Secure Hash Algorithm (SHA) 256 voor Customer Voice Portal (CVP)

Inhoud

Inleiding

In dit document wordt de procedure beschreven om SHA256 met CVP te gebruiken.

Voorwaarden

Vereisten

Cisco raadt kennis van de volgende onderwerpen aan:

- CVP
- Certificaten

Gebruikte componenten

De informatie in dit document is gebaseerd op CVP 10.5.

De informatie in dit document is gebaseerd op de apparaten in een specifieke laboratoriumomgeving. Alle apparaten die in dit document worden beschreven, hadden een opgeschoonde (standaard)configuratie. Als uw netwerk levend is, zorg er dan voor dat u de mogelijke impact van om het even welke opdracht begrijpt.

Achtergrondinformatie

Vanaf januari 2016 wezen alle webbrowsers SHA1-ondertekende certificaten af. Dit heeft de gevraagde diensten niet correct verricht, tenzij u van SHA1 naar SHA256 overgaat.

Dankzij de recente ontwikkeling van automatiseringsalgoritmen en de explosieve computercapaciteit is SHA1 dag in dag uit zwakker geworden. Dit leidde tot een fundamentele aantastingsbotsing van de SHA1 en een uiteindelijke ondergang.

Configureren

Certificaatuitwisselingsprocedure tussen CVP Operations Console (OAMP):

Over OAMP

Stap 1. ExportOMA-CERT.

c:\Cisco\CVP\jre\bin\keytool.exe -export-v -keystore.keystore-storetype JCEKS -alias oamp_certificate -file oamp_security_76.cer

Stap 2. Kopieer het OAMP-certificaat naar CallServer en importeren.

c:\Cisco\CVP\jre\bin\keytool.exe -import-trustcacerts-keystore.keystore-storetype JCEKS -alias orm_oamp_certificate -file oamp_security_76.cer

On-Call Server

Stap 1. Exporteren van CALLSERVER CERT.

c:\Cisco\CVP\jre\bin\keytool.exe -export -v -keystore.ormkeystore-storetype JCEKS -alias orm_certificaat -file_security_108.cer

Stap 2. Kopieer CALLSERVER CERT naar OAMP en importeren.

c:\Cisco\CVP\jre\bin\keytool.exe -import-trustcacerts-keystore.keystore-storetype JCEKS -alias oamp_orm_certificaat -file_security_108.cer

Stap 3. Exportcertificaat in Call Server-toetsenbord.

C:\Cisco\CVP\conf\security>c:\Cisco\CVP\jre\bin\keytool.exe -import-trustcacerts-keystore.keystore-storetype JCEKS-alias vxml_orm_certificaat -file_security_108.cer

Verifiëren

U kunt valideren als de beveiligde communicatie tussen onderdelen tot stand is gebracht. Navigeren in naar **OAMP Pagina > Apparaatbeheer > <beheerde server> Statistieken**

De instellingen moeten worden weergegeven.

U kunt de toetsencombinatie gebruiken om een verbinding tot stand te brengen indien de beveiliging correct is ingesteld:

Stap 1. **c:\Cisco\CVP\conf\orm_jmx.conf** op OAMP ziet er zo uit:

```
javax.net.debug = all
com.sun.management.jmxremote.ssl.need.client.auth = false
com.sun.management.jmxremote.authenticate = false
com.sun.management.jmxremote.port = 2099
com.sun.management.jmxremote.ssl = true
javax.net.ssl.keyStore=C:\Cisco\CVP\conf\security\.ormkeystore
```

```
javax.net.ssl.keyStorePassword=<local security password>
```

Stap 2. Open console uit opdracht. Gebruik de opdracht:

C:\Cisco\CVP\jre\bin>jconsole.exe -J-Djavax.net.ssl.trustStore=C:\Cisco\CVP\conf\security\.keystore-J-Djavax.net.ssl.trustStorePassword=<oamp security wachtwoord/jconsole client> - J-Djavax.net.ssl.keyStore=C:\Cisco\CVP\conf\security\.keystore-J-Djavax.net.ssl.keyStorePassword=<oamp security wachtwoord/console client> -J-Djavax.net.ssl.keyStoreType=JCEKS-debug -J-Djavax.net.ssl.trustStoreType=JCEKS

Belangrijk in **<managed server-ip>:<secure jmx poort, bijvoorbeeld:2099>** in veld **Remote** Procesveld.

> **Opmerking**: JConsole moet zich zonder aanwijzingen voor de toepassing aansluiten om de beveiligde methode te omzeilen.

Stap 3. Wireshark terwijl de verbindingslijn van de console wordt aangehaald. De opname geeft je het inzicht in de details waarover is onderhandeld terwijl de security handdruk bestaat.

Traces in JMX

De implementatie van JMX maakt gebruik van java.util.logging om sporen te loggen die het fout opleveren. Veel van deze sporen hebben betrekking op onbelichte interne klassen, maar ze kunnen je helpen te begrijpen wat er met je applicatie gebeurt.

De JMX-implementatie bestaat uit twee groepen loggers:

- **`javax.management.\*`**: alle loggers met betrekking tot de JMX API
- **`javax.management.Remote.\*`**: Bloggers die specifiek gerelateerd zijn aan JMX Remote API

U vindt hier een volledigere beschrijving van JMX Loggers.

U kunt de JMX-sporen op twee verschillende manieren activeren:

- Statisch, met het gebruik van een bestand **logging.Properties**
- Dynamisch, met het gebruik van een JMXTracing MBean. In Java SE 6 kunt u dit voor een toepassing doen, zelfs als de JMX-connector niet op de opdrachtregel is ingeschakeld.

Gebruik een bestand logging.Properties

Start uw applicatie met deze vlaggen:

```
java -Djava.util.logging.config.file=<logging.properties> ....
```

waarin **logging.Properties** sporen activeert voor JMX loggers:

```
handlers= java.util.logging.ConsoleHandler
.level=INFO
```

```
java.util.logging.FileHandler.pattern = %h/java%u.log
java.util.logging.FileHandler.limit = 50000
java.util.logging.FileHandler.count = 1
java.util.logging.FileHandler.formatter = java.util.logging.XMLFormatter

java.util.logging.ConsoleHandler.level = FINEST
java.util.logging.ConsoleHandler.formatter = java.util.logging.SimpleFormatter

// Use FINER or FINEST for javax.management.remote.level - FINEST is
// very verbose...
//
javax.management.level=FINEST
javax.management.remote.level=FINER
```