IT-Sicherheit-Empfehlung

Verschlüsselte Datenbank-Kommunikation (MS SQL Server)

Stand: Oktober 2021

Inhalt

1 Allgemeines

Ergänzend zum Empfehlungskatalog **Sicherer kommunaler IT-Arbeitsplatz** behandelt dieses Dokument die Umsetzung des Anteils **Verschlüsselung der Netzwerk-Kommunikation** (Abschnitt 3.7) in ausführlicher Form. Es handelt sich hierbei um eine priorisierte Empfehlung der Studie, die für sämtliche Verfahren (Uniface Client-Server, SYNERGO®, Sonstige) innerhalb der Behörde mit Datenbankanbindung gültig ist. In den folgenden Kapiteln werden die Möglichkeiten der Verschlüsselung der Datenbank-Kommunikation für Microsoft SQL Server beschrieben.

Analoge Schritte sind gleichermaßen bei Einsatz von Oracle als Datenbanksystem geboten.

Mit der TLS-Verschlüsselung (Transport Layer Security) können verschlüsselte Daten in einem Netzwerk zwischen einer Instanz von SQL Server und einer Clientanwendung übertragen werden.

Transport Layer Security (TLS) ist ein Protokoll zum Einrichten eines sicheren Kommunikationskanals, um das Abfangen von kritischen oder vertraulichen Informationen im Netzwerk und bei anderen Formen der Internetkommunikation zu verhindern. Durch TLS können der Client und der Server gegenseitig ihre Identität authentifizieren. Nach dem Authentifizieren der Teilnehmer stellt TLS verschlüsselte Verbindungen zwischen ihnen bereit, damit die Nachrichten sicher übertragen werden können.

Das Aktivieren der TLS-Verschlüsselung erhöht die Sicherheit von Daten, die netzwerkübergreifend zwischen Instanzen von SQL Server und Anwendungen übertragen werden. Dies kann – je nach Menge der verschlüsselt übertragenen Daten – zu etwas erhöhten Antwortzeiten in der Client-Server-Kommunikation führen.

2 Selbstsigniertes Zertifikat – die Verschlüsselungsstärke entspricht einem verifizierbaren Zertifikat

SQL Server unterstützt die Verwendung eines selbstsignierten Zertifikates zur Verbindungsverschlüsselung. Bei selbstsignierten Zertifikaten kann die Sicherheit nicht garantiert werden. Als Mindest-Sicherheitsstufe wird jedoch damit das Lesen der im Netzwerk übertragenen Daten im Klartext verhindert.
Sie sollten für eine sichere Konnektivität unbedingt ein verifizierbares Zertifikat für SQL Server bereitstellen. Ein selbstsigniertes Zertifikat entspricht aber von der Verschlüsselungsstärke dem eines verifizierbaren Zertifikats.

Ausführliche Informationen finden Sie unter:

- Verwenden von Verschlüsselung ohne Überprüfung in SQL Server Native Client
- Protokolle für MSSQLSERVER-Eigenschaften (Registerkarte Flags)

2.1 Konfiguration der Verschlüsselung

Die folgenden Einstellungen werden ausschließlich auf dem Datenbank-Server vorgenommen.

SQL Server Konfigurations-Manager:

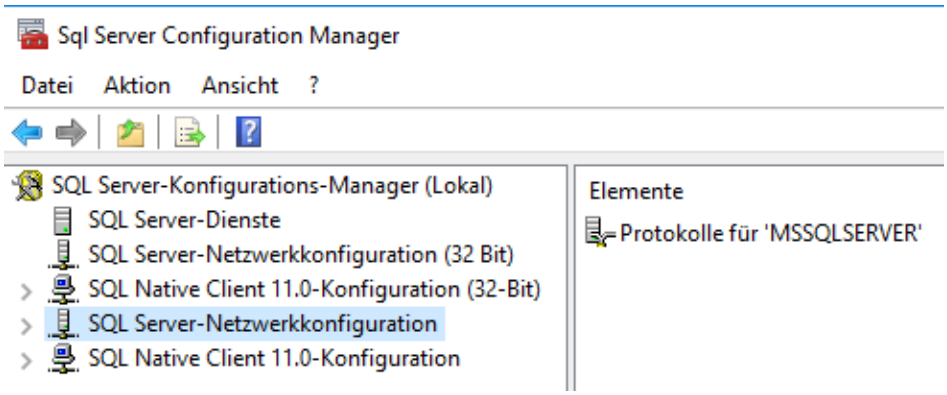

Abbildung 1

➡ SQL SERVER-NETZWERKKONFIGURATION – PROTOKOLLE FÜR '…' – EIGENSCHAFTEN

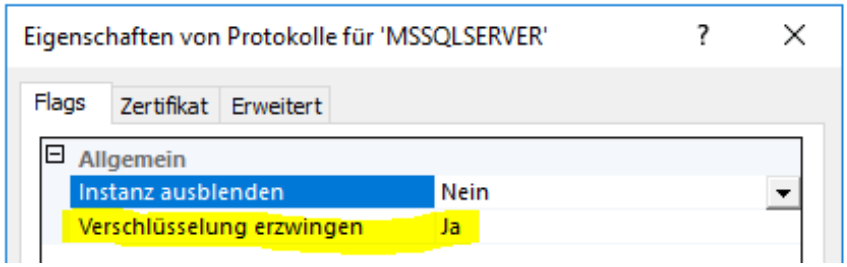

Abbildung 2

➡ Um Verbindungen mit 32- und 64-Bit-Treibern zu verschlüsseln, müssen die folgenden Konfigurationen an beiden Stellen im Konfigurations-Manager vorgenommen werden.

➡ SQL NATIVE CLIENT KONFIGURATION (32-BIT) – EIGENSCHAFTEN

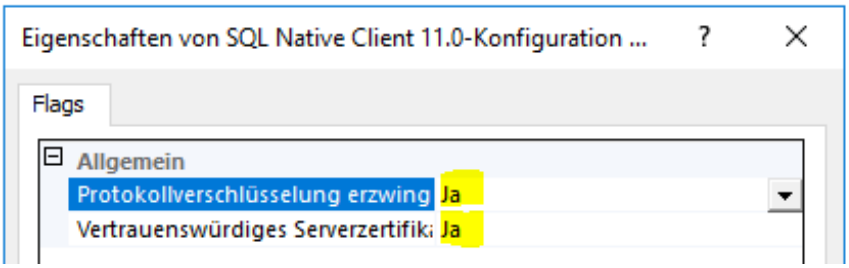

Abbildung 3

➡ SQL NATIVE CLIENT KONFIGURATION – EIGENSCHAFTEN

➡ Um Verbindungen mit 32- und 64-Bit-Treibern zu verschlüsseln, muss die Konfiguration an beiden Stellen im Konfigurations-Manager vorgenommen werden.

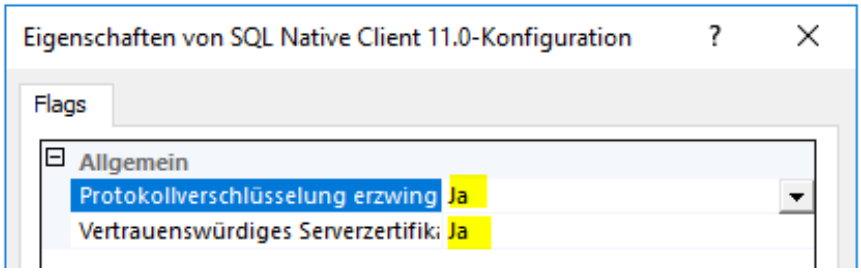

Abbildung 4

➡ Starten Sie den SQL Server-Dienst neu.

Auf dem Client sind keine zwingenden Änderungen notwendig, die Verschlüsselung wird vom Server erzwungen.

