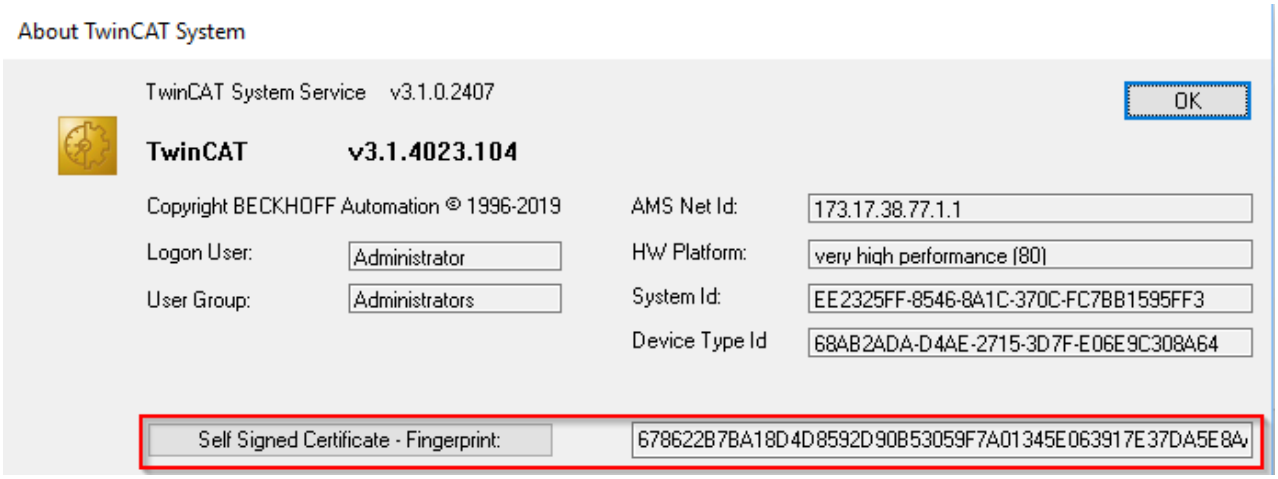
Selbst-Signierte Zertifikate benötigen beim Einrichten der Verbindung die Überprüfung des Kommunikationsteilnehmers, da automatisiert kein Vertrauensverhältnis existiert.

Diese Überprüfung wird in TwinCAT durch den Fingerprint des gegenüberliegenden Systems ermöglicht.

Anzeige des SSC-Fingerprints auf einem System

Der Fingerprint des eigenen Systems wird im **About TwinCAT** Dialog angezeigt:



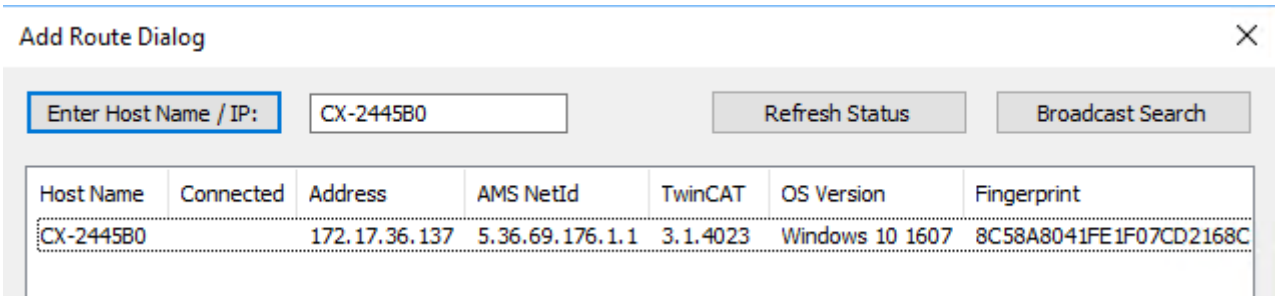


Der Button **Self Signed Certificate – Fingerprint:** kopiert den rechts aufgeführten Fingerprint in die Zwischenablage.

