IBM i
バージョン 7.2

セキュリティー SSL (Secure Sockets Layer)

IBM i
バージョン 7.2

セキュリティー SSL (Secure Sockets Layer)

ご注意

本書および本書で紹介する製品をご使用になる前に、 41 ページの『特記事項』に記載されている情報をお読みください。

本製品およびオプションに付属の電源コードは、他の電気機器で使用しないでください。

本書にはライセンス内部コードについての参照が含まれている場合があります。ライセンス内部コードは機械コードであり、IBM 機械コードのご使用条件に基づいて使用権を許諾するものです。

お客様の環境によっては、資料中の円記号がバックスラッシュと表示されたり、バックスラッシュが円記号と表示されたりする場合があります。

原典： IBM i
Version 7.2
Security
Secure Sockets Layer

発行： 日本アイ・ビー・エム株式会社

担当： トランスレーション・サービス・センター

第1刷 2014.4

目次

Secure Sockets Layer (SSL)

このトピックでは、ご使用のサーバーで Secure Sockets Layer (SSL) を使用する方法について説明します。

Secure Sockets Layer (SSL) は、インターネットのように保護されていないネットワーク上でアプリケーションがセキュアな通信セッションを実行できるようにするための業界標準です。

IBM i 7.2 の新機能

Secure Sockets Layer の新規の情報または大幅に変更された情報に関して以下を参照してください。

- Transport Layer Security (TLS) プロトコルのサポートが追加されました。TLS バージョン 1.2 および TLS バージョン 1.1 がサポートされます。
- 5 ページの『システム SSL プロパティー』 に新規 SSL プロトコルと暗号スイートが追加され、署名アルゴリズムとハンドシェーク・ネゴシエーションに新規プロパティーが導入されました。
- セキュア環境に複数の証明書を構成する機能が追加されました。 複数の証明書の選択機能により、セキュア環境で、楕円曲線ディジタル署名アルゴリズム (ECDSA) 証明書を有効にして、同時に RSA証明書も引き続き使用できるようにすることが可能になります。
- システム SSL 属性の一部に 13 ページの『DCM アプリケーション定義』へのサポートが追加されました。
- 18 ページの『Online Certificate Status Protocol』向けのクライアントのサポートが追加されました。

新規箇所または変更箇所を見つける方法

技術的変更が加えられた箇所を確認できるように、Information Center では以下のものを使用しています。

- » 記号は、新規の情報または変更された情報の開始点を示します。
- « 記号は、新規の情報または変更された情報の終了を示します。

PDF ファイルでは、新規および変更された情報の左マージンに、リビジョン・バー (|) が表示されることがあります。

このリリースの新機能または変更に関する他の情報を見つけるには、『プログラム資料説明書』を参照してください。

SSL の PDF ファイル

この情報の PDF ファイルを表示および印刷することができます。

この文書の PDF 版を表示またはダウンロードするには、「Secure Sockets Layer (SSL)」を選択します。

PDF ファイルの保存

表示用または印刷用の PDF ファイルをワークステーションに保存するには、次のようにします。

1. ご使用のブラウザーで PDF のリンクを右クリックする。
2. ローカルに PDF を保存するオプションをクリックする。

3. PDF を保存したいディレクトリーに進む。
4. 「**保存**」をクリックする。

Adobe Reader のダウンロード

これらの PDF を表示または印刷するには、Adobe Reader がシステムにインストールされている必要があります。Adobe Reader は、Adobe の Web サイト (www.adobe.com/products/acrobat/readstep.html)  から無償でダウンロードすることができます。

SSL 概念

SSL 概念は補足情報であり、Secure Sockets Layer (SSL) プロトコルを構成する基本的な構築ブロックについて説明します。

SSL プロトコルを使用することによって、クライアントとサーバー・アプリケーション間でセキュアな接続を確立して、通信セッションの一方のエンドポイントまたは両方のエンドポイントを認証できるようになります。SSL は、クライアントとサーバー・アプリケーション間でやり取りするデータのプライバシーと保全性も維持します。

SSL の機能

SSL は、実際は 2 つのプロトコルからなっています。つまり、レコード・プロトコルとハンドシェーク・プロトコルです。レコード・プロトコルは、SSL セッションの 2 つのエンドポイント間のデータの流れを制御します。

ハンドシェーク・プロトコルは、SSL セッションの一方のエンドポイントまたは両方のエンドポイントを認証し、その SSL セッション用データの暗号化や暗号化解除に使用する鍵のセットを生成する固有な対称鍵を 1 つ設定します。SSL は、非対称暗号、ディジタル証明書、および SSL ハンドシェーク・フローを使用して、SSL セッションの一方のエンドポイントまたは両方のエンドポイントを認証します。通常 SSL はサーバーを認証しますが、オプションでクライアントを認証します。認証局によって発行されるディジタル証明書は、各エンドポイントに割り当てることも、または接続の各エンドポイントで SSL を使用するアプリケーションに割り当てることもできます。

ディジタル証明書は、公開鍵と、トラステッド認証局 (CA) がディジタル署名した識別情報からなっています。各公開鍵には、秘密鍵が 1 つずつ関連付けられています。秘密鍵は、証明書と一緒に、またはその一部として保管されることはありません。サーバー認証の場合もクライアント認証の場合も、認証されるエンドポイントは、ディジタル証明書に含まれている公開鍵に関連付けられた秘密鍵にアクセスできることを証明しなければなりません。

SSL ハンドシェークは、公開鍵と秘密鍵を使用する暗号操作のために、パフォーマンス集約型の操作になってしまいます。2 つのエンドポイント間で最初に SSL セッションが確立されたときに、これらの 2 つのエンドポイントとアプリケーションに関する SSL セッション情報をセキュアなメモリーにキャッシュすることで、後続の SSL セッションを迅速に使用可能にすることができます。SSL セッションが再開されると、2 つのエンドポイントはハンドシェーク・フローを簡略化して、それぞれのエンドポイントが固有の情報に対するアクセス権を持っていることを、公開鍵や秘密鍵を使用することなく認証します。両方のエンドポイントがこの固有の情報にアクセスできることを証明できた場合は、次に、新しい対称鍵が設定され、SSL セッションが「再開」されます。TLS バージョン 1.2、1.1、1.0 と SSL バージョン 3.0 のセッションでは、キャッシュに入れられた情報が、24 時間を超えてセキュア・メモリーに残っていることはありません。暗号化ハードウェアを使用して、メイン CPU に対する SSL ハンドシェークのパフォーマンスの影響を最小限にすることができます。

関連情報:

ディジタル証明書の概念

暗号化ハードウェア

サポートされている SSL および Transport Layer Security プロトコル

このトピックでは、IBM® i インプリメンテーションがサポートする Secure Sockets Layer (SSL) および Transport Layer Security (TLS) プロトコルのバージョンについて説明します。

いくつかのバージョンの SSL プロトコルが定義されています。IBM i インプリメンテーションは、以下のバージョンの SSL プロトコルおよび TLS プロトコルをサポートします。

- TLS バージョン 1.2
- TLS バージョン 1.1
- TLS バージョン 1.0
- SSL バージョン 3.0
- SSL バージョン 2.0

注:

1. 同時に複数のプロトコルを指定すると、互換モードになります。互換とは、可能な場合は一番上位に指定されているプロトコルがネゴシエーションされ、それが不可能な場合は次に上位に指定されているプロトコルがネゴシエーションされることを意味します。指定されたプロトコルのいずれもネゴシエーションできない場合は、SSL ハンドシェークは失敗します。
2. 互換モードでは、使用可能な最上位と最下位のプロトコルの間のすべてのプロトコルを指定することを推奨します。TLS バージョン 1.2 と TLS バージョン 1.0 が指定されていて、 TLS バージョン 1.1 が指定されていないと、予測不能な結果となる場合があります。

TLS バージョン 1.2 と TLS バージョン 1.1 の対比

最新の業界標準 SSL プロトコルは、Transport Layer Security (TLS) バージョン 1.2 です。その仕様は、Internet Engineering Task Force (IETF) により RFC 5246 (「The TLS Protocol Version 1.2」) で定義されています。

TLS バージョン 1.2 では TLS バージョン 1.1 と比べて以下の機能拡張があります。

- TLSv1.2 でネゴシエーションされるすべての暗号は、最低限 SHA256 を使用する必要があります。名前に SHA(1) を持つ既存の暗号は SHA256 を使用します。
- デジタル署名された要素にある MD5/SHA-1 の組み合わせは、単一のハッシュで置き換えられます。これで、署名された要素に、使用するハッシュ・アルゴリズムを明示的に指定するフィールドが含まれるようになります。
- 拡張サポートは別個に定義されるのではなく、RFC にマージされます。
- DES 暗号は許可されません。これは、この暗号スイートが TLSv1.2 でネゴシエーションできないことを意味します。
  - 09 = *RSA_DES_CBC_SHA

TLS バージョン 1.1 と TLS バージョン 1.0 の対比

2 番目に新しい業界標準 SSL プロトコルは、Transport Layer Security (TLS) バージョン 1.1 です。その仕様は、Internet Engineering Task Force (IETF) により RFC 4346 (「The TLS Protocol Version 1.1」) で定義されています。