Beispiel ODBC-Verbindung:

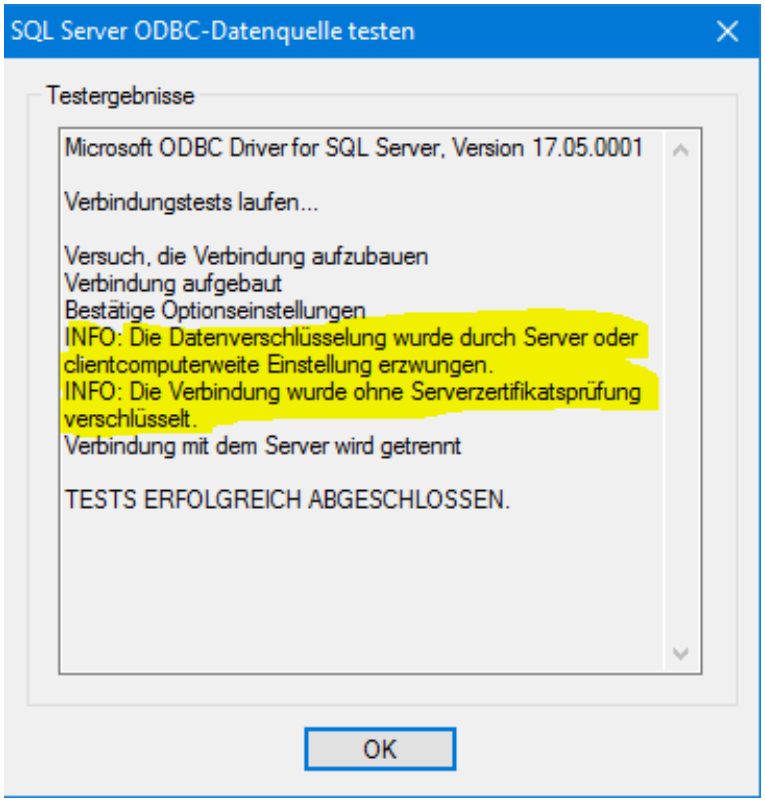

Abbildung 5

2.2 Überprüfung der Verschlüsselung

```
SELECT session_id, connect_time, net_transport, encrypt_option, auth_scheme,
client_net_address

FROM sys.dm_exec_connections;
```

Ergebnisse | Meldungen

	session_id	connect_time	net_transport	encrypt_option	auth_scheme	client_net_address
1	51	2020-09-21 13:12:13.330	Shared memory	TRUE	SQL	<local machine>
2	52	2020-09-21 13:12:17.127	Shared memory	TRUE	SQL	<local machine>
3	53	2020-09-21 13:19:44.477	TCP	TRUE	SQL	172.18.27.84
4	54	2020-09-21 13:23:06.040	TCP	TRUE	SQL	172.18.27.84
5	55	2020-09-21 11:22:34.310	TCP	TRUE	SQL	172.18.27.84
6	57	2020-09-21 11:22:34.983	TCP	TRUE	SQL	172.18.27.84

Abbildung 6

encrypt_option nvarchar(40) Boolescher Wert, der angibt, ob die Verschlüsselung für diese Verbindung aktiviert ist. Lässt keine NULL-Werte zu.

WICHTIG:

Bei einem selbstsigniertem SQL Server-Zertifikat reicht die obige Konfigurierung aus. Das Zertifikat wird zur Laufzeit vom SQL Server zur Verfügung gestellt. Die Verbindung wird verschlüsselt (TLS) und entspricht vom Verschlüsselungsgrad einem verifizierbaren Zertifikat (siehe Kapitel 3).

3 Verifizierbares Zertifikat (CA – Zertifizierungsstelle)

Das zur Laufzeit vom SQL Server bereitgestellte Zertifikat ist nicht überprüfbar. Um die Überprüfung zu ermöglichen, wird empfohlen, ein von einer Zertifizierungsstelle signiertes Zertifikat zu verwenden. Dadurch wird die Identität des Computers und der Instanz von SQL Server durch die Zertifikatkette sichergestellt.

Im Kontext einer Active Directory Domäne kann ein Serversystem die Rolle einer vertrauenswürdigen Zertifizierungsstelle übernehmen, die von allen Clients der Domäne als legitimer Vertrauensanker akzeptiert wird.

