記事
Shintaro Kaminaka · 2020年4月28日 9m read

SSL/TLS用Cachéクライアントアプリケーションの構成

CachéまたはEnsembleへの接続にStudio、ODBC、またはターミナルを使用している場合、その接続をどのように保護すれば良いのか疑問に思うかもしれません。 選択肢の一つに、TLS（別名SSL）を接続に追加することが挙げられます。 Cachéクライアントアプリケーション（TELNET、ODBC、Studio）にはすべて、TLSを接続に追加する機能があります。 あとは単純にその構成を行うだけです。

2015.1以降はこれらのクライアントを簡単に設定できるようになりました。 ここでは、その新しい方法について説明します。 既に古い方法を使用している場合も引き続き機能しますが、新しい方法への切り替えを検討することをお勧めします。

背景

これらのクライアントアプリケーションは、サーバーがインストールされていないマシンにインストールできます。 ただし、CACHESYSデータベースやcpfファイルなど、設定を保存する通常の場所へのアクセスに依存することはできません。 その代わり、受け付ける証明書やプロトコルの設定はテキストファイルに保存されます。
このファイルの設定の多くは、管理ポータルのSSL/TLS構成の設定に似ています。

設定ファイルはどこにありますか？

独自のファイルを作成する必要があります。 クライアントインストーラーは設定ファイルを作成しません。

設定ファイルはデフォルトでSSLDefs.iniという名前になっており、32ビット共通プログラムファイルのディレクトリ配下のInterSystems\Cacheディレクトリに配置する必要があります。 このディレクトリは、CommonProgramFiles(x86)（64ビット版Windowsの場合）またはCommonProgramFiles（32ビット版Windowsの場合）というWindows環境変数で確認できます。

例えばWindows 8.1の場合、デフォルトのファイルは次のようになります。

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

設定ファイルの場所を変更する場合は、クライアントの実行可能ファイルに設定ファイルの場所を伝える必要があります。 そのためには、環境変数ISCSSLconfigurationsを定義し、それを使用するファイルの完全なパスとファイル名に設定する必要があります。 この操作には管理者権限が必要な場合があります。

設定ファイルには何が記載されていますか？

ファイルには2つのセクションがあります。 最初のセクションは、接続名とTLS構成に関する設定です。
例えば、development.intersystems.com に接続する際に「DefaultSettings」というセクションを使用してTLSパラメーターを検出するようにStudioに指示します。

2番目のセクションは、接続に使用するTLS設定を定義するものです。 例えば、サーバーの証明書を署名する認証局を定義します。 これらのセクションの設定は、CachéまたはEnsembleサーバーのSSL/TLS構成の設定に非常に似ています。

最初のセクションは次のようになります。

```
[Development Server]

Address=10.100.0.17

Port=1972

TelnetPort=23?

SSLConfig=DefaultSettings?
```

カッコ内の名前は任意に設定できます。
この名前は、どの接続であるかを把握しやすくするためだけにあります。

Address、Port、およびTelnetPort設定を使用して、このセクションに一致する接続を決定します。
2016.1以降のクライアントのアドレスには、IPアドレスかDNS名を使用できます。 構成を使用するにはAddress、およびPortかTelnetPortの両方がクライアントアプリケーションの接続先と一致する必要があります。

最後のパラメーター（SSLConfig）は、TLS設定を取得する構成の名前です。
ファイル内の構成のいずれかの名前と一致する必要があります。

セクションの2番目のタイプは次のようになります。

```
[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
```

セクションの名前は [DefaultSettings] のように最初の行に記述されており、上記の最初のセクションの例にあるSSLConfigパラメーターに記述されている名前と一致しています。
したがって、この構成はサーバー10.100.0.17のポート1972またはポート23への接続に使用されます。

上記の例でコピーアンドペーストを使用すると、多くの場合はテキストファイルに印刷不可能な文字が挿入されてしまいます。
ファイルをテキストのみで保存し、再度開くなどで余計な文字を確実に削除するようお願いします。

各パラメーターの意味は次のとおりです。

- VerifyPeer

このオプションは、0=なし、1=要求、2=必須です。 必須（2）が推奨値です。
なし（0）を選択した場合、悪意のあるサーバーによって接続先サーバーが偽装される可能性があります。 要求（1）を選択した場合、信頼する認証局を入力してCAFileの値に指定された証明書を検証する必要があります。

これは、ポータルの「サーバー証明書の検証」に相当するものです。 （注意：要求はクライアント構成の場合は意味がありませんが、オプションが0と2である理由を理解していただくためにここに含めています。）