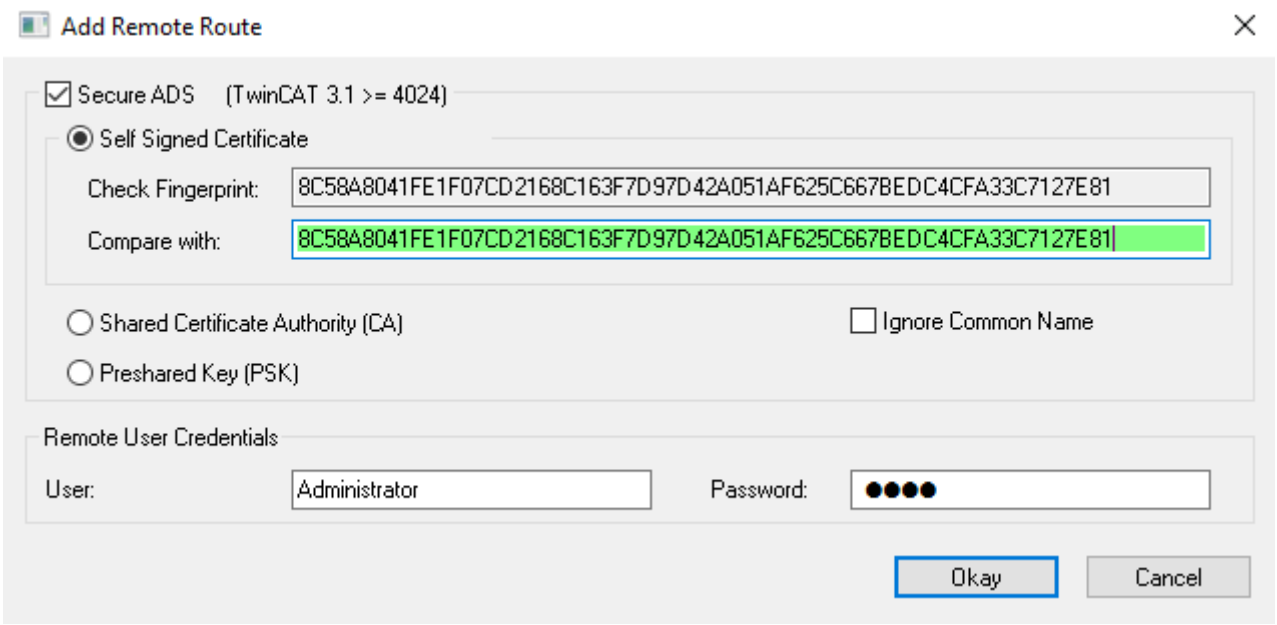
Für CE Systeme existiert dieser Dialog nicht. Hier kann in der Datei `\Hard Disk\TwinCAT\3.1\Target \TcSelfSigned.xml` der Fingerprint angezeigt werden.

Aufbau der Verbindung

Der Fingerprint wird rein informativ und kryptographisch nicht gesichert nach dem Discovery angezeigt:



Die endgültige Überprüfung des Fingerprints findet beim Einrichten der Route statt:



Das **Compare with** Feld kann dabei z. B. beim Copy&Paste zur Überprüfung verwendet werden: Wird dort der gleiche Fingerprint eingetragen, erscheint das Feld grün, sonst rot.

Damit kann beispielsweise eine RDP-Verbindung genutzt werden, um den Fingerprint eines Systems über den **Self Signed Certificate – Fingerprint-**Button in die Zwischenablage zu bekommen und hier einzufügen.

Damit das Zielsystem den Routenaufbau akzeptiert, wird ein dort gültiger System-Login mit entsprechenden Administrator-Rechten genutzt.
Diese Login-Daten werden bereits verschlüsselt übertragen.

Bei CE-Systemen wird mit TwinCAT 3.1 4024.5 immer der Hostname eingetragen, auch wenn beim Anlegen der Route **IP-Adresse** ausgewählt wurde. Sollte also ein Netzwerk ohne funktionierenden Hostname-Lookup genutzt werden, muss in der Datei `\Hard Disk\TwinCAT\3.1\Target\StaticRoutes.xml` manuell der Hostname durch die IP-Adresse geändert werden.