TLS バージョン 1.1 では TLS バージョン 1.0 と比べて以下の機能拡張があります。

- 暗黙的な初期設定ベクトル (IV) は、明示的な IV に置き換えられ、暗号化ブロック・チェーン (CBC) の攻撃から保護します。明示的な IV により、AES 暗号および DES 暗号の内部処理が変更されます。
- エクスポート暗号は許可されません。これは、これら 2 つの現在サポートされている暗号スイートが TLSv1.1 でネゴシエーションできないことを意味します。
  - 03 = *RSA_EXPORT_RC4_40_MD5
  - 06 = *RSA_EXPORT_RC2_CBC_40_MD5
- 各種の内部改善。詳細は RFC 4346 を参照してください。

TLS バージョン 1.0 と SSL バージョン 3.0 の対比

SSL バージョン 3.0 に基づく最初の業界標準 SSL プロトコルが、Transport Layer Security (TLS) バージョン 1.0 でした。その仕様は、Internet Engineering Task Force (IETF) により RFC 2246、『*The TLS Protocol*』に定義されています。

TLS の主要な目標は、SSL をよりセキュアにし、このプロトコルの仕様をより正確かつ完全にすることです。TLS は、SSL バージョン 3.0 に対して以下のような拡張を行っています。

- よりセキュアな MAC アルゴリズム
- より細分化されたアラート
- 「グレー・エリア」仕様のより明確な定義

TLS では、以下のようなセキュリティーの改善を行っています。

- **メッセージ認証のための鍵付きハッシュ化** TLS はメッセージ認証コード用の鍵付きハッシュ化 (HMAC) を使用して、インターネットのようなオープン・ネットワークを経由して移動する間にレコードを変更できないようにします。 SSL バージョン 3.0 も鍵付きメッセージ認証を提供しますが、SSL バージョン 3.0 が使用する MAC (Message Authentication Code (メッセージ確認コード)) よりも、HMAC の方がよりセキュアです。
- **Enhanced Pseudorandom Function (PRF)** PRF は、鍵データを生成します。TLS では、PRF は HMAC で定義されます。PRF は、そのセキュリティーを保証する 2 つのハッシュ・アルゴリズムを使用します。いずれかのアルゴリズムが露出した場合は、2 番目のアルゴリズムが露出しない限り、そのデータがセキュアな状態を持続します。
- **終了メッセージ検査の改善** TLS バージョン 1.0 と SSL バージョン 3.0 はどちらも、交換されたメッセージが変更されなかったことを認証する終了メッセージを両方のエンドポイントに提供します。ただし、TLS の場合は、この終了メッセージは PRF 値および HMAC 値に基づいて作成されるので、SSL バージョン 3.0 よりもセキュアです。
- **一貫性のある証明書処理** SSL バージョン 3.0 と異なり、TLS は、TLS インプリメンテーション間で交換する必要のある証明書のタイプを指定します。
- **特定のアラート・メッセージ** TLS は、より具体的な内容の追加のアラートを提供して、いずれかのセッション・エンドポイントで検出された問題を指摘します。TLS は、特定のアラートをいつ送信するかについても文書化します。

SSL バージョン 3.0 と SSL バージョン 2.0

SSL バージョン 3.0 は、SSL バージョン 2.0 とは大きく異なるプロトコルです。この両者の大きな違いは、以下のとおりです。

- SSL バージョン 3.0 のハンドシェーク・プロトコル・フローは、SSL バージョン 2.0 のフローと異なっています。

- SSL バージョン 3.0 には、いくつかのタイミング攻撃向けの修正と SHA-1 ハッシュ・アルゴリズムが組み込まれています。SHA-1 ハッシュ・アルゴリズムは、MD5 ハッシュ・アルゴリズムよりもセキュアであると考えられます。SHA-1 によって、MD5 の代わりに SHA-1 を使用する追加の暗号スイートを SSL バージョン 3.0 がサポートできるようになります。
- SSL バージョン 3.0 プロトコルは、SSL ハンドシェーク処理中に man-in-the-middle (MITM) (中継) アタックの発生を抑えます。SSL バージョン 2.0 では、まれに MITM アタックにより暗号化仕様が弱められる可能性がありました。暗号化が弱まると、無許可の人に SSL セッション鍵を壊す機会を与える可能性があります。

関連情報:

RFC 5246: 「The Transport Layer Security (TLS) Protocol Version 1.2)」

RFC 4346: 「The Transport Layer Security (TLS) Protocol Version 1.1)」

RFC 2246: 「The TLS Protocol Version 1.0)」

システム SSL

システム SSL は、SSL/TLS プロトコルを使用して TCP/IP 通信を保護するために、IBM i Licensed Internal Code (LIC) で提供される一般的なサービスのセットです。システム SSL はオペレーティング・システム、および追加のパフォーマンスおよびセキュリティーを特別に供給するソケット・コードと密結合しています。

アプリケーション開発者は、以下のプログラミング・インターフェースおよび JSSE インプリメンテーションからシステム SSL にアクセスすることができます。

- Global Security Kit (GSKit) API
  - これらの ILE C API は他の ILE 言語からアクセス可能です
- 統合 IBM i SSL_ API
  - これらの ILE C API は他の ILE 言語からアクセス可能です
  - この API セットの使用は推奨されていません。推奨されている C インターフェースは GSKit です。
- 統合 IBM i JSSE インプリメンテーション
  - IBM i JSSE インプリメンテーションは JDK 1.6、JDK 7、および JDK 8 で使用可能です。

IBM、IBM ビジネス・パートナー、独立系ソフトウェア・ベンダー (ISV)、または上記にリストされた 3 つのシステム SSL インターフェースの 1 つを使用するお客様によって作成される SSL アプリケーションは、システム SSL を使用します。例えば、FTP および Telnet は、システム SSL を使用する IBM アプリケーションです。すべての SSL がシステム SSL 使用の IBM i 上でのアプリケーションの実行を可能にしたわけではありません。

システム SSL プロパティー

システム SSL 属性のサブセットでは、そのプロパティーをシステム・レベルで変更することができます。この属性のサブセットは、システム SSL プロパティーと呼ばれます。

システム SSL には、セキュア環境やセキュア・セッションの作成方法を決定する多くの属性があります。アプリケーションが作成される際に、設計担当者はそれぞれの属性の値を選択します。明示的に属性値を固定値に設定するように選択が行われる場合もあります。一部の属性については、設計担当者は、アプリケーション管理者が属性値を制御できるようにするユーザー・インターフェースを提供します。大多数の属性で

は、設計担当者は、属性を変更するコードを持たないようにして、属性のシステム SSL デフォルト値を使用します。アプリケーションのコンパイル後に新規のシステム SSL 属性が追加される場合も、常にデフォルト属性値が使用されます。

すべての属性と同様、システム SSL プロパティーはサポートされる値とデフォルト値によって制限されます。サポートされる値は、属性の機能においてシステム SSL オプションを制限します。デフォルト値は、設計担当者が明示的に属性を設定しない場合の動作を決定します。

以下のシステム SSL プロパティーでは、デフォルト値、サポートされる値、あるいはその両方を変更することができます。システム値またはシステム保守ツール (SST) 拡張分析コマンド SSLCONFIG を指定された通りに使用します。

関連概念:

24 ページの『SSLCONFIG マクロ』

SSLCONFIG マクロにより、システム全体のシステム SSL デフォルト・プロパティー を表示および変更することができます。

関連情報:

SSL システム値: QSSLPCL

SSL システム値: QSSLCSLCTL

SSL システム値: QSSLCSL

SSL プロトコル:

システム SSL には、マルチプロトコルをサポートするインフラストラクチャーがあります。

以下のプロトコルはシステム SSL によるサポートが可能です。

- Transport Layer Security バージョン 1.2 プロトコル (TLSv1.2)
- Transport Layer Security バージョン 1.1 プロトコル (TLSv1.1)
- Transport Layer Security バージョン 1.0 プロトコル (TLSv1.0)
- Secure Sockets Layer バージョン 3.0 プロトコル (SSLv3)
- Secure Sockets Layer バージョン 2.0 プロトコル (SSLv2)
  - TLSv1.2 がサポートされている場合は SSLv2 を使用することはできません。

出荷時の SSL サポート・プロトコル

システム SSL は以下のサポート・プロトコルで出荷されます。

- Transport Layer Security バージョン 1.0 プロトコル (TLSv1.0)
- Transport Layer Security バージョン 1.1 プロトコル (TLSv1.1)
- Transport Layer Security バージョン 1.2 プロトコル (TLSv1.2)

注: SSLv3 および SSLv2 は、出荷時にはシステム SSL で使用不可となっています。 QSSLPCL システム値は、任意のプロトコルを使用不可または使用可能に設定するために使用できます。

出荷時の SSL デフォルト・プロトコル

以下のデフォルト・プロトコルは、アプリケーションによって要求されたときシステム SSL によって使用されます。

- Transport Layer Security バージョン 1.0 プロトコル (TLSv1)

| • Transport Layer Security バージョン 1.1 プロトコル (TLSv1.1)
| • Transport Layer Security バージョン 1.2 プロトコル (TLSv1.2)

| **注:** SSLv3 が管理者によってサポート・プロトコル・リストに追加される場合、デフォルトのプロトコル
| に追加されます。ただし、SSLv2 が管理者によってサポート・プロトコル・リストに追加される場合、デ
| フォルト・プロトコルには追加されません。サポート・プロトコル・リストからデフォルト・プロトコルを
| 除去した場合、デフォルト・プロトコル・リストからも除去されます。

関連情報:

SSL システム値: QSSLPCL

SSL 暗号スイート:

システム SSL には、複数の暗号スイートをサポートするインフラストラクチャーがあります。

各プログラミング・インターフェースには、異なる方法で暗号スイートが指定されます。システム値フォーマットで表示されている以下の暗号スイートは、システム SSL によってサポートすることができます。

| • *RSA_AES_128_GCM_SHA256
| • *RSA_AES_256_GCM_SHA384
| • *ECDHE_ECDSA_NULL_SHA
| • *ECDHE_ECDSA_RC4_128_SHA
| • *ECDHE_ECDSA_3DES_EDE_CBC_SHA
| • *ECDHE_RSA_NULL_SHA
| • *ECDHE_RSA_RC4_128_SHA
| • *ECDHE_RSA_3DES_EDE_CBC_SHA
| • *ECDHE_ECDSA_AES_128_CBC_SHA256
| • *ECDHE_ECDSA_AES_256_CBC_SHA384
| • *ECDHE_RSA_AES_128_CBC_SHA256
| • *ECDHE_RSA_AES_256_CBC_SHA384
| • *ECDHE_ECDSA_AES_128_GCM_SHA256
| • *ECDHE_ECDSA_AES_256_GCM_SHA384
| • *ECDHE_RSA_AES_128_GCM_SHA256
| • *ECDHE_RSA_AES_256_GCM_SHA384
- *RSA_AES_128_CBC_SHA256
- *RSA_AES_256_CBC_SHA256
- *RSA_NULL_SHA256
- *RSA_NULL_MD5
- *RSA_NULL_SHA
- *RSA_EXPORT_RC4_40_MD5
- *RSA_RC4_128_MD5
- *RSA_RC4_128_SHA
- *RSA_EXPORT_RC2_CBC_40_MD5
- *RSA_DES_CBC_SHA
- *RSA_3DES_EDE_CBC_SHA

- *RSA_AES_128_CBC_SHA
- *RSA_AES_256_CBC_SHA
- *RSA_RC2_CBC_128_MD5
- *RSA_DES_CBC_MD5
- *RSA_3DES_EDE_CBC_MD5

出荷時の SSL サポート暗号化仕様リスト

暗号化仕様リストは暗号スイートのリストを含みます。システム SSL は 29 の暗号スイートがサポートされた状態で出荷されます。管理者はシステム値 QSSLCSL および QSSLCSLCTL を使用して、システム SSL によってサポートされる暗号を制御できます。暗号スイートは、必要とする SSL プロトコルがサポートされていなければ、サポートすることはできません。

以下の暗号スイートはシステム SSL によってサポートされた状態で出荷されています。

- *ECDHE_ECDSA_AES_128_CBC_SHA256
- *ECDHE_ECDSA_AES_256_CBC_SHA384
- *ECDHE_ECDSA_AES_128_GCM_SHA256
- *ECDHE_ECDSA_AES_256_GCM_SHA384
- *RSA_AES_128_CBC_SHA256
- *RSA_AES_128_CBC_SHA
- *RSA_AES_256_CBC_SHA256
- *RSA_AES_256_CBC_SHA
- *RSA_AES_128_GCM_SHA256
- *RSA_AES_256_GCM_SHA384
- *ECDHE_RSA_AES_128_CBC_SHA256
- *ECDHE_RSA_AES_256_CBC_SHA384
- *ECDHE_RSA_AES_128_GCM_SHA256
- *ECDHE_RSA_AES_256_GCM_SHA384
- *ECDHE_ECDSA_3DES_EDE_CBC_SHA
- *ECDHE_RSA_3DES_EDE_CBC_SHA
- *RSA_3DES_EDE_CBC_SHA
- *ECDHE_ECDSA_RC4_128_SHA
- *ECDHE_RSA_RC4_128_SHA
- *RSA_RC4_128_SHA
- *RSA_RC4_128_MD5
- *RSA_DES_CBC_SHA
- *RSA_EXPORT_RC4_40_MD5
- *RSA_EXPORT_RC2_CBC_40_MD5
- *ECDHE_ECDSA_NULL_SHA
- *ECDHE_RSA_NULL_SHA
- *RSA_NULL_SHA256
- *RSA_NULL_SHA

- *RSA_NULL_MD5

サポートされている暗号化仕様リストは、システムよってサポートされる SSL プロトコル、およびシステム値 QSSLCSL になされた変更によって影響を受けます。QSSLCSL の値を表示して、お使いのシステムの暗号化仕様リストを確認することができます。

出荷時の SSL デフォルト暗号化仕様リスト

以下にはデフォルト暗号化仕様リストの出荷時の順序が表示されています。

- *ECDHE_ECDSA_AES_128_CBC_SHA256
- *ECDHE_ECDSA_AES_256_CBC_SHA384
- *ECDHE_ECDSA_AES_128_GCM_SHA256
- *ECDHE_ECDSA_AES_256_GCM_SHA384
- *RSA_AES_128_CBC_SHA256
- *RSA_AES_128_CBC_SHA
- *RSA_AES_256_CBC_SHA256
- *RSA_AES_256_CBC_SHA
- *RSA_AES_128_GCM_SHA256
- *RSA_AES_256_GCM_SHA384
- *ECDHE_RSA_AES_128_CBC_SHA256
- *ECDHE_RSA_AES_256_CBC_SHA384
- *ECDHE_RSA_AES_128_GCM_SHA256
- *ECDHE_RSA_AES_256_GCM_SHA384
- *ECDHE_ECDSA_3DES_EDE_CBC_SHA
- *ECDHE_RSA_3DES_EDE_CBC_SHA
- *RSA_3DES_EDE_CBC_SHA
- *ECDHE_ECDSA_RC4_128_SHA
- *ECDHE_RSA_RC4_128_SHA
- *RSA_RC4_128_SHA

出荷時のデフォルト暗号化仕様リストは、QSSLCSL システム値を変更することによって削減および再配列することができます。

以下の表は、各プロトコル・バージョンでサポートされている暗号仕様を示します。各プロトコルでサポートされる暗号仕様は、該当する列にある「X」で示しています。

表 1. TLS および SSL プロトコルでサポートされる暗号仕様

QSSLCSL システム値表記	TLSv1.2	TLSv1.1	TLSv1.0	SSLv3	SSLv2
*RSA_AES_128_GCM_SHA256	X				
*RSA_AES_256_GCM_SHA384	X				
*ECDHE_ECDSA_NULL_SHA	X				
*ECDHE_ECDSA_RC4_128_SHA	X				
*ECDHE_ECDSA_3DES_EDE_CBC_SHA	X				
*ECDHE_RSA_NULL_SHA	X				
*ECDHE_RSA_RC4_128_SHA	X				

表 1. TLS および SSL プロトコルでサポートされる暗号仕様 (続き)

QSSLCSL システム値表記	TLSv1.2	TLSv1.1	TLSv1.0	SSLv3	SSLv2
*ECDHE_RSA_3DES_EDE_CBC_SHA	X				
*ECDHE_ECDSA_AES_128_CBC_SHA256	X				
*ECDHE_ECDSA_AES_256_CBC_SHA384	X				
*ECDHE_RSA_AES_128_CBC_SHA256	X				
*ECDHE_RSA_AES_256_CBC_SHA384	X				
*ECDHE_ECDSA_AES_128_GCM_SHA256	X				
*ECDHE_ECDSA_AES_256_GCM_SHA384	X				
*ECDHE_RSA_AES_128_GCM_SHA256	X				
*ECDHE_RSA_AES_256_GCM_SHA384	X				
*RSA_AES_256_CBC_SHA256	X				
*RSA_AES_128_CBC_SHA256	X				
*RSA_AES_256_CBC_SHA	X	X	X		
*RSA_AES_128_CBC_SHA	X	X	X		
*RSA_3DES_EDE_CBC_SHA	X	X	X	X	
*RSA_RC4_128_SHA	X	X	X	X	
*RSA_RC4_128_MD5	X	X	X	X	X
*RSA_DES_CBC_SHA		X	X	X	
*RSA_EXPORT_RC4_40_MD5			X	X	X
*RSA_EXPORT_RC2_CBC_40_MD5			X	X	X
*RSA_NULL_SHA256	X				
*RSA_NULL_SHA	X	X	X	X	
*RSA_NULL_MD5	X	X	X	X	
*RSA_RC2_CBC_128_MD5					X
*RSA_3DES_EDE_CBC_MD5					X
*RSA_DES_CBC_MD5					X

関連情報:

SSL システム値: QSSLCSLCTL

SSL システム値: QSSLCSL

署名アルゴリズム:

TLSv1.2 プロトコルでは、デジタル署名に使用される署名アルゴリズムおよびハッシュ・アルゴリズムを 1 つの独立した属性にしました。以前は、ネゴシエーション対象の暗号スイートがこれらの アルゴリズムを決定していました。システム SSL には、複数署名アルゴリズムをサポートするインフラストラクチャーがあります。

許可された署名アルゴリズムとハッシュ・アルゴリズムのペアの番号付きリストは、TLSv1.2 において以下の 2 つの目的のために役立ちますが、以前のプロトコルでは意味がありません。

証明書の選択

署名アルゴリズムの番号付きリストは、ハンドシェーク時にシステム SSL が証明書を要求する際にピアに送信されます。ピアは受け取ったリストを使用して、証明書選択プロセスを進めます。ピ

アはリストに準拠する証明書を選択する必要がありますが、そのことがすべてのインプリメンテーションおよび構成に当てはまるわけではありません。システム SSL は、オプションのクライアント認証が構成されている場合を除き、推奨されない署名アルゴリズムを使用する証明書を受け取ると、それをセッション・エラーとして処理します。

