Artículo
Ricardo Paiva · 16 dic, 2019 Lectura de 7 min

Configuración de aplicaciones cliente de Caché para SSL/TLS

¡Hola a tod@s!

Al usar Studio, ODBC o una conexión de terminal a Caché o Ensemble, quizás se haya preguntado cómo podría implementar una conexión segura. Una opción es agregar TLS (también conocido como SSL) a su conexión. Las aplicaciones cliente de Caché (TELNET, ODBC y Studio) todas entienden cómo agregar TLS a la conexión. Tan solo es necesario configurarlas para hacerlo.

Configurar esos clientes es más fácil en la versión 2015.1 y posteriores. A continuación discutiré este nuevo método. Si ya usa el viejo método, este seguirá funcionando, pero le recomiendo considerar pasarse al nuevo.

Contexto

Estas aplicaciones cliente pueden instalarse en una máquina que no tenga la instalación de servidor. No pueden depender de tener acceso a los lugares normales para guardar configuraciones, tales como la base de datos CACHESYS o el archivo cpf. En vez, su configuración sobre qué certificados o protocolos aceptar se almacena en un archivo de texto. Muchos de los ajustes contenidos en este archivo son similares a los ajustes de una configuración SSL/TLS en el portal de gestión.

¿Dónde está el archivo de configuración?

Deberá crear su propio archivo. El instalador del cliente no crea uno por usted.

De forma predeterminada, el archivo de configuración se llama SSLDefs.ini y debería colocarse en el directorio InterSystems\Cache bajo el directorio para archivos de programa comunes de 32 bits. Este directorio se encuentra en la variable de entorno de Windows CommonProgramFiles(x86) en Windows de 64 bits o en CommonProgramFiles en Windows de 32 bits.

Por ejemplo, en Windows 8.1, el archivo predeterminado es:

```
C:\Program Files (x86)\Common Files\InterSystems\Cache\SSLdefs.ini
```

Si desea cambiar esto, deberá indicarle a los ejecutables del cliente dónde encontrar el archivo de configuración. Para hacer esto, defina la variable de entorno ISCSSLconfigurations y configúrela para toda la ruta y el nombre de archivo de su archivo. Puede que necesite permisos de administrador para hacerlo.

¿Qué hay en el archivo de configuración?

El archivo tiene dos tipos de secciones. El primer tipo vincula conexiones con configuraciones TLS. Por ejemplo, puede que le indique a Studio usar la sección llamada "Default Settings" para encontrar sus parámetros TLS al conectarse a development.intersystems.com.

El segundo tipo define los ajustes de TLS a usar para la conexión. Por ejemplo, esto definiría qué Autoridad de Certificación debería esperarse que firme el certificado del servidor. Los ajustes de estas secciones son muy similares a los ajustes de una configuración SSL/TLS en un servidor Caché o Ensemble.

El primer tipo de selección se ve así:

```
[Development Server]
Address=10.100.0.17
Port=1972
TelnetPort=23?
SSLConfig=DefaultSettings?
```

El nombre entre corchetes puede ser lo que usted desee. Sólo está ahí para permitirle hacer un seguimiento más fácil de qué conexión se trata.

Los ajustes de Address (dirección), Port (puerto) y TelnetPort (puerto Telnet) se usan para decidir qué conexiones deberían coincidir con esta sección. Se pueden usar ya sea direcciones IP o nombres DNS para la dirección en clientes 2016.1 o posteriores. Tanto la dirección como ya sea el puerto o el puerto Telnet deben coincidir con la conexión de la aplicación cliente para poder usar la configuración.

El parámetro final (SSLConfig) es el nombre de la configuración de la cual se tomarán los ajustes TLS. Debe coincidir con el nombre de una de las configuraciones del archivo.

El segundo tipo de sección se ve así:

```
[DefaultSettings]
VerifyPeer=2
VerifyHost=1
CAfile=c:\InterSystems\certificates\CAcert.pem
CertFile=c:\InterSystems\certificates\ClientCert.pem
KeyFile=c:\InterSystems\certificates\ClientKey.key
Password=
KeyType=2
Protocols=24
CipherList=ALL:!aNULL:!eNULL:!EXP:!SSLv2
```

El nombre de la sección se lista en la primera línea: [DefaultSettings] y coincide con el nombre listado en el parámetro SSLConfig de la primera sección del ejemplo anterior. Por lo tanto, esta configuración se usará para conexiones al servidor 10.100.0.17 en el puerto 1972 o el puerto 23.

Usar copiar+pegar en el ejemplo anterior a menudo genera caracteres que no se imprimen en su archivo de texto. Por favor, asegúrese de haber quitado cualquier carácter extra, por ejemplo guardando el archivo como solo texto y abriéndolo nuevamente.

Aquí puede ver una descripción de lo que significan los parámetros:

- **VerifyPeer**

Las opciones para esto son 0=ninguna, 1=solicitar y 2=requerir. Requerir es el valor recomendado. Si elige "ninguna", un servidor malicioso podría simular ser el servidor al que usted pretende conectarse. Si elige requerir, deberá ingresar una Autoridad certificadora en la que confíe para que verifique los certificados para el valor CAFile. Esto es el equivalente a la "Verificación de certificado de servidor" en el portal. (Nota: la solicitud no tiene sentido para una configuración de cliente, pero la incluyo aquí para que pueda entender por qué las opciones son 0 y 2.)