6.3 Preshared Keys (PSK)

Pre-Shared-Keys werden auf einer Seite als Server eingerichtet und auf einer Seite zur Authentisierung und Autorisierung genutzt.

Einrichten von Preshared Keys als Server

Preshared Keys werden im Normalfall mit Server-Verbindungen eingesetzt werden.
Die Konfiguration erfolgt über einen Eintrag in der Routen-Konfiguration.

Hierfür können in der *C:\TwinCAT\3.x\Target\StaticRoutes.xml* Datei folgende Einträge vorgenommen werden:

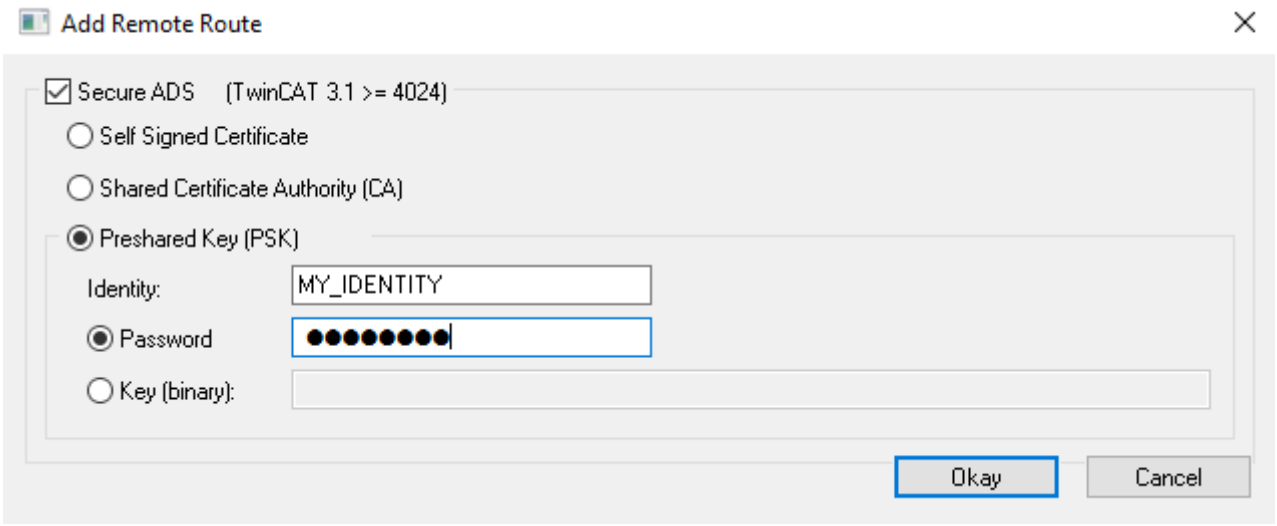
```
<?xml version="1.0"?>
<TcConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<RemoteConnections>
<Server>
<Tls>
<Psk>
<Identity>MY_IDENTITY</Identity>
<Pwd>MySecret</Pwd>
</Psk>
<Psk>
<Identity>MY_IDENTITY2</Identity>
```

```
<Pwd>MyOtherSecret</Pwd>
</Psk>
</Tls>
</Server>
</RemoteConnections>
</TcConfig>
```

Gespeicherte Änderungen werden übernommen, wenn der TwinCAT Router initialisiert wird, was beispielsweise bei den Übergängen von RUN->CONFIG oder auch CONFIG->CONFIG erfolgt.

Nutzen einer Preshared Keys Servers

Beim Hinzufügen einer Route wird der Eintrag **Preshared Key (PSK)** ausgewählt und die entsprechenden Credentials eingetragen.



Wenn dieses erfolgreich ist, wird eine konkrete Route im Zielsystem hinterlegt, welche für zukünftige Verbindungsaufbauten genutzt wird.

6.4 Kunden-bereitgestellte Zertifikate (CA mit Zertifikaten)

Die Konfiguration von Kunden-bereitgestellten Zertifikaten erfolgt über einen Eintrag in der Routen-Konfiguration.