システム SSL は、認証要求を受け取ったときに、準拠する証明書を選択できない場合、使用可能な非準拠 RSA 証明書を送信します。ピアはこの証明書がセッション・エラーになるかどうかを判別します。システム SSL 証明書の選択ロジックについて詳しくは、22 ページの『複数の証明書の選択』を参照してください。

メッセージ・シグニチャー

アルゴリズム・ペアのリストは、ハンドシェーク・メッセージのディジタル署名に使用できる署名アルゴリズムおよびハッシュ・アルゴリズムを制限します。TLSv1.2 ハンドシェーク・メッセージ署名は、セッションに使用される証明書の署名とは異なる可能性があります。たとえばハンドシェーク・メッセージは、 MD5 証明書がセッションで選択されていても、SHA512 で保護できます。

システム SSL には、以下の署名アルゴリズムをサポートするインフラストラクチャーがあります。

- ECDSA_SHA512
- ECDSA_SHA384
- ECDSA_SHA256
- ECDSA_SHA224
- ECDSA_SHA1
- RSA_SHA512
- RSA_SHA384
- RSA_SHA256
- RSA_SHA224
- RSA_SHA1
- RSA_MD5

出荷時に SSL がサポートする署名アルゴリズム

システム SSL は、出荷時には以下のリストの署名アルゴリズムをサポートしています。

- ECDSA_SHA512
- ECDSA_SHA384
- ECDSA_SHA256
- ECDSA_SHA224
- ECDSA_SHA1
- RSA_SHA512
- RSA_SHA384
- RSA_SHA256
- RSA_SHA224
- RSA_SHA1
- RSA_MD5

出荷時の SSL デフォルト署名アルゴリズム

以下に、出荷時のデフォルト署名アルゴリズム・リストを順序どおりに示します。

- ECDSA_SHA512
- ECDSA_SHA384
- ECDSA_SHA256
- ECDSA_SHA224
- ECDSA_SHA1
- RSA_SHA512
- RSA_SHA384
- RSA_SHA256
- RSA_SHA224
- RSA_SHA1
- RSA_MD5

出荷時のデフォルト署名アルゴリズムのリストは、システム保守ツール (SST) 拡張分析コマンド SSLCONFIG を使用して変更できます。

関連概念:

24 ページの『SSLCONFIG マクロ』

SSLCONFIG マクロにより、システム全体のシステム SSL デフォルト・プロパティー を表示および変更することができます。

再ネゴシエーション:

既存のセキュア・セッション内部の新しいハンドシェーク・ネゴシエーションを開始することを、再ネゴシエーションといいます。システム SSL 再ネゴシエーションの特性を決定する 2 つのプロパティーがあります。

アプリケーションが再ネゴシエーションを使用する複数の理由があります。再ネゴシエーションはクライアントおよびサーバーのどちらからも開始できます。アプリケーション層は、セキュア・セッションがピアの要求によって再ネゴシエーションされることを認識していない場合もあります。gsk_secure_soc_misc() は、再ネゴシエーションを開始するために GSKit システム SSL アプリケーションが使用します。

ベース RFC によって定義された SSL および TLS プロトコル・アーキテクチャーには、再ネゴシエーションの不具合があります。プロトコルは、セッションの再ネゴシエーションが既存のセキュア・セッションにリンクされることを確認する暗号検査を実施できません。追加の RFC 5746 では、この問題を解決するために、基本プロトコルへのオプションの拡張を定義しています。

RFC 5746 は以前に定義されたプロトコルに最近追加されたものであるため、すべての SSL インプリメンテーションが現在これをサポートしているわけではありません。SSL インプリメンテーションの中には、RFC 5746 をサポートするように更新されていないものも、更新できないものもあります。様々な遷移の段階において事業継続性や相互運用性を可能にするために、 2 つの再ネゴシエーション・プロパティーがあります。

SSL 再ネゴシエーション・モード

システム SSL はデフォルトで、RFC 5746 セマンティクスをすべての再ネゴシエーション・ハンドシェークで使用することを要求します。デフォルト・モードは、システム保守ツール (SST) 拡張分析コマンド SSLCONFIG で変更できます。

このモードは、すべての非セキュアな再ネゴシエーションを許可するか、または短縮された非セキュアな再ネゴシエーションのみを許可するように設定することができます。これらのモードは、十分に検討してから使用してください。

ピアが開始したすべてのハンドシェーク再ネゴシエーションを使用不可にするモードがあります。このモードはセキュアなネゴシエーション (RFC 5746 セマンティクス) および非セキュアなネゴシエーションを妨げます。このモードは結果として、再ネゴシエーションの使用を必要とするアプリケーションの相互運用性の問題を引き起こす可能性があります。gsk_secure_soc_misc() のようなローカルに開始されたセキュアなネゴシエーションは、このモードで引き続き可能です。

SSL 拡張再ネゴシエーション・クリティカル・モード

拡張再ネゴシエーション・クリティカル・モードは、初期セッション・ネゴシエーション中に、システム SSL がどのような場合にすべてのピアに RFC 5746 再ネゴシエーション表示を要求するかを決定します。

セキュア・セッションの両サイドを、再ネゴシエーションの脆弱性から完全に保護するには、すべての初期ネゴシエーションが RFC 5746 のサポートを示す必要があります。この表示は、「renegotiation_info」TLS 拡張の形式または RFC 5746 で定義される Signaling Cipher Suite Value (SCSV) にすることができます。

クリティカル・モードはデフォルトでは無効で、 RFC 5746 をインプリメントしない SSL インプリメンテーションとの相互運用性を維持します。クリティカル・モードが有効にされると、システム SSL は RFC 5746 をインプリメントしたシステムとのネゴシエーションに制限されます。どちらのサイドでも再ネゴシエーションをサポートせず、使用しない場合でも、この制限が課されます。すべてのシステム SSL ピアが RFC 5746 をサポートすると判断された場合、このモードは有効に変更できます。

デフォルトの拡張再ネゴシエーション・クリティカル・モードは、システム保守ツール (SST) 拡張分析コマンド SSLCONFIG で変更できます。クライアント・アプリケーションのプロパティーとサーバー・アプリケーションの別個のプロパティーがあります。

システム SSL は常に、ClientHello で「renegotiation_info」TLS 拡張 または SCSV を送信します。SCSV は、 ClientHello の一部である拡張が他にない場合にのみ送信されます。

関連概念:

24 ページの『SSLCONFIG マクロ』
SSLCONFIG マクロにより、システム全体のシステム SSL デフォルト・プロパティー を表示および変更することができます。

関連情報:

RFC 5746: 「Transport Layer Security (TLS) Renegotiation Indication Extension」

DCM アプリケーション定義

ディジタル証明書マネージャー (DCM) は、アプリケーション定義を含むアプリケーション・データベースを管理します。各アプリケーション定義は、特定のアプリケーションに対する証明書処理情報をカプセル化

します。IBM i 7.1 のリリース以降は、アプリケーション定義はアプリケーションの システム SSL 属性の一部もカプセル化します。システム SSL のユーザーは、このアプリケーション定義を「アプリケーション ID」として認識します。

IBM i が提供する多くのアプリケーションは、アプリケーション定義を使用してそのアプリケーションの証明書情報を構成します。いずれのアプリケーション開発者も、アプリケーション定義を使用するようにアプリケーションを設計することができます。

DCM アプリケーション定義には、その識別に使用される 2 個のフィールドがあります。「**アプリケーション記述 (Application description)**」フィールドは、DCM のアプリケーション定義を検出して対話するために使用します。「**アプリケーション ID (Application ID)**」フィールドは、システム SSL が、構成情報を保持するアプリケーション定義を識別するために使用します。

以下の各 システム SSL プログラミング・インターフェースには、使用する「アプリケーション ID」を識別する方法があります。

- Global Security Kit (GSKit) API
  - gsk_attribute_set_buffer (属性 GSK_IBMI_APPLICATION_ID を使用)
- 統合 IBM i SSL_ API
  - SSL_Init_Application (構造体 SSLInitAppStr の設定値)
- 統合 IBM i JSSE インプリメンテーション
  - Java™ システム・プロパティー os400.secureApplication を設定します。

以下の DCM アプリケーション定義フィールドを使用して、アプリケーションの対応するシステム SSL 属性を制御することができます。

SSL プロトコル:

「**SSL プロトコル (SSL protocols)**」アプリケーション定義フィールドは、アプリケーションがサポートする SSL プロトコルのバージョンを決定します。

デフォルト値は *PGM です。この値は、この「アプリケーション ID」を使用するプログラムが、 SSL プロトコル属性を適切な値に設定することを意味します。すべてのシステム SSL プログラムには、 API 呼び出しを介して明示的に、またはシステム・デフォルトの使用を許可することにより暗黙的に設定されたプロトコル属性値があります。必須属性値がプログラムで設定されていないことが分かっている場合を除いて、*PGM を使用します。

*PGM によって適切なプロトコルが設定されない場合、このアプリケーション定義フィールドはこのアプリケーションでサポートされているプロトコルをオーバーライドできます。ここで識別されるプロトコルのうち少なくとも 1 つが QSSLPCL システム値によってシステムで有効にされている場合、システムで有効にされていないプロトコルはサイレントで無視されます。可能な場合は、アプリケーション定義フィールドを使用せず、アプリケーション資料に記載されている構成手順に従ってプロトコルを設定してください。管理者は、このフィールドを使用して、以前に可能であったよりも弱い IBM アプリケーションのセキュリティー・プロパティーを構成できます。

関連情報:

SSL システム値: QSSLPCL

SSL 暗号仕様オプション:

「SSL 暗号仕様オプション (SSL cipher specification options)」アプリケーション定義フィールドは、アプリケーションがサポートする SSL 暗号スイートを決定します。

デフォルト値は *PGM です。この値は、この「アプリケーション ID」を使用するプログラムが、サポートされる暗号スイート属性を適切な値に設定することを意味します。プログラムでは、API 呼び出しを介して明示的に、またはシステム・デフォルトの使用を許可することにより暗黙的に値を設定できます。

*PGM の値が正しくない場合、このフィールドでは、このアプリケーションがサポートする SSL 暗号スイートを定義できます。QSSLCSL システム値によって使用不可にされている暗号スイートは、少なくとも 1 個の暗号スイートが使用可能な場合は無視されます。

サーバー・アプリケーションは、優先順位付けリストによってサポートされる暗号スイートを制御します。管理者は、セキュリティー・ポリシー、パフォーマンス、および相互運用性の考慮事項を組み合わせて使用することにより、適切な構成を決定します。リストに変更を加える際は注意してください。ユーザー定義リストの柔軟性のために、*PGM を使用した場合よりも弱いセキュリティー構成が可能になります。セキュリティーは以下のようないくつかの方法で弱めることができます。

- 相対的に弱い暗号化アルゴリズムに高優先順位を選択する
- 相対的に強い暗号化アルゴリズムを使用不可にする
- 相対的に弱い暗号化アルゴリズムを使用可能にする

番号付きリストでの暗号スイートの優先度にかかわらず、サーバーは、サーバーが許可する最も弱い暗号スイートと同程度にしかセキュアになりません。

関連情報:

SSL システム値: QSSLCSL

拡張再ネゴシエーション・クリティカル・モード (Extended Renegotiation Critical Mode):

このアプリケーション定義フィールドは、初期ハンドシェーク時にアプリケーションがピアに RFC 5746 再ネゴシエーションを表示するように要求するかどうかを決定します。

この概念の詳細については、『システム SSL プロパティー』にある 12 ページの『再ネゴシエーション』トピックで 『SSL 拡張再ネゴシエーション・モード』を参照してください。

デフォルト値は *PGM です。この値は、この「アプリケーション ID」を使用するプログラムが、このモードをすでに適切な値に設定していることを意味します。プログラムは、この属性にシステム SSL のデフォルト値を使用しているか、 gsk_attribute_set_enum() API 呼び出しによって明示的に設定される値を使用しています。

初期ハンドシェークが正常に行われるために、RFC 5746 再ネゴシエーション表示がそのハンドシェークに含まれていなければならない場合は、「使用可能」に設定します。設計上、このアプリケーションは、RFC 5746 をサポートするように更新されていないピア、または RFC 5746 をサポートするように更新できないピアとは、ハンドシェークができなくなります。

初期ハンドシェークでアプリケーションがピアからの RFC 5746 再ネゴシエーション表示を必要としない場合は、このアプリケーション定義フィールドを「使用不可」に設定します。 RFC 5746 再ネゴシエーション表示は、すべての再ネゴシエーションされたハンドシェークで依然として必要とされます。

注: このアプリケーションは、この設定値にかかわらず、RFC 5746 再ネゴシエーション表示情報を常にピアに提供します。

関連情報:

RFC 5746: 「Transport Layer Security (TLS) Renegotiation Indication Extension」

Server Name Indication:

「**Server Name Indication (SNI)**」アプリケーション定義フィールドを使用して、このアプリケーションの限定的な SNI サポートを提供するようにシステム SSL に指示します。

RFC 6066 によって定義される SNI により、TLS クライアントは接続するサーバー名を TLS サーバーに提供できます。この機能を使用すると、単一の基本ネットワーク・アドレスで複数の「仮想」サーバーをホストするサーバーに対するセキュアな接続が容易になります。このような方法で、クライアントまたはサーバーとして SNI を使用するには、 gsk_attribute_set_buffer() を使用してアプリケーションの SNI を構成する必要があります。

限定的な SNI サポートが必要な場合は、サーバー・アプリケーションの完全修飾ドメイン・ネーム (FQDN) を入力します。不要な場合は、既存の SNI アプリケーション構成をオーバーライドしない限定的な SNI サポートであるデフォルト値 *NONE を受け入れます。

アプリケーションが gsk_attribute_set_buffer() を使用して SNI 情報を構成する場合、このアプリケーション定義フィールドに設定される値は、既存の情報の最後に追加されます。既存の情報がクリティカルとなるように構成されている場合、この値もまたクリティカルとなります。クリティカルとは、クライアント FQDN がサーバー・リスト内の名前と一致しない場合、サーバーがセッション・ネゴシエーションのエラーを生成することを意味します。既存の情報がない場合は、限定的な SNI サポートはクリティカルではありません。

限定的なサポート SNI フィールドを設定するユースケース: 企業のサービスを使用するユーザーが、その企業に問い合わせをしています。ユーザーには、通信しているすべての TLS サーバーがサーバー SNI 肯定応答を提供する、という新しいセキュリティー要件 があります。企業のサーバーは、仮想サーバー構成の必要がない、1 つのサービスのみで使用される単純なサーバーです。

サーバーの FQDN を設定することで、要求があればシステム SSL は SNI サーバー肯定応答を送信できます。送信されるサーバーの証明書は、アプリケーション定義に割り当てられたものです。サーバーのパースペクティブからは何も変更がありませんが、ピア・クライアントはその要件を満たしていました。

このフィールドが未設定であるか、またはフィールド値が異なるために、クライアントが要求した FQDN が一致しない場合、サーバーの SNI 肯定応答は送信されません。サーバーは、SNI が要求されていないかのように、ハンドシェーク・ネゴシエーションを継続します。クライアントは、この条件がセッション・ネゴシエーションの致命的エラーであるかどうかを判別します。

関連情報:

RFC 6066: 「Transport Layer Security (TLS) Extensions: Extension Definitions」

Online Certificate Status Protocol 属性:

アプリケーション定義の Online Certificate Status Protocol (OSCP) 属性のフィールドは、OCSP が使用可能かどうかを制御するために使用されます。

OSCP は、証明書の失効状況を決定するメカニズムです。この処理の動作を理解するには、詳細な OCSP の記述を参照してください。 GSKit API により、OCSP の処理を決定する多数の属性を構成できます。

このアプリケーション定義には、OCSP が使用可能かどうかを制御するために使用する 2 個の OCSP 属性フィールドがあります。その他の OCSP 属性はアプリケーション定義によって変更できません。これらのその他の属性値は、 GSKit API 設定または内部デフォルト値によって決定されます。

関連概念:

18 ページの『Online Certificate Status Protocol』
Online Certificate Status Protocol (OCSP) により、アプリケーションにディジタル証明書の失効状況を判別する方法が提供されます。 OCSP によってチェックされた証明書の失効状況は、CRL を通じて提供される状況情報よりも新しい情報です。

関連情報:

RFC 2560: X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP

OCSP URL:

「**Online Certificate Status Protocol (OCSP) URL**」アプリケーション定義フィールドは、このアプリケーションが一般的な OCSP レスポンダーを使用して、エンド・エンティティー証明書の検証時に要求を送信するかどうかを決定します。

URL が存在する場合、すべてのエンド・エンティティー証明書について、指定された OCSP レスポンダーにコンタクトをとり、失効状況を決定します。

このフィールドのデフォルト値は *PGM です。この値は、この「アプリケーション ID」を使用するプログラムが、属性を適切な値に設定することを意味します。すべてのシステム SSL 属性に、初期のデフォルト値があります。この属性のデフォルト値は、「URL 値なし (no URL value)」です。プログラムは gsk_attribute_set_buffer() を呼び出して明示的に URL 値を設定することもできます。

*PGM が必要な OCSP レスポンダーを指定しない場合、このフィールドに適切な OCSP レスポンダー URL を入力します。サポートされる URL プロトコルは、HTTP のみです。したがって、この値は、「http://」で始める必要があります。この値を設定すると、プログラムが内部的に URL 宛先に設定した構成がオーバーライドされます。ただし、これ以外の構成済み OCSP 属性は引き続き必要に応じて使用されます。

*PGM が OCSP レスポンダーを使用するアプリケーションを指定するが、一般的な OCSP レスポンダー処理が不要な場合は、このフィールドを「使用不可」に設定します。この設定は、gsk_attribute_set_buffer() を使用して内部的に構成された URL をオーバーライドします。OCSP を無効にすると、アプリケーションのセキュリティー・モデルが脆弱になるので、この選択をする際は十分配慮してください。

関連概念:

21 ページの『証明書の失効』
証明書の失効チェックは、セッション・ネゴシエーションの一部として実行される証明書の検証の 1 フェーズです。証明書チェーンは、証明書が失効していないことを確認するために検証されます。

OCSP AIA 処理:

「**Online Certificate Status Protocol (OCSP) 機関情報アクセス (AIA) 処理 (Online Certificate Status Protocol (OCSP) Authority Information Access (AIA) processing)**」 アプリケーション定義フィールドは、OCSP 証明書の失効チェックを AIA 証明書拡張情報を使用して実行するかどうかを決定します。

失効チェックは、OCSP 失効状況が不確定で、かつ以下の条件の両方が満たされている場合に、AIA 証明書拡張情報を使用して実行されます。

- OCSP AIA チェックが有効になっている。

- 検証される証明書に、OCSP レスポンダーの HTTP 位置の URI を含む、PKIK_AD_OCSP アクセス方式を使用する AIA 拡張がある。

  注: AIA 拡張で識別される最初の OCSP レスポンダーは、失効状況の照会が実行されます。

このフィールドのデフォルト値は *PGM です。この値は、この「アプリケーション ID」を使用するプログラムが、属性を適切な値に設定することを意味します。すべてのシステム SSL 属性に、初期のデフォルト値があります。この属性のデフォルト値は「無効 (disabled)」です。プログラムは、gsk_attribute_set_enum() によって OCSP AIA を明示的に使用可能または使用不可にすることができます。

