Artigo
Andre Larsen Barbosa · Jul. 19, 2021 7min de leitura

# Configurando aplicativos cliente Caché para SSL / TLS

Ao usar o Studio, ODBC ou uma conexão de terminal para Caché ou Ensemble, você deve estar se perguntando como proteger a conexão. Uma opção é adicionar TLS (também conhecido como SSL) à sua conexão. Os aplicativos cliente Caché - TELNET, ODBC e Studio - todos sabem como adicionar TLS à conexão. Eles só precisam ser configurados para fazer isso.

Configurar esses clientes é mais fácil em 2015.1 e posteriores. Vou discutir esse novo método. Se você já estiver usando o método antigo e legado, ele continuará funcionando, mas eu recomendo que você considere mudar para o novo.

## Background

Esses aplicativos cliente podem ser instalados em uma máquina que não tenha o servidor instalado. Eles não podem depender de ter acesso aos locais normais para armazenar configurações, como o banco de dados CACHESYS ou o arquivo cpf. Em vez disso, suas configurações para os certificados ou protocolos a serem aceitos são armazenadas em um arquivo de texto. Muitas das configurações neste arquivo são semelhantes às configurações em uma configuração SSL / TLS no portal de gerenciamento.

## Onde está o arquivo de configurações?

Você terá que criar seu próprio arquivo. O instalador do cliente não cria um para você.

Por padrão, o arquivo de configurações é chamado SSLDefs.ini e deve ser colocado no diretório InterSystems \ Cache sob o diretório para arquivos de programas comuns de 32 bits. Este diretório é encontrado na variável de ambiente do Windows * CommonProgramFiles (x86) * no Windows de 64 bits ou * CommonProgramFiles * no Windows de 32 bits.

Por exemplo, no Windows 8.1, o arquivo padrão seria:

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

Se quiser mudar isso, você terá que dizer aos executáveis do cliente onde encontrar o arquivo de configurações. Você pode fazer isso definindo a variável de ambiente * ISCSSLconfigurations * e configurando-a para todo o caminho e nome de arquivo do seu arquivo. Você pode precisar de permissões de administrador para fazer isso.

## O que há no arquivo de configurações?

O arquivo possui dois tipos de seções. O primeiro tipo combina conexões com configurações TLS. Por exemplo, pode dizer ao Studio para usar a seção chamada "Configurações padrão" para encontrar seus parâmetros TLS ao se conectar a development.intersystems.com.

O segundo tipo define as configurações de TLS a serem usadas para a conexão. Por exemplo, eles definiriam por qual Autoridade de Certificação esperar que o certificado do servidor seja assinado. As configurações nestas seções são muito semelhantes às configurações em uma configuração SSL / TLS em um servidor Caché ou Ensemble.

O primeiro tipo de seção se parece com isto:

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

O nome entre colchetes pode ser o que você quiser. Ele está lá apenas para tornar mais fácil para você manter o controle de qual conexão é essa.

As configurações de Endereço, Porta e TelnetPort são usadas para decidir quais conexões devem corresponder a esta seção. Os endereços IP ou nomes DNS podem ser usados para o endereço em clientes 2016.1 ou posteriores. Tanto o endereço quanto a porta ou TelnetPort devem corresponder ao local onde o aplicativo cliente está se conectando para que a configuração seja usada.

O parâmetro final (SSLConfig) é o nome da configuração da qual obter as configurações de TLS. Ele precisa corresponder ao nome de uma das configurações no arquivo.

O segundo tipo de seção tem a seguinte aparência:

```
[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
```

O nome da seção está listado na primeira linha: [DefaultSettings] e corresponde ao nome listado no parâmetro SSLConfig da primeira seção do exemplo acima. Portanto, esta configuração será usada para conexões com o servidor 10.100.0.17 na porta 1972 ou na porta 23.

Usar copiar e colar no exemplo acima geralmente causa caracteres não imprimíveis em seu arquivo de texto. Certifique-se de ter removido quaisquer caracteres extras, por exemplo, salvando o arquivo como somente texto e abrindo-o novamente.

Aqui está uma descrição do que os parâmetros significam:

- **VerifyPeer**

As opções para isso são 0 = nenhum, 1 = solicitação e 2 = exigir. Exigir é o valor recomendado. Se você não escolher nenhum, é possível que um servidor malicioso finja ser o servidor ao qual você pretende se conectar. Se você escolher requerer, precisará preencher uma Autoridade de certificação em que você confia para verificar os certificados para o valor CAFile. Isso é equivalente a "Verificação do certificado do servidor" no portal. (Observação: a solicitação não faz sentido para uma configuração de cliente, mas eu a incluí aqui para que você possa entender por que as opções são 0 e 2.)