Hierfür können in der *C:\TwinCAT\3.x\Target\StaticRoutes.xml* Datei folgende Einträge vorgenommen werden:

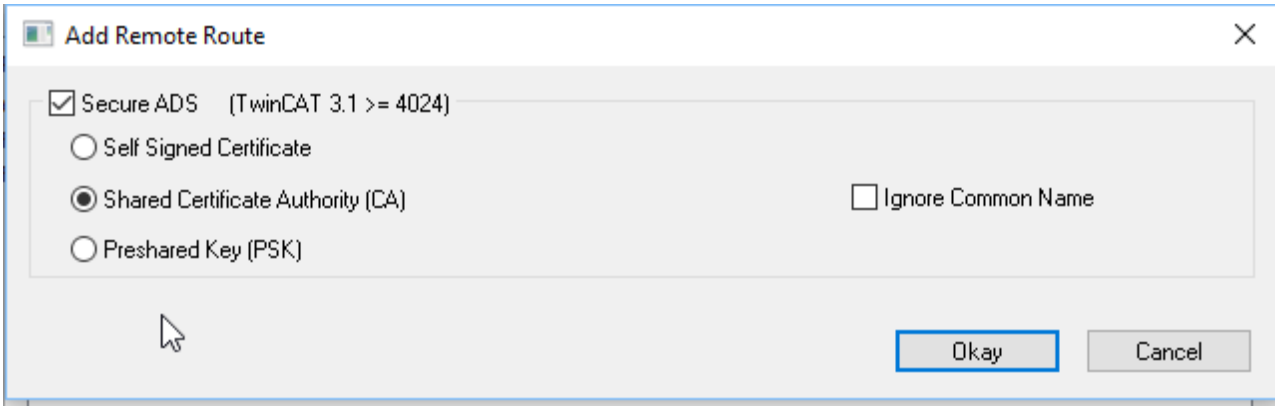
```
<?xml version="1.0"?>
<TcConfig xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<RemoteConnections>
<Server>
<Tls IgnoreCn="true"> <!--see below-->
  <Ca>C:\TwinCAT\3.1\Target\CACerts\rootCA.pem</Ca>
  <Cert>C:\TwinCAT\3.1\Target\CACerts\ipc.crt</Cert>
              <Key>C:\TwinCAT\3.1\Target\CACerts\ipc.key</Key>
            </Tls>
          </Server>
</RemoteConnections>
</TcConfig>
```

Gespeicherte Änderungen werden übernommen, wenn der TwinCAT Router initialisiert wird, was beispielsweise bei den Übergängen von RUN->CONFIG oder auch CONFIG->CONFIG erfolgt.

Die Zertifikate sind dabei X.509 Zertifikate, wie sie beispielsweise mit OpenSSL generiert werden können. Sollte der Schlüssel (XML-Element `<Key>`) durch ein Passwort geschützt sein, kann dieser über das XML-Element `<KeyPwd>` angegeben werden. Es wird das .der und .pem Format unterstützt.

Der „CommonName“ des Zertifikates muss dabei dem beim Verbindungsaufbau genutzten Namen (XML-Element `<Name>`) entsprechen. Dieses Verhalten kann durch die Option `IgnoreCn="true"` abgeschaltet werden.

Wenn beide Seiten passende Zertifikate einer gemeinsamen CA haben, kann die Route ohne weitere Informationen mittels diesen Dialogs Angelegt werden:



Wie unter Server [▶ 12] beschrieben, wird hierdurch auf beiden Seiten eine konkrete Route angelegt.

6.5 Abschalten von ADS

- Das unverschlüsselte ADS wird über den TCP Port 48898 (0xBF02) übertragen
- Das Discovery („Broadcast Search“) wird über den UDP Port 48899 (0xBF03) übertragen

Beide Ports können in der Firewall blockiert werden.

Das Zielsystem kann in Bezug auf die zu nutzenden Ports konfiguriert werden.

