E-Mails TLS-verschlüsselt

Mit den Freigaben der AHB-Anwendungen vom 07.09.2018 (AHB Zeitwirtschaft Web V6.85, AHB Zutrittskontrolle Web V6.09, AHB Zeitkonto Web V1.85, AHB Connector V1.82) unterstützt der AHB Tornado-Server **TLS-Verschlüsselung** beim E-Mail-Versand.
Der verschlüsselte Versand von E-Mails ist insbesondere für Nachrichten mit personenbezogenen Daten unerlässlich. So kann z.B. das AHB Zeitkonto Web bei entsprechender Konfiguration Informationen über neue oder genehmigte Anträge, über Saldoverletzungen und ähnliches versenden.
Für den Versand von E-Mails aus den AHB-Anwendungen gibt es damit zwei grundsätzliche Möglichkeiten:

- Wenn der **aufnehmende SMTP-Server** TLS-Verschlüsselung anbietet, wird zwingend TLS-Verschlüsselung genutzt. Der aufnehmende SMTP-Server zeigt dies durch das Schlüsselwort STARTTLS an. Zur Konfiguration des aufnehmenden SMTP-Servers und Zertifikatsfragen siehe unten ‚Konfiguration'.
- Bietet der SMTP-Server dagegen **keine TLS-Verschlüsselung** an, werden die Mails wie bisher unverschlüsselt versendet.

TLS-Verschlüsselung

Damit die TLS-Verschlüsselung genutzt werden kann, reicht es allerdings nicht aus, dass der SMTP-Server die Verschlüsselung per STARTTLS anbietet, es müssen zusätzlich folgende Bedingungen erfüllt sein:

1. Der SMTP-Server muss über ein **TLS-Zertifikat** verfügen.
2. Der **Hostname** des SMTP-Servers im Zertifikat muss zum Hostname des SMTP-Servers passen. Ist im Tornado-Server als Hostname des SMTP-Servers z.B. „mail.kunde.de" konfiguriert, dessen TLS-Zertifikat lautet aber auf „smtp.kunde.de", passen diese beiden Namen nicht zusammen. Es scheint dann das falsche Zertifikat hinterlegt zu sein und der Mail-Versand scheitert.
   Einträge wie „localhost" oder die IP-Adresse in der Tornado-Konfiguration sind demnach ungeeignet, denn sie werden höchstwahrscheinlich nicht zu einem Hostnamen in einem Zertifikat passen.
3. Das Zertifikat des SMTP-Servers muss zum Zeitpunkt der Nutzung **gültig** sein.
4. Das Zertifikat des SMTP-Servers muss für den AHB Tornado-Server **verifizierbar** sein, d.h. als vertrauenswürdig eingestuft werden. Dabei gibt es relevante Unterschiede, je nachdem welche Zertifizierungsstelle (Certificate Authority – CA) das Zertifikat ausgestellt hat:

- **Zertifikat einer öffentlichen Zertifizierungsstelle**
  Die Java-Laufzeitumgebung des AHB Tornado-Servers beinhaltet standardmäßig

Informationen über alle öffentlichen (vertrauenswürdigen) Zertifizierungsstellen. Zertifikate dieser CAs lassen sich daher durch Java in der Regel problemlos authentifizieren.
Handelt es sich um eine relativ neue öffentliche Zertifizierungstelle, muss ggf. Java aktualisiert werden, denn dann kann es sein, dass eine ältere Java-Version diese neue öffentliche CA noch nicht kennt.

- **Zertifikat einer eigenen Zertifizierungsstelle**
  Verfügt der Kunde über eine eigene (nicht öffentliche) Zertifizierungsstelle und nutzt ein von dieser eigenen CA ausgestelltes Zertifikat für den verschlüsselten Mail-Versand zwischen Tornado-Server und dem SMTP-Server, muss mit Hilfe der Java-Keytools das Root-Zertifikat dieser eigenen Zertifizierungsstelle (plus evtl. notwendige Zwischenzertifikate) in die vom AHB Tornado Server genutzte Java-Laufzeitumgebung eingebunden werden (s.u. ‚Konfiguration Java cacerts'). Erst dann kann Java ein solches Zertifikat authentifizieren.
- **Selbstzertifiziertes Zertifikat**
  Wird auf dem SMTP-Server ein selbstzertifiziertes Zertifikat verwendet (ohne Nutzung einer CA), muss dieses Zertifikat mit Hilfe der Java Keytools ebenfalls in die Java-Laufzeitumgebung eingebunden werden (s.u. ‚Konfiguration Java cacerts').