- **VerifyHost**

As opções para isso são 0 = nenhum, 1 = obrigatório. Esta opção verifica se o certificado do servidor lista o nome do host ou IP ao qual você pediu para se conectar nos campos Nome comum do assunto ou subjectAlternativeName. Este campo não possui equivalente no portal, mas é o mesmo tipo de verificação da propriedade SSLCheckServerIdentity da classe% Net.HttpRequest. Só é configurável se o seu cliente estiver usando o Caché / Ensemble 2018.1 ou posterior, ou qualquer versão da InterSystems IRIS Data Platform.

- **CAfile**

O caminho para o arquivo de autoridade de certificação (CA) confiável. Deve ser a CA que assinou o certificado do * outro * lado (o servidor), não o seu próprio certificado. Isso deve ser preenchido se você escolheu um valor VerifyPeer de 2. Isso é o equivalente a "Arquivo contendo certificado (s) de autoridade de certificação confiável (s)" no portal. Os certificados devem estar no formato PEM.

- **CertFile**

O caminho para seu próprio certificado. Deve ficar em branco se o seu cliente não tiver um. Equivale a "Arquivo contendo o certificado deste cliente" no portal. Os certificados devem estar no formato PEM.

- **KeyFile**

O caminho para a chave privada correspondente para CertFile. Deve ser preenchido se você tiver um CertFile e em branco se não tiver. Isso é equivalente a "Arquivo contendo chave privada associada" no portal.

- **Password**

A senha necessária para descriptografar sua chave privada. Deve ficar em branco se você não estiver usando um certificado para este cliente ou se a chave privada do certificado não estiver criptografada no disco.

- **KeyType**

A sua chave privada é RSA (2) ou DSA (1)? O valor é relevante apenas para configurações que possuem CertFile e KeyFile definidos. Se você não tiver certeza de qual é, sua chave provavelmente é RSA.

- **Protocols**

Esta é uma representação decimal de valores de bits para as versões de SSL / TLS suportadas. As opções são: 1 = SSLv2, 2 = SSLv3, 4 = TLSv1, 8 = TLSv1.1, 16 = TLSv1.2. SSLv2 e SSLv3 têm problemas conhecidos e não são recomendados. Mais de uma versão pode ser especificada adicionando números. Por exemplo, 24 é TLSv1.1 e TLSv1.2. Isso equivale às caixas de seleção "Protocolos" do portal. (Observação: os 8 e 16 bits não estão em 2015.1. Se quiser usá-los, você precisa atualizar para 2015.2 ou superior.)

- **CipherList**

Isso é o equivalente a "ciphersuites ativados" no portal. Isso controla exatamente quais tipos de criptografia e hashing serão aceitáveis para este cliente. ** ALL:! ANULL:! ENULL:! EXP:! SSLv2 ** é o valor padrão para esta configuração no portal de gerenciamento. Se você está tendo problemas com sua conexão, provavelmente não é isso. Alterar isso pode tornar sua conexão menos segura, permitindo uma criptografia fraca. Você pode encontrar mais informações sobre este valor no site do openssl.

# Final notes

Isso é tudo que você precisa fazer! Se você criar seu arquivo e colocá-lo no local conhecido, ele será usado automaticamente se o nome ou endereço IP e a porta que você está conectando corresponderem a uma das conexões listadas no arquivo.

## Server setup

Este artigo é sobre como configurar o lado do cliente de sua conexão para usar SSL, mas não se esqueça de que o servidor ao qual você está se conectando também precisa entender como aceitar SSL. A documentação sobre como configurar o SuperServer para usar SSL pode ser encontrada aqui:

http://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=GCASssltls#GCASssltlssuperserver

E a documentação de configuração do serviço Telnet está aqui:

http://docs.intersystems.com/latest/csp/docbook/DocBook.UI.Page.cls?KEY=GCASssltls#GCASssltlstelnetsvr

O método $ SYSTEM.Security.Users.SetTelnetSSLSetting () permite controlar se o servidor Telnet permite ou exige o uso de SSL. Ele está disponível em 2016.1 e posteriores.

## DSN configuration

Você não precisa alterar o DSN para uma conexão ODBC, contanto que tenha um endereço de conexão e uma porta correspondentes em seu arquivo de configurações. SSL será usado mesmo se a senha for selecionada para o método de autenticação no DSN. As opções **Password with SSL / TLS** e ** SSL / TLS server name ** eram para o estilo pré-2015.1 de configuração de SSL para ODBC.

## Documentation link

A documentação sobre TLS para aplicativos cliente está agora disponível no site de documentos IRIS:

https://irisdocs.intersystems.com/irislatest/csp/docbook/DocBook.UI.Page.cls?KEY=GCASssltls#GCASssltlswindotinifile

#Segurança #SSL #Caché

**URL de origem:**https://pt.community.intersystems.com/post/configurando-aplicativos-cliente-cach%C3%A9-para-ssl-tls