3.1 Installation auf einem einzelnen Server

Mit SQL Server 2019 (15.x) ist die Zertifikatverwaltung im SQL Server-Konfigurations-Manager integriert. Der SQL Server-Konfigurations-Manager für SQL Server 2019 (15.x) kann mit früheren Versionen von SQL Server verwendet werden. Informationen zum Hinzufügen eines Zertifikats auf einer einzelnen SQL Server-Instanz finden Sie unter:

- Zertifikatverwaltung (SQL Server-Konfigurations-Manager)
- https://sid-500.com/2017/03/31/active-directory-zertifikatsdienste-teil-1-installation-einer-enterprise-root-ca/

  UND
- Aktivieren von verschlüsselten Verbindungen zur Datenbank-Engine

3.1.1 Konfigurations-Manager bis SQL Server 2017

Wenn Sie SQL Server 2012 (11.x) bis SQL Server 2017 (14.x) verwenden und der SQL Server-Konfigurations-Manager für SQL Server 2019 (15.x) nicht verfügbar ist, führen Sie die folgenden Schritte aus:

- Klicken Sie im Menü **Start** auf **Ausführen**, geben Sie in das Feld **Öffnen** den Wert **MMC** ein und klicken Sie dann auf **OK**.
- Klicken Sie in der MMC-Konsole im Menü **Datei** auf **Snap-In hinzufügen/entfernen**.
- Klicken Sie im Dialogfeld Snap-In hinzufügen/entfernen auf **Hinzufügen**.
- Klicken Sie im Dialogfeld **Eigenständiges Snap-In hinzufügen** auf **Zertifikate**, und klicken Sie dann auf **Hinzufügen**.
- Klicken Sie im **Dialogfeld Zertifikate-Snap-In** auf **Computerkonto**, und klicken Sie dann auf **Fertig stellen**.
- Klicken Sie im Dialogfeld **Eigenständiges Snap-In hinzufügen** auf **Schließen**.
- Klicken Sie im Dialogfeld **Snap-In hinzufügen/entfernen** auf **OK**.
- Erweitern Sie im Dialogfeld **Zertifikate-Snap-In** die Option **Zertifikate**, erweitern Sie **Eigene Zertifikate**, und klicken Sie dann mit der rechten Maustaste auf **Zertifikate**, zeigen Sie auf **Alle Aufgaben** und klicken Sie anschließend auf **Importieren**.

- Klicken Sie mit der rechten Maustaste auf das importierte Zertifikat, zeigen Sie auf **Alle Aufgaben** und klicken Sie dann auf **Privatschlüssel verwalten**.
- Fügen Sie im Dialogfeld **Sicherheit** die Leseberechtigung für das Benutzerkonto hinzu, das vom SQL Server-Dienstkonto verwendet wird.
- Ergänzen Sie die Angaben im **Zertifikatimport-Assistenten**, um dem Computer ein Zertifikat hinzuzufügen und schließen Sie dann die MMC-Konsole.

Weitere Informationen zum Hinzufügen von Zertifikaten zu einem Computer finden Sie in der Windows-Dokumentation.

3.1.2 Konfigurations-Manager ab SQL Server 2019

- Erweitern Sie im SQL Server-Konfigurations-Manager im Konsolenbereich den Knoten **SQL Server-Netzwerkkonfiguration**.
- Klicken Sie mit der rechten Maustaste auf **Protokolle für** <*Instanzname*> und klicken Sie dann auf **Eigenschaften**.
- Wählen Sie die Registerkarte **Zertifikat** und anschließend **Importieren** aus.
- Klicken Sie auf **Durchsuchen** und dann auf die Zertifikatsdatei.
- Klicken Sie auf **Weiter**, um das Zertifikat zu überprüfen.
- Wenn keine Fehler vorliegen, klicken Sie auf **Weiter**, um das Zertifikat in die lokale Instanz zu importieren.

ACHTUNG:

Das SQL Server-Dienstkonto muss über Leseberechtigungen für das Zertifikat verfügen, das zum Erzwingen der Verschlüsselung auf der SQL Server-Instanz verwendet wird. Für ein nicht privilegiertes Dienstkonto müssen dem Zertifikat Leseberechtigungen hinzugefügt werden. Ist dies nicht der Fall kann beim Neustart des SQL Server-Diensts ein Fehler auftreten.

3.2 Installation auf mehreren Servern

Mit SQL Server 2019 (15.x) ist die Zertifikatverwaltung im SQL Server-Konfigurations-Manager integriert. Der SQL Server-Konfigurations-Manager für SQL Server 2019 (15.x) kann mit früheren Versionen von SQL Server verwendet werden (siehe Abschnitt 3.1). Weitere Informationen zum Hinzufügen eines Zertifikats in einer Failovercluster-Konfiguration oder in einer Verfügbarkeitsgruppenkonfiguration finden Sie unter:

Zertifikatverwaltung (SQL Server-Konfigurations-Manager).