*PGM が OCSP AIA の検証を実行せず、失効チェックが必要な場合は、このフィールドを「使用可能」に設定します。内部設定はオーバーライドされ、AIA 情報が使用可能な場合は OCSP チェックが実行されます。

*PGM が OCSP AIA 検証を実行するが、失効チェックが不要の場合は、このフィールドを「使用不可」に設定します。OCSP を無効にすると、アプリケーションのセキュリティー・モデルが脆弱になるので、この選択をする際は十分配慮してください。

関連概念:

21 ページの『証明書の失効』

証明書の失効チェックは、セッション・ネゴシエーションの一部として実行される証明書の検証の 1 フェーズです。証明書チェーンは、証明書が失効していないことを確認するために検証されます。

SSL 署名アルゴリズム (SSL Signature Algorithms):

「**SSL 署名アルゴリズム (SSL signature algorithms)**」アプリケーション定義フィールドは、アプリケーションがサポートするアルゴリズムを決定します。

署名アルゴリズムの内容および使用方法の概要については、『システム SSL プロパティー』にあるトピック 10 ページの『署名アルゴリズム』を参照してください。

デフォルト値は *PGM です。この値は、この「アプリケーション ID」を使用するプログラムが、 SSL 署名アルゴリズムを適切な値に設定することを意味します。すべてのシステム SSL 属性に、初期のデフォルト値があります。プログラムは gsk_attribute_set_buffer() を使用してリストを明示的に定義できます。

*PGM により不適切な構成となった場合、サポートされる署名アルゴリズムの必要な番号付きリストを含むようにこのフィールドを設定します。

Online Certificate Status Protocol

Online Certificate Status Protocol (OCSP) により、アプリケーションにディジタル証明書の失効状況を判別する方法が提供されます。 OCSP によってチェックされた証明書の失効状況は、CRL を通じて提供される状況情報よりも新しい情報です。

OCSP 失効状況チェックのインプリメンテーションは RFC 2560 に従って実行されます。OCSP 証明書の失効状況チェックはエンド・エンティティー証明書に対して使用可能です。HTTP を経由したプロトコル・バージョン 1 および基本応答タイプがサポートされます。

以下の条件のうち少なくとも 1 つが満たされている場合に、証明書の失効状況が OCSP を介してアプリケーションによりチェックされます。

- OCSP レスポンダーの URL アドレスが構成されている。

- 機関情報アクセス (AIA) チェックが有効で、検証される証明書に AIA 拡張がある。AIA 拡張には、OCSP レスポンダーの HTTP 位置を示す URI を使用する PKIK_AD_OCSP アクセス方式が含まれる必要があります。

  注: AIA 拡張で識別される最初の OCSP レスポンダーのみに、失効状況の照会が実行されます。

URL および AIA チェックが有効である場合、URL レスポンダーに送信された照会によって失効状況が不確定であることが分かった場合のみ AIA チェックが実行されます。

関連概念:

21 ページの『証明書の失効』
証明書の失効チェックは、セッション・ネゴシエーションの一部として実行される証明書の検証の 1 フェーズです。証明書チェーンは、証明書が失効していないことを確認するために検証されます。

関連情報:

RFC 2560: X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP

OCSP の構成:

Online Certificate Status Protocol (OCSP) の使用可能化に加え、アプリケーションがOCSP クライアントの動作をカスタマイズするために構成できるプロパティーが複数あります。

OCSP 失効チェックが有効な場合、HTTP 要求は OCSP レスポンダーに送信されます。要求には、失効状況が照会される証明書を識別する情報とオプションの署名が含まれます。要求のオプション署名は、レスポンダーがクライアントから受信した有効な要求を確認できるようにするために使用されます。要求の署名はデフォルトでは無効にされています。要求は、 HTTP を経由して GET または POST メソッドでレスポンダーに送信されます。 GET メソッドで送信された要求は HTTP キャッシュを有効にします。構成で GET メソッドが優先されていることが示され、かつ要求が 255 バイトより小さい場合、要求は GET メソッドで送信されます。それ以外の場合は、要求は POST メソッドで送信されます。GET メソッドはデフォルトで優先されます。

要求が送信されると、応答がレスポンダーから受信されるまで、またはタイムアウトになるまで、OCSP 失効チェックはブロックされます。失効チェックはセッション・ネゴシエーションの一部として実行されます。そのため、失効チェックが実行される間セッション・ネゴシエーションがブロックされます。セッション・ネゴシエーションで設定されたタイムアウトが、構成された OCSP タイムアウトよりも小さい場合、小さい方の値が OCSP タイムアウトに使用されます。OCSP タイムアウト値のデフォルトは 10 秒ですが、アプリケーションによって構成することができます。

有効な応答は、署名されており、照会された証明書の失効状況を示すデータを含んでいます。証明書の失効状況は、有効、不明、または失効です。応答は、以下の少なくとも 1 つの要件を満たす証明書によって署名されます。

- 署名証明書はローカルの証明書ストアで信頼されている。
- 署名証明書は検証する証明書を発行した認証局 (CA) のものである。
- 署名証明書は、ExtendedKeyUsage 拡張に id-ad-ocspSigning の値を含んでおり、対象の証明書を発行した CA によって発行されている。

応答のサイズはさまざまです。許可される最大の応答サイズの決定は、アプリケーションによります。デフォルトでは、許可される最大の応答サイズは 20480 バイトです。応答が、許可される最大サイズより大きい場合、応答は無視され、不明な証明書状況を示す応答と同じように処理されます。

暗号 nonce (ランダム・ストリング) 値は、受信した応答が特定の要求に対する応答であることを確認するために使用することができる、セキュリティー・メカニズムです。 nonce 値はランダムに生成されたビット・ストリングで、要求と応答の両方の一部として計算され、組み込まれます。nonce チェックが有効な場合、応答に含まれる nonce 値は、要求で送信された nonce 値と照らし合わせて検証されます。 nonce 値が一致しない場合は、応答は無視されます。 nonce チェックはデフォルトでは無効にされています。

失効チェックにより、セッション・ネゴシエーションが低速になることがあります。ただし、OCSP 応答のキャッシングにより、クライアントは同じ要求を再度送信することなく、以前の要求から失効状況を獲得することができます。 OCSP 応答キャッシュはデフォルトで有効ですが、アプリケーションで無効にすることができます。

プロキシー HTTP サーバーを中間サーバーとして使用して、キャッシュされた応答からの OCSP 要求を処理したり、構成されたレスポンダーに要求を転送したりすることができます。あるアプリケーションにプロキシー・サーバーが構成されている場合、そのアプリケーションのすべての OCSP 要求は、構成されたサーバーに送信されます。デフォルトのプロキシー・ポートは 80 です。プロキシー・サーバーはデフォルトでは構成されていません。

Global Security Kit (GSKit) API である、gsk_attribute_set_buffer()、 gsk_attribute_set_numeric_value()、および gsk_attribute_set_enum() は、以下の API 属性を使用して OCSP を構成するために使用することができます。

- GSK_OCSP_URL: OCSP 要求の送信先 OCSP レスポンダーの URL
- GSK_OCSP_ENABLE: AIA チェックを有効にする
- GSK_OCSP_REQUEST_SIGKEYLABEL: OCSP 要求に署名するために使用する証明書の証明書ラベル
- GSK_OCSP_REQUEST_SIGALG: OCSP 要求の署名を生成するために使用する署名アルゴリズム
- GSK_OCSP_RETRIEVE_VIA_GET: OCSP 要求の送信に使用されるメソッド
- GSK_OCSP_TIMEOUT: OCSP レスポンダーからの応答を待つ秒数
- GSK_OCSP_MAX_RESPONSE_SIZE: OCSP レスポンダーから受け入れる最大応答サイズ (バイト単位)
- GSK_OCSP_CLIENT_CACHE_SIZE: OCSP クライアントの応答キャッシュを有効または無効にする
- GSK_OCSP_NONCE_GENERATION_ENABLE: OCSP 要求の一部として nonce 拡張を送信する
- GSK_OCSP_NONCE_CHECK_ENABLE: OCSP 応答の nonce 拡張が OCSP 要求で送信されたものと一致するか検証する
- GSK_OCSP_NONCE_SIZE: nonce 値を生成するために使用するバイト数
- GSK_OCSP_PROXY_SERVER_NAME: OCSP 要求の送信先プロキシー・サーバーのサーバー名
- GSK_OCSP_PROXY_SERVER_PORT: OCSP 要求の送信先プロキシー・サーバーのポート番号

統合 IBM i SSL_ API または IBM i JSSE を使用するアプリケーションには、OCSP を構成するインターフェースがありません。ただし「アプリケーション ID」を使用するプログラムは、 DCM による OCSP 失効チェックを有効または無効にすることができます。これ以外のすべての OCSP 構成オプションにデフォルト値が使用されます。

関連概念:

16 ページの『Online Certificate Status Protocol 属性』
アプリケーション定義の Online Certificate Status Protocol (OSCP) 属性のフィールドは、OCSP が使用可能かどうかを制御するために使用されます。

関連情報:

Global Security Kit (GSKit) API

OCSP 失効状況:

OCSP 失効状況は、OCSP 要求への返答として送られた OCSP 応答によって決まります。 2 つのタイプの応答が受信されます。一方は OCSP レスポンダーが有効な応答を送信したことを示します。もう一方は、レスポンダーが前の要求を処理する際に問題が発生したことを通知します。

レスポンダーで要求の処理中に発生する可能性のある問題には、以下があります。

- malformedRequest - 要求の形式が誤っている。
- internalError - OCSP レスポンダーで内部エラーが発生した。
- tryLater - OCSP レスポンダーは一時的に応答ができなくなったため、後で要求を再試行する。
- sigRequired - OCSP レスポンダーで、要求が署名されることが必須となっている。
- unauthorized - OCSP クライアントに OCSP レスポンダーを照会する権限がない。

有効な応答には照会された証明書の失効状況が含まれます。証明書状況の値には、有効、失効、不明があります。 OCSP 失効状況チェックは、証明書の有効または失効という失効状況を受け取った場合に完了します。状況が有効の場合はハンドシェークが続行でき、失効の場合はハンドシェークが失敗します。

URL および AIA チェックの両方の失効状況が不確定の場合は、状況が失効ではないかのように検証が続行されます。失効状況が不確定である証明書に関する情報は、 gsk_attribute_get_buffer() および属性 GSK_UNKNOWNREVOCATIONSTATUS_SUBJECT を使用して検索できます。

不確定な失効状況

以下の応答があると、失効状況は不確定になります。

- 指定されたタイムアウト期間内に受け取った応答がない
- レスポンダーに問題が発生したことを示す OCSP 応答状況
- 以下の条件のいずれかを満たす有効な応答
  - 不明の応答タイプ (PKIX_AD_OCSP_basic 応答タイプのみサポート)
  - 不明な応答バージョン (バージョン 1 のみサポート)
  - 無効な署名または無効な署名証明書
    - 署名証明書は以下の基準のいずれかを満たす必要があります。
      - ローカル鍵ストアに信頼されている
      - 対象の証明書を発行した認証局 (CA) の証明書である
      - ExtendedKeyUsage 拡張に id-ad-ocspSigning の値を含み、対象の証明書を発行した CA によって発行されている
  - nonce チェックが必要な場合で、応答に nonce が含まれない
  - nonce チェックが必要な場合で、応答に無効な nonce 値がある
  - 無効であるかまたは有効期限が切れた nextUpdate 値が応答で指定されている
  - レスポンダーが証明書の状況を認識していないことを示す、不明という証明書状況値

証明書の失効

証明書の失効チェックは、セッション・ネゴシエーションの一部として実行される証明書の検証の 1 フェーズです。証明書チェーンは、証明書が失効していないことを確認するために検証されます。

以下の手順を実行して証明書の失効チェックを行います。

1. 証明書失効リスト (CRL) の位置を使用して失効状況をチェックします。

a. CRL 位置がディジタル証明書マネージャー (DCM) を通じて構成されている場合、 CRL データベース (LDAP) サーバーに対して、証明書の失効状況を含む CRL についての照会が行われます。
   - 証明書が失効している場合、証明書検証の証明書の失効フェーズが完了し、セッション・ネゴシエーションが失敗します。
   - それ以外の場合、証明書の失効処理を継続します。

   注: CRL 位置は、ローカルの証明書ストアで各認証局 (CA) に対して個々に構成されます。

2. Online Certificate Status Protocol (OCSP) で失効状況を確認します。
   a. OCSP URL レスポンダー・アドレスが構成されている場合、レスポンダーに照会します。
      - 証明書が失効している場合、証明書検証の証明書の失効フェーズが完了し、セッション・ネゴシエーションが失敗します。
      - 証明書が有効である場合、証明書検証の証明書の失効フェーズが完了し、証明書検証が続行されます。
      - 証明書の失効状況が不確定の場合、証明書の失効処理を続行します。
   b. AIA チェックが有効で、証明書に HTTP 位置を示す URI を使用する PKIK_AD_OCSP アクセス方式がある場合、レスポンダーに照会します。
      - 証明書が失効している場合、証明書検証の証明書の失効フェーズが完了し、セッション・ネゴシエーションが失敗します。
      - 証明書が有効である場合、証明書検証の証明書の失効フェーズが完了し、証明書検証が続行されます。
      - 証明書の失効状況が不確定の場合、証明書検証の証明書の失効フェーズが完了し、証明書検証が続行されます。

   注: 失効状況が不確定の場合、GSKit は失効状況が不確定な証明書に関する情報を保管し、状況が失効ではないかのように処理を続行します。アプリケーションは、 gsk_attribute_get_buffer() と属性 GSK_UNKNOWNREVOCATIONSTATUS_SUBJECT を使用して不確定な証明書状況情報を検索し、接続を継続するか終了するかに関するポリシーを決定します。

注: DCM で構成されるアプリケーション定義は、 アプリケーション定義を使用するアプリケーションによって構成される CRL および OCSP 失効チェックをオーバーライドできます。

関連概念:

18 ページの『Online Certificate Status Protocol』
Online Certificate Status Protocol (OCSP) により、アプリケーションにディジタル証明書の失効状況を判別する方法が提供されます。 OCSP によってチェックされた証明書の失効状況は、CRL を通じて提供される状況情報よりも新しい情報です。

関連情報:

CRL 位置の管理

複数の証明書の選択

システム SSL は、最大で 4 つの証明書をセキュアな環境に割り当てることができます。複数証明書の目的は、楕円曲線ディジタル署名アルゴリズム (ECDSA) 証明書を使用可能にすると同時に、 RSA を必要とするクライアントで RSA 証明書を引き続き使用できるようにすることにあります。

相互運用性を最大にするために、サーバーは、様々な SSL 機能を持つ幅広いクライアントとのネゴシエーションが可能である必要があります。 あるクライアントが TLSv1.2 または ECDSA 証明書をサポートしていない場合、サーバーは RSA 証明書とのネゴシエーションが可能な状態を保持する必要があります。

1 つの環境に対して複数の証明書を構成するには 2 つの方法があります。

- ディジタル証明書マネージャー (DCM) を使用して、セキュアな環境に対して構成されるアプリケーション定義に複数の証明書を割り当てます。
- GSKit API gsk_attribute_set_buffer (GSK_KEYRING_LABEL_EX) を使用して、使用する証明書ラベルのリストを構成します。

各 SSL ハンドシェーク時に、セッションの属性に基づき、セッションが使用する適切な証明書が選択されます。選択プロセスは、クライアントとサーバーの両方の SSL 属性を使用して証明書を決定します。属性の組み合わせによっては、使用可能な証明書がないこともあります。

選択時の考慮事項

ネゴシエーション対象のプロトコルが TLSv1.1、TLSv1.0、SSLv3、または SSLv2 の場合、RSA 署名も持つ最初に検出された RSA 証明書が使用されます。

ネゴシエーション対象のプロトコルが TLSv1.2 の場合、選択プロセスでは複数の手順が実行されます。

1. クライアントでサポートされている、サーバーの最も推奨する暗号スイートのキー・タイプが選択プロセスを開始します。証明書には ECDSA または RSA キーのいずれかがあります。暗号スイートの名前に「ECDSA」が含まれる場合は、ECDSA キー/証明書が使用される必要があります。暗号スイートの名前に「RSA」が含まれる場合は、RSA キー/証明書が使用される必要があります。適合する証明書がない場合、証明書のいずれかで機能する暗号スイートが見つかるまで、選択はリスト内の次の暗号スイートに移動していきます。

   1 つ以上の適合する証明書キー・タイプが検出された場合は、追加の基準がチェックされ、残りのハンドシェーク属性に基づいて使用できるキー・タイプが判別されます (そのようなものがある場合)。
2. 検討しているキー・タイプが ECDSA である場合、指定された曲線 (キー・サイズと等価である) がピアによってサポートされる必要があります。検討対象の ECDSA 証明書に別の指定曲線がある場合、ピアの推奨する指定曲線の順序を使用して推奨される証明書を決定します。この手順の後、1 つ以上の証明書を選択対象としてまとめることができます。この手順の後まで適格な状態を維持する証明書がない場合は、前の手順に戻ります。
3. RSA 証明書と ECDSA 証明書の両方に証明書の発行認証局からの署名があります。署名のキー・タイプは、サーバーの証明書キー・タイプとは異なることがあります。このフェーズで複数の証明書が選択項目として有効なままである場合、ピアが最も推奨する署名のあるものが選択されます。

   ECDSA 証明書には、ピアの順序付けされた署名アルゴリズム・リストで許可された署名アルゴリズムが必要です。この手順で、サポートされる署名アルゴリズムを持つ ECDSA 証明書がない場合は、前の手順に戻り、優先順位が低い指定曲線を追加の選択処理で検討することができます。

   この手順で、サポートされる署名アルゴリズムを持つ RSA 証明書がない場合は、最初の手順に戻り、異なるキー・タイプを追加の選択処理で検討することができます。

   この手順の後に、依然として複数の証明書が同等の場合、最初に処理されたものが選択されます。残りの同等な証明書は、最初の証明書と暗号的に同一となるため、選択されることはありません。構成から追加の証明書を削除して、余分な選択処理が行われないようにします。
4. 以前のリリースとの互換性のため、RSA 証明書について最終的に考慮すべき事項が 1 つあります。前述の選択基準に一致する証明書がない状態で処理が完了した場合、RSA 暗号スイートがリストにあれば、処理された最初の RSA 証明書が選択されます。