Unterhalb von KEY_LOCAL_MACHINE\SOFTWARE\[WOW6432Node\]Beckhoff\TwinCAT3\System sind die folgenden Schlüssel verfügbar:

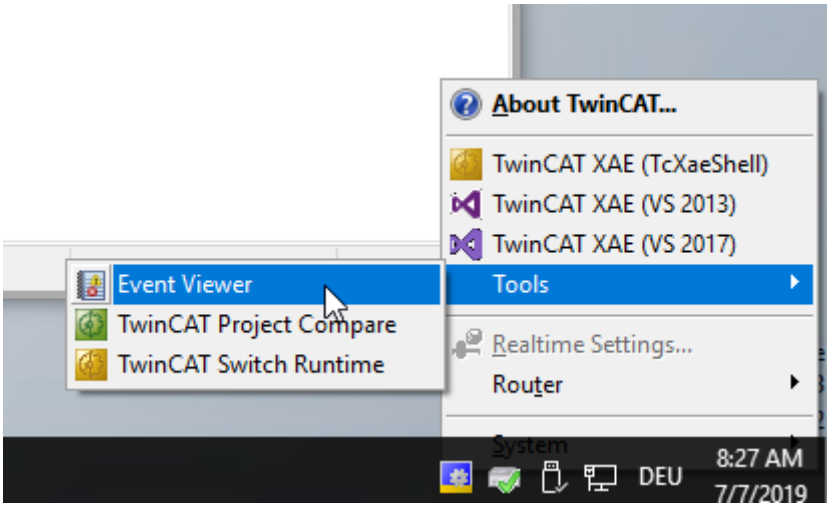
ADS Ports		
DisableAdsTcpListening	REG_DWORD	1 = Verhindert das Öffnen des TCP Ports 0xBF02 für unverschlüsseltes ADS.
DisableAdsTlsListening	REG_DWORD	1 = Verhindert das Öffnen des TCP Ports 8016 für Secure ADS
DisableAdsDiscovery	REG_DWORD	1 = Verhindert das Öffnen des UDP Ports 0xBF03 für das ADS Discovery („Broadcast Search“)

Über die Datei StaticRoutes.xml kann zusätzlich das Attribut `SecureOnly="True"` verwendet werden. Der ADS Port 0xBF02 wird dabei weiterhin geöffnet, jedoch wird über den Port keine ADS Kommunikation mehr erlaubt.

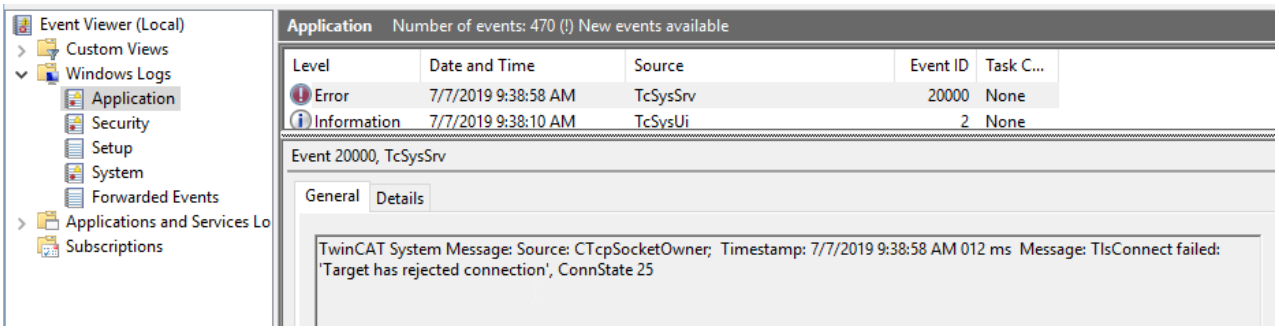
```
<RemoteConnections SecureOnly="True">
```

6.6 Logging

Secure ADS schreibt Informationen über fehlgeschlagene Verbindungsaufbauten in das Windows Event Log, welches über das TwinCAT System Tray Icon verfügbar ist.