Wenn Sie SQL Server 2012 (11.x) bis SQL Server 2017 (14.x) verwenden und der SQL Server-Konfigurations-Manager für SQL Server 2019 (15.x)nicht verfügbar ist:

➡ Rufen Sie folgenden Link auf:

https://docs.microsoft.com/de-de/sql/database-engine/configure-windows/enable-encrypted-connections-to-the-database-engine?view=sql-server-ver15#single-server

➡ Führen Sie unter obigen Link für jeden Server die Schritte im Abschnitt „**So stellen Sie ein Zertifikat auf einem einzelnen Server bereit bzw. installieren Sie ein Zertifikat aus“** aus.

3.3 SQL Server-Konfiguration

➡ Konfigurieren Sie den Server, sodass er verschlüsselte Verbindungen erzwingt.

ACHTUNG:

Das SQL Server-Dienstkonto muss über Leseberechtigungen für das Zertifikat verfügen, das zum Erzwingen der Verschlüsselung auf der SQL Server-Instanz verwendet wird.Für ein nicht privilegiertes Dienstkonto müssen dem Zertifikat Leseberechtigungen hinzugefügt werden.Ist dies nicht der Fall, kann beim Neustart des SQL Server-Diensts ein Fehler auftreten.

➡ Erweitern Sie im **SQL Server-Konfigurations-Manager** den Eintrag **SQL Server-Netzwerkkonfiguration**, klicken Sie mit der rechten Maustaste auf **Protokolle für** *<server instance>* und klicken Sie dann auf **Eigenschaften**.

➡ Wählen Sie auf der Registerkarte **Zertifikat** im Dialogfeld **Eigenschaften** unter **Protokolle für** *<instance name>* das gewünschte Zertifikat aus der Dropdownliste für das Feld **Zertifikat** aus und klicken Sie dann auf **OK**.

➡ Aktivieren Sie auf der Registerkarte **Flags** im Feld **ForceEncryption** die Option **Ja** und klicken Sie dann auf **OK**, um das Dialogfeld zu schließen.

➡ Starten Sie den SQL Server-Dienst neu.

3.4 Zertifikats-Leseberechtigung für SQL Server einrichten

Das SQL Server-Dienstkonto muss über Leseberechtigungen für das Zertifikat verfügen, das zum Erzwingen der Verschlüsselung auf der SQL Server-Instanz verwendet wird. Für ein nicht privilegiertes Dienstkonto müssen dem Zertifikat Leseberechtigungen hinzugefügt werden. Ist dies nicht der Fall, kann beim Neustart des SQL Server-Diensts ein Fehler auftreten.

- Klicken Sie im Menü **Start** auf **Ausführen**, geben Sie in das Feld **Öffnen** den Wert **MMC** ein und klicken Sie dann auf **OK**.
- Klicken Sie in der MMC-Konsole im Menü **Datei** auf **Snap-In hinzufügen/entfernen**.
- Klicken Sie im Dialogfeld **Snap-In hinzufügen/entfernen** auf **Hinzufügen**.
- Klicken Sie im Dialogfeld **Eigenständiges Snap-In hinzufügen** auf **Zertifikate** und klicken Sie dann auf **Hinzufügen**.
- Klicken Sie im **Dialogfeld Zertifikate-Snap-In** auf **Computerkonto** und klicken Sie dann auf **Fertig stellen**.
- Erweitern Sie im Dialogfeld **Zertifikate-Snap-In** die Option **Zertifikate**, erweitern Sie **Eigene Zertifikate** und wählen das gewünscht Zertifikat.
- Klicken Sie mit der rechten Maustaste auf das gewählte Zertifikat, zeigen Sie auf **Alle Aufgaben** und klicken Sie dann auf **Privatschlüssel verwalten**.Fügen Sie im Dialogfeld **Sicherheit** die Leseberechtigung für das Benutzerkonto hinzu, das vom SQL Server-Dienstkonto verwendet wird.