Ist auch nur eine dieser vier Bedingungen nicht erfüllt, **scheitert** der Mail-Versand. Es gibt auch kein sogenanntes Fallback auf den unverschlüsselten Mail-Versand – sobald der aufnehmende SMTP-Server die Bereitschaft zum Aufbau einer verschlüsselten TLS-Verbindung anbietet, müssen alle genannten Voraussetzungen erfüllt sein.
Sind diese Bedingungen nicht zu erfüllen, darf der SMTP-Server (zumindest dem AHB Tornado-Server gegenüber) keinen verschlüsselten Versand anbieten.

Konfiguration

Der aufnehmende SMTP-Server, an den die E-Mails der AHB-Anwendungen gesendet werden, kann als „Smart Host" in der Konfiguration des AHB Tornado-Servers eingetragen werden (Parameter „MAILERSmartHost"). Dabei ist zu beachten, dass dieser Hostname im Zertifikat des SMTP-Servers eingetragen ist. Zusätzlich können an dieser Stelle auch Username und Kennwort für eine Anmeldung am SMTP-Server gesetzt werden.
Alternativ kann der Eintrag des „Smart Host" auch leer gelassen werden.
Dann erfolgt lokal vom AHB Tornado-Server aus eine DNS-Anfrage (MX-Record) zur Ermittlung des SMTP-Servers, der für die Aufnahme der Domain in der E-Mail-Adresse des Empfängers zuständig ist. Für diese dann jeweils dynamisch ermittelten SMTP-Server müssen alle oben genannten Bedingungen für einen erfolgreichen verschlüsselten Mail-Versand genauso erfüllt sein.

Konfiguration Java cacerts

Jede Java-Laufzeitumgebung verwaltet eine Liste **vertrauenswürdiger Zertifikate** in der Datei %JAVA_HOME%\lib\security**cacerts**. %JAVA_HOME% steht für den Installationspfad der Java-Laufzeitumgebung, die vom AHB Tornado-Server genutzt wird. Unter ‚Administration – System-Infos…'

kann in jeder AHB-Anwendung dieser Pfad ermittelt werden.

	Java auf dem Server	
	Java-Version	1.8.0_131
	Java-Runtime	Java(TM) SE Runtime Environment / **Version** 1.8.0_131-b11
	Java-Architektur	64 Bit
	Java-Installation	C:\Program Files\Java\jre1.8.0_131

Java Installationspfad

In diese Datei können eigene vertrausenswürde Zertfikate eingefügt werden: selbstzertifizierte oder auch eine Kette vertrausenwürdiger Zertifikate einer eigenen Zertifizierungsstelle (zunächst das root-Zertifikat, dann evtl. weitere Zwischenzertifikate). Ein neues Zertifikat muss in Dateiform auf dem Server vorliegen (Format: X.509 v1, v2 oder v3, binär oder Base64).
Zum Einfügen eines neuen, vertrauenswürdigen Zertifikats in die cacerts kann das Tool ‚keytool.exe' der Java-Installation gemäß folgenden Hinweisen genutzt werden (dabei ist ‚changeit' das Standard-Passwort der cacerts einer Java-Installation). In einer adminsitrativen Eingabeaufforderung auf dem Server:

- cd %JAVA_HOME%
- Für Hilfe zum keytool:
- bin\keytool.exe -help bzw. bin\keytool.exe -COMMAND -help
- Zur Kontrolle eine Liste des bisherigen Inhalts der cacerts:
- bin\keytool.exe -list -keystore „lib\security\cacerts" -storepass changeit
- Hinzufügen eines neuen Zertifikats:
- bin\keytool.exe -importcert -alias ALIAS -file „NEUES_ZERTIFIKAT" -keystore lib\security\cacerts -storepass changeit
- Dabei ist ALIAS ein sinnvoller Name für das neue Zertifikat, so dass es bei späteren Auflistungen der cacerts leicht identifiziert werden kann. NEUES_ZERTIFIKAT ist die komplette Pfadangabe der Datei mit dem neuen Zertifikat.

Nach derartigen Änderungen an der cacerts muss der AHB Tornado-Server neu gestartet werden.

Alternatives Tool

Es gibt Alternativen zur Nutzung der ‚keytool.exe', z.B. kann mit dem interaktiven Open-Source-Tool ‚**Portecle**, (http://portecle.sourceforge.net/) eine komfortablere Ergänzung der cacerts um weitere Zertifikate erreicht werden. Außerdem kann mit diesem Tool ein eigenes Zertifikat, das auf einem Web-Server installiert und über einen https-Zugriff erreichbar ist, heruntergeladen und in die cacaerts importiert werden.

Java Updates

Wenn es notwendig ist, die Java-Laufzeitumgebung mit zusätzlichen Zertifikatsinformationen zu versorgen (siehe oben zur Anpassung der cacerts), müssen diese Aktionen nach einem **Update** der Java-Laufzeitumgebung **erneut** ausgeführt werden, denn die neue Java-Version beinhaltet dann wieder nur die Authentifizierungsdaten zu den öffentlichen Zertifizierungsstellen.