Die Meldungen befinden sich unter der Kategorie **Windows Logs > Application**:



7 Beispiel

7.1 Kunden-bereitgestellte Zertifikate (CA mit Zertifikaten)

An dieser Stelle wird mittels OpenSSL Zertifikate erzeugt, die für die Secure ADS Verbindung genutzt werden können.

Diese Anleitung stellt keine umfassende Beratung zur Erstellung und Umgang mit Zertifikaten dar. Insbesondere die Laufzeiten müssen beachtet werden, welches organisatorische Maßnahmen erfordert um vor Ablauf der Gültigkeiten (hier: 3600 Tage für die CA und 360 Tage für die jeweiligen Zertifikate) für einen Austausch zu sorgen.

In diesem Beispiel wird eine Certificate Authority (CA) erzeugt, die für beide Seiten (hier IPC und CX genannt) der Kommunikation ein Zertifikat unterschreibt.

Die Bedeutung der Aufrufparameter kann im Detail durch „`openssl help`“ nachgesehen werden.

✓ OpenSSL ist installiert und als verfügbar von der Kommandozeile.

1. Erzeugen eines Schlüssels für die Certificate Authority, der später vertraut wird.
   ```
   openssl genrsa -out rootCA.key 2048
   ```

2. Erzeugen des Zertifikats mit einer Laufzeit von 3600 Tagen. Über den Parameter „-subj“ werden Inhaberinformationen beigesteuert.
   ```
   openssl req -x509 -new -nodes -key rootCA.key -sha256 -subj "/C=DE/ST=NRW/
   L=Verl/O=Bk/OU=TCPM/CN=RootCA" -days 3600 -out rootCA.pem
   ```

3. Erzeugen eines Schlüssels für den IPC
   ```
   openssl genrsa -out ipc.key 2048
   ```

4. Erzeugen eines Certificate Signing Requests (CSR) für diesen Schlüssel:
   Bitte beachten: Die als CN angegebene Adresse (hier IP Adresse) muss als Namen beim Verbindungsaufbau verwendet werden. Alternativ muss die Route mit IgnoreCN parametriert warden.
   ```
   openssl req -out ipc.csr -key ipc.key -subj "/C=DE/ST=NRW/L=Verl/O=Bk/
   OU=TCPM/CN=192.168.2.1" -new
   ```

5. Signieren des CSR des IPC mit der CA mit Gültigkeit von 360 Tagen
   ```
   openssl x509 -req -in ipc.csr -CA rootCA.pem -CAkey rootCA.key -
   CAcreateserial -out ipc.crt -days 360 -sha256
   ```
   ⇨ Mittels dieser Dateien kann die Route nun auf dem IPC eingerichtet werden: rootCA.pem, ipc.key und ipc.pem

6. Erzeugen eines Schlüssels für den CX
   ```
   openssl genrsa -out cx.key 2048
   ```

7. Erzeugen eines Certificate Signing Requests (CSR) für diesen Schlüssel:
   Bitte beachten: Die als CN angegebene Adresse (hier IP Adresse) muss als Namen beim Verbindungsaufbau verwendet werden. Alternativ muss die Route mit IgnoreCN parametriert warden.
   ```
   openssl req -out cx.csr -key cx.key -subj "/C=DE/ST=NRW/L=Verl/O=Bk/OU=TCPM/
   CN=192.168.2.2" -new
   ```

8. Signieren des CSR des IPC mit der CA mit Gültigkeit von 360 Tagen
   ```
   openssl x509 -req -in cx.csr -CA rootCA.pem -CAkey rootCA.key -CAcreateserial
   -out cx.crt -days 360 -sha256
   ```
   ⇨ Mittels dieser Dateien kann die Route nun auf dem CX eingerichtet werden: rootCA.pem, cx.key und cx.pem

⇨ Die Route kann genutzt werden.

Mehr Informationen:
www.beckhoff.de/te1000/

Beckhoff Automation GmbH & Co. KG
Hülshorstweg 20
33415 Verl
Deutschland
Telefon: +49 5246 9630
info@beckhoff.de
www.beckhoff.de