4 Konfiguration des Clients

Der Anmeldeprozess ist immer verschlüsselt. Wenn ForceEncryption (Verschlüsselung erzwingen auf dem SQL-Server, siehe Abschnitt 2.1) auf Ja festgelegt ist, wird jegliche Client/Server-Kommunikation verschlüsselt.
Je nach Art des verwendeten Zertifikats (selbstsigniert, Active Directory Zertifikatsdienste, extern ausgestellt) müssen Clients, die mit Datenbank-Engine verbunden sind, konfiguriert werden, dass sie der Stammzertifizierungsstelle des Serverzertifikats vertrauen. An dieser Stelle ist nicht das Clientseitige Erzwingen der Verschlüsselung gemeint.
Weitere Informationen finden Sie im Abschnitt **„Vorgehensweise: Aktivieren von verschlüsselten Verbindungen zum Datenbank-Engine (SQL Server-Konfigurations-Manager)"** in der SQL Server-Onlinedokumentation, siehe:

Protokolle für MSSQLSERVER-Eigenschaften (Registerkarte Flags)

4.1 Konfiguration mit selbstsigniertem SQL Server-Zertifikat

Bei einem selbstsignierten SQL Server-Zertifikat (Kapitel 2) müssen am Client keine Änderungen vorgenommen werden.

4.2 Windows-Netzwerk mit Domain-Controller

Im Netzwerk mit aktivierten Active-Directory-Zertifikatsdiensten müssen an einem Client, der Mitglied der Domäne ist, keine Änderungen vorgenommen werden. In einer Produktionsumgebung sollten die Active-Directory-Zertifikatsdienste nicht auf dem Domain-Controller laufen, sondern auf einem Member-Server.

4.3 Konfiguration ohne Zertifizierungsdienst

In dem Fall muss auf dem Client das extern ausgestellte Zertifikat installiert werden. Abhängig von den Voraussetzungen muss das Zertifikat auf den Client kopiert und aktiviert werden. Das Zertifikat kann mit Hilfe des **Zertifikate** -Snap-Ins installiert werden.

4.4 Konfiguration mit extern ausgestelltem Zertifikat auf Client

- Konfigurieren Sie den Client, so, dass er verschlüsselte Verbindungen anfordert. Die Konfiguration sollte für 32- und 64-Bit Verbindungen erfolgen.
- Kopieren Sie entweder das Originalzertifikat oder die exportierte Zertifikatsdatei auf den Clientcomputer.
- Installieren Sie auf dem Clientcomputer mithilfe des **Zertifikate** -Snap-Ins entweder das Stammzertifikat oder die exportierte Zertifikatsdatei.
- Klicken Sie im SQL Server-Konfigurations-Manager (auf dem SQL Server) mit der rechten Maustaste auf **SQL Server Native Client-Konfiguration – 64-Bit** und klicken Sie dann auf **Eigenschaften**.
- Klicken Sie auf der Seite **Flags** im Feld **Protokollverschlüsselung erzwingen** auf **Ja**.

- Klicken Sie im SQL Server-Konfigurations-Manager (auf dem SQL Server) mit der rechten Maustaste auf **SQL Server Native Client-Konfiguration (32-Bit)** und klicken Sie dann auf **Eigenschaften**.
- Klicken Sie auf der Seite **Flags** im Feld **Protokollverschlüsselung erzwingen** auf **Ja**.

5 Erstellen eines Zertifikats

5.1 Active-Directory-Zertifikatsdienste einrichten

Anbei zwei Links zum Einrichten von Active-Directory Zertifikatsdiensten:

- https://www.computerweekly.com/de/ratgeber/Active-Directory-Zertifikatsdienste-richtig-einrichten
- CA Checkliste

5.2 Zertifikat anfordern

- https://sid-500.com/2017/03/31/active-directory-zertifikatsdienste-teil-1-installation-einer-enterprise-root-ca/

6 Quellen

- Verwenden von Verschlüsselung ohne Überprüfung
- Verwenden von Verschlüsselung
- Aktivieren von verschlüsselten Verbindungen zur Datenbank-Engine
- Zertifikatverwaltung (SQL Server-Konfigurations-Manager)
- Empfehlungskatalog - Sicherer kommunaler IT-Arbeitsplatz