- **VerifyHost**

Las opciones para esto son 0=ninguna, 1=requerido. Esta opción verifica que el certificado del servidor lista el nombre de host o la IP a la que ha solicitado conectarse en los campos Subject's Common Name o Subject's Common Name. Este campo no tiene un equivalente en el portal, pero es del mismo tipo de verificación que la

propiedad SSLCheckServerIdentity de la clase %Net.HttpRequest. Solo es configurable si su cliente usa Caché / Ensemble 2018.1 o posterior, o cualquier versión de la plataforma de datos IRIS de InterSystems

- **CAfile**

La ruta al archivo de la Autoridad Certificadora (CA) de confianza. Esto debería ser la CA que firmó el certificado del otro lado (el servidor), no su propio certificado. Debería completar esto si no ha elegido un valor de VerifyPeer de 2. Este es el equivalente de "Archivo que contiene certificado(s) de Autoridad certificadora confiable" en el portal. Los certificados deben estar en formato PEM

- **CertFile**

La ruta a su propio certificado. Esto debería quedar en blanco si su cliente no cuenta con uno. Este es el equivalente de "Archivo que contiene el certificado del cliente" en el portal. Los certificados deben estar en formato PEM

- **KeyFile**

La ruta a la clave privada correspondiente para CertFile. Debe completar esto si tiene un CertFile, o dejarlo en blanco en caso contrario. Este es el equivalente de "Archivo que contiene clave privada asociada" en el portal

- **Contraseña**

La contraseña necesaria para descifrar su clave privada. Esto debe dejarse en blanco si no usa un certificado para este cliente, o si la clave privada del certificado no está cifrada en el disco

- **KeyType**

¿Su clave privada es RSA (2) o DSA (1)? El valor solo es importante para configuraciones que tienen configurado CertFile y KeyFile. Si no está seguro de cuál es, su clave probablemente sea RSA

- **Protocols**

Esta es una representación decimal de valores de bit para las versiones de SSL/TLS soportadas. Las opciones son: 1=SSLv2, 2=SSLv3, 4=TLSv1, 8=TLSv1.1, 16=TLSv1.2. SSLv2 y SSLv3 tienen problemas conocidos y no se recomiendan. Es posible especificar más de una versión agregando números. Por ejemplo, 24 es TLSv1.1 y TLSv1.2. Esto es el equivalente de las casillas "Protocols" en el portal. (Nota: los 8 y 16 bits no están en la versión 2015.1. Si quiere usarlos, necesitará actualizar a 2015.2 o superior)

- **CipherList**

Este es el equivalente de "Enabled ciphersuites" en el portal. Esto controla exactamente qué tipos de cifrado y hashing serán aceptados por este cliente. ALL:!aNULL:!eNULL:!EXP:!SSLv2 es el valor predeterminado para este ajuste en el portal de gestión. Si experimenta problemas con su conexión, probablemente no sea esto. Cambiar esto puede hacer que su conexión sea menos segura, ya que permitirá un cifrado más débil. Puede encontrar más información sobre este valor en el sitio web de openssl

Notas finales

¡Eso es todo lo que necesita hacer! Si crea su archivo y lo coloca en la ubicación conocido, se usará automáticamente si el nombre o la dirección IP y puerto al que se está conectando coinciden con una de las

conexiones enumeradas en el archivo.

Configuración del servidor

Este artículo trata sobre cómo configurar el lado de cliente de su conexión para que use SSL, pero no olvide que el servidor al que se está conectando también debe comprender cómo aceptar SSL. Puede encontrar la documentación sobre la configuración del SuperServer para que use SSL aquí: http://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=...

Y la documentación para configurar el servicio Telnet está aquí: http://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=...

El método $SYSTEM.Security.Users.SetTelnetSSLSetting() le permite controlar si el servidor Telnet permite o requiere el uso de SSL. Está disponible en la versión 2016.1 y posteriores.

Configuración de DSN

No necesita cambiar la DSN para una conexión ODBC siempre que tenga en su archivo de configuración una dirección y puerto de conexión correspondientes. Se usará SSL incluso si se selecciona Contraseña como método de autenticación el DSN. Las opciones Password with SSL/TLS (contraseña con SSL/TLS) y SSL/TLS server name (nombre de servidor SSL/TLS) eran la forma de configurar SSL para ODBC antes de la versión 2015.1.

Enlace a la documentación

La documentación sobre TLS para aplicaciones cliente ahora está disponible en el sitio de documentos de IRIS: https://irisdocs.intersystems.com/irislatest/csp/docbook/DocBook.UI.Page...

#Seguridad #SSL #Caché

URL de fuente:https://es.community.intersystems.com/post/configuraci%C3%B3n-de-aplicaciones-cliente-de-cach%C3%A9-para-ssltls