- VerifyHost

このオプションは、0=なし、1=必須です。 このオプションは、サーバーの証明書のサブジェクトのコモンネームまたはsubjectAlternativeNameフィールドに、接続するように要求したホスト名またはIPがサーバーの証明書に記述されているかどうかをチェックします。 ポータルにはこのフィールドと同等のフィールドはありませんが、 %Net.HttpRequestクラスのSSLCheckServerIdentityプロパティと同じタイプのチェックを行います。 クライアントがCaché / Ensemble 2018.1以降、またはInterSystems IRIS
Data Platformの任意のバージョンを使用している場合にのみ構成可能です。

- CAfile

信頼できる認証局（CA）ファイルへのパスです。
自分が所有する証明書ではなく、相手側（サーバー）の証明書に署名したCAを指定しなければなりません。
VerifyPeerの値に2を選択した場合は値を入力しなければなりません。 これは、ポータルの「信頼できる認証局の証明書を含むファイル」に相当するものです。 証明書はPEM形式である必要があります。

- CertFile

所有する証明書へのパスです。 クライアントが証明書を持たない場合は空白になります。 これは、ポータルの「クライアント証明書を含むファイル」に相当するものです。 証明書はPEM形式である必要があります。

- KeyFile

CertFileに対応する秘密鍵へのパスです。 CertFileがある場合は入力し、ない場合は空白にする必要があります。
これは、ポータルの「関連する秘密鍵を含むファイル」に相当するものです。

- Password

秘密鍵を復号化するために必要なパスワードです。 このクライアントに証明書を使用していない場合、または証明書の秘密鍵がディスク上で暗号化されていない場合は空白になります。

- KeyType

秘密鍵はRSA (2) またはDSA (1) ですか？ この値は、CertFileとKeyFileが設定されている構成にのみ関係します。
所有している鍵がどちらか分からない場合はRSAの可能性が高いと思われます。

- Protocols

これは、サポートされているSSL/TLSバージョンのビット値を10進数で表した値です。
オプションは、1=SSLv2、2=SSLv3、4=TLSv1、8=TLSv1.1、16=TLSv1.2です。
SSLv2とSSLv3には既知の問題があるため、お勧めしません。
数字を足すと、複数のバージョンを指定できます。 例えば、24はTLSv1.1とTLSv1.2を表します。
これは、ポータルの「プロトコル」チェックボックスに相当するものです。
（注意：2015.1には8ビットと16ビットはありません。
これらを使用する場合は、2015.2以降にアップグレードする必要があります。）

- CipherList

これは、ポータルの「有効な暗号スイート」に相当するものです。 このオプションは、このクライアントで受け入れられる暗号化とハッシュのタイプを正確に制御します。 ALL:!aNULL:!eNULL:!EXP:!SSLv2 が管理ポータルのこの設定のデフォルト値です。 接続に問題がある場合はおそらくこの値になっていません。
このオプションを変更すると、弱い暗号化が受け入れられて接続の安全性が低下する可能性があります。
この値に関する詳細は、OpenSSLのウェブサイトを参照してください。

最後の補足

やるべきことは以上です！ ファイルを作成して既知の場所に配置すると、接続先の名前かIPアドレスとポートがファイルに記述されている接続のいずれかと一致する場合にそのファイルが自動的に使用されます。

サーバーのセットアップ

この記事ではクライアント側の接続でSSLを使用するように構成する方法について説明していますが、接続先のサーバーもSSLの受け入れ方法を理解している必要があることを忘れないでください。
SuperServerでSSLを使用する設定に関するドキュメントは次の場所にあります。

http://docs.intersystems.com/latestj/csp/docbook/DocBook.UI.Page.cls?KEY=...

また、Telnetサービスの構成に関するドキュメントは次の場所にあります。

http://docs.intersystems.com/latestj/csp/docbook/DocBook.UI.Page.cls?KEY=...

$SYSTEM.Security.Users.SetTelnetSSLSetting() メソッドを使用すると、TelnetサーバーがSSLの使用を許可または要求するかどうかを制御できます。 2016.1以降で使用可能です。

DSNの構成

設定ファイルに対応する接続先アドレスとポートが一致していれば、ODBC接続のDSNを変更する必要はありません。 SSLはDSNの認証方法にパスワードが選択されている場合でも使用されます。 パスワードとSSL/TLSおよびSSL/TLS Server Nameオプションは、ODBCにSSLを設定する2015.1以前のスタイル用のものでした。

ドキュメントのリンク

クライアントアプリケーション用のTLSに関するドキュメントは、IRISのドキュメントサイトから入手できます。

https://irisdocs.intersystems.com/irislatestj/csp/docbook/DocBook.UI.Page...

#SSL #セキュリティ #Caché

ソースURL:
https://jp.community.intersystems.com/post/ssltls%E7%94%A8cach%C3%A9%E3%82%AF%E3%83%A9%E3%82%A4%E3%82%A2%E3%83%B3%E3%83%88%E3%82%A2%E3%83%97%E3%83%AA%E3%82%B1%E3%83%BC%E3%82%B7%E3%83%A7%E3%83%B3%E3%81%AE%E6%A7%8B%E6%88%90