SSLCONFIG マクロ

SSLCONFIG マクロにより、システム全体のシステム SSL デフォルト・プロパティー を表示および変更することができます。

SSL 構成の IBM 提供マクロ・サポートを使用するには、以下のステップに従います。

1. SST を使用して、システム保守ツールにアクセスします。
2. 「**保守ツールの開始**」を選択します。
3. 「**表示/変更/ダンプ**」を選択します。
4. 「**ストレージの表示/変更 (Display/Alter storage)**」を選択します。
5. 「**ライセンス内部コード (LIC) データ (Licensed Internal Code (LIC) data)**」を選択します。
6. 「**拡張分析 (Advanced analysis)**」を選択します。(このオプションを表示するには、ページ送りをする必要があります。)
7. 「SSLCONFIG」オプションが表示されるまで、ページを送ります。次に、このオプションの横に 1 を入力 (選択) し、Enter キーを押します。 「拡張分析オプションの指定 (Specify Advanced Analysis Options)」ウィンドウが表示されます。コマンドは SSLCONFIG と表示されます。
8. 括弧なしで「-h」と入力し、Enter キーを押して使用可能なオプションを表示します。

サーバー認証

サーバー認証の場合、クライアントは、サーバー証明書が有効であり、このクライアントが信頼する認証局 (CA) によってそれが署名されていることを確認します。

SSL は、非対称暗号およびハンドシェーク・プロトコル・フローを使用して、この固有な SSL セッションだけに使用する対称鍵を生成します。対称鍵は、SSL セッションを流れるデータの暗号化と暗号化解除に使用する鍵のセットを生成するために使用します。次に、SSL ハンドシェークが完了すると、通信リンクの一方のエンドポイントまたは両方のエンドポイントが認証されます。そして、データの暗号化と暗号化解除に使用する固有な鍵が生成されます。ハンドシェークが完了すると、暗号化されたアプリケーション層データがその SSL セッションを流れます。

クライアント認証

多くのアプリケーションは、クライアント認証を使用可能にするオプションを備えています。クライアント認証の場合、サーバーは、クライアント証明書が有効で、かつサーバーが信頼する認証局によって署名されていることを確認します。

以下の IBM i アプリケーションは、クライアント認証をサポートします。

- IBM HTTP Server for i
- FTP サーバー
- Telnet サーバー
- マネージメント・セントラル・エンドポイント・システム
- IBM Tivoli® Directory Server for IBM i

関連情報:

ディレクトリー・サーバーでの Secure Sockets Layer (SSL) および Transport Layer Security (TLS)

Transport Layer Security または Secure Sockets Layer を使用した FTP クライアントの保護

SSL を使用した Telnet の保護

HTTP Server の管理 (ADMIN) サーバー用の SSL のセットアップ

SSL の前提条件

このトピックでは、IBM i でのシステム SSL の前提条件、およびいくつかの役に立つヒントを示しています。

SSL を使用する前に、以下のオプションがインストールされていることを確認します。

- IBM ディジタル証明書マネージャー (DCM) (5770-SS1 オプション 34)

  注: IBM Java Secure Socket Extension (JSSE) および OpenSSL は DCM を必要としません。
- IBM TCP/IP Connectivity Utilities for i (5770-TC1)
- IBM HTTP Server for i (5770-DG1)
- HTTP サーバーを使用して DCM を使用する場合には、IBM Developer Kit for Java (5770-JV1) がインストール済みであることを確認してください。そうでない場合、HTTP 管理サーバーは開始しません。
- 暗号化ハードウェアをインストールし、SSL で使用するように構成して、SSL ハンドシェーク処理の速度を高めることもできます。暗号化ハードウェアをインストールする場合は、 IBM CCA Service Provider (5770-SS1 オプション 35) および IBM Cryptographic Device Manager (5733-CY3) もインストールする必要があります。

関連概念:

39 ページの『SSL のトラブルシューティング』
これはごく基本のトラブルシューティング情報であり、 IBM i プラットフォームが SSL を使用する際に発生する可能性のある問題のリストを削減することを目的としています。

関連情報:

暗号化ハードウェア

公開証明書と秘密証明書

DCM の構成

SSL によるアプリケーション・セキュリティー

IBM i の多数のアプリケーションを SSL で保護することができます。詳しくは、各アプリケーションの資料を参照してください。

関連概念:

30 ページの『シナリオ: SSL によるマネージメント・セントラル・サーバーへのすべての接続の保護』
このシナリオは、SSL を使用して IBM i とのすべての接続を保護する方法を説明しています。IBM i は System i® ナビゲーターのマネジメント・セントラル・システムを使用して、セントラル・システムとして機能しています。

関連情報:

エンタープライズ識別マッピング

SSL を使用した FTP サーバーの保護

HTTP サーバー

Secure Sockets Layer の管理 (iSeries Access for Windows のトピック)

Telnet シナリオ: SSL で保護された Telnet

Secure Sockets API

シナリオ: SSL

SSL シナリオは、IBM i で SSL を使用可能にすることによって得られるメリットを最大にすることを目的としています。

SSL 使用の実現可能な例を提供する SSL シナリオを読むことで、IBM i での SSL の実行に関する理解を深めることができます。

関連情報:

シナリオ: SSL を使用した Telnet の保護

シナリオ: 暗号化ハードウェアを使用した秘密鍵の保護

シナリオ: SSL によるマネージメント・セントラル・サーバーへのクライアント接続の保護

このシナリオは、System i ナビゲーター のマネージメント・セントラル・サーバーを使用することでセントラル・システムとして機能している IBM i と、リモート・クライアントとの間の接続を、SSL を使用して保護する方法を説明しています。

状況:

ある企業が、数台の IBM i システムを組み込んだローカル・エリア・ネットワーク (LAN) をオフィスに構築しています。この企業のシステム管理者であるボブは、この IBM i システムの 1 つを LAN のセントラル・システム (今後、システム A と呼びます) に指定しました。ボブは、システム A でマネージメント・セントラル・サーバーを使用して、LAN 上にある他のエンドポイントをすべてを管理しています。

ボブは、システム A のマネージメント・セントラル・サーバーに、社内 LAN の外部のネットワークから接続されることを心配しています。ボブは出張が多いため、外出している間、マネージメント・セントラル・サーバーへのセキュアな接続が必要です。彼はオフィスにいない場合、自分の PC とマネージメント・セントラル・サーバーの間の接続を確実にセキュアにしたいと思っています。ボブは、自分の PC とシステム A のマネージメント・セントラル・サーバーで、SSL を使用可能にすることを決めました。このように SSL を使用可能にすると、出張時にマネージメント・セントラル・サーバーへの接続を確実にセキュアにすることができます。

目的:

ボブは、自分の PC とマネージメント・セントラル・サーバーの間の接続を確実にセキュアにしたいと思っています。ボブは、システム A 上のマネージメント・セントラル・サーバーと LAN 上のエンドポイントの間の接続に、セキュリティーを追加する必要は感じていません。この企業のオフィスで働いている他の従業員たちも、マネージメント・セントラル・サーバーへの接続に関して、追加のセキュリティーを必要としていません。ボブの計画は、接続でサーバー認証を使用するように、自分の PC とシステム A のマネージメント・セントラル・サーバーを構成することです。他の PC または LAN 上の IBM i システムからマネージメント・セントラル・サーバーへの接続は、SSL により保護されていません。

詳細:

次の表は、PC クライアント上で SSL が使用可能であるか使用不可であるかに基づいて、使用される認証のタイプを説明したものです。

表 2. SSL によるクライアントとマネージメント・セントラル・サーバー間の接続の保護に必要な要素

ボブの PC での SSL の状況	システム A のマネージメント・セントラル・サーバーに指定された認証レベル	SSL 接続が使用可能か
SSL 設定はオフ	任意	いいえ
SSL 設定はオン	任意	はい (サーバー認証)

サーバー認証は、ボブの PC でマネージメント・セントラル・サーバーの証明書を認証することを意味します。マネージメント・セントラル・サーバーに接続する場合は、ボブの PC は SSL クライアントとして機能します。マネージメント・セントラル・サーバーは、SSL サーバーとして機能し、ID を証明しなければなりません。マネージメント・セントラル・サーバーは、ボブの PC が信頼する認証局 (CA) により発行された証明書を提供することによって、ID を証明します。

前提条件および前提事項

ボブは、自分の PC とシステム A のマネージメント・セントラル・サーバーの間の接続を保護するため、以下の管理タスクおよび構成タスクを行わなければなりません。

1. システム A を SSL の前提条件に合わせます。
2. システム A は i5/OS™ V5R3 以降で稼働します。
3. PC クライアントは IBM i Access for Windows V5R3 以降で稼動します。
4. IBM i システムの認証局 (CA) を取得します。
5. システム A 用に CA によって署名された証明書を作成します。
6. CA および証明書をシステム A に送信し、それらを鍵データベースにインポートします。
7. マネージメント・セントラル・サーバー ID およびすべての IBM i システムのアプリケーション ID を証明書に割り当てます。 TCP セントラル・サーバー、データベース・サーバー、データ待ち行列サーバー、ファイル・サーバー、ネットワーク・プリント・サーバー、リモート・コマンド・サーバーおよびサインオン・サーバーは、すべて IBM i システムです。
   a. システム A で、IBM ディジタル証明書マネージャーを始動します。ボブが証明書の取得または作成を行います。もしくは、ここで認証システムのセットアップまたは変更を行います。
   b. 「**証明書ストアの選択**」を選択します。
   c. 「***SYSTEM**」を選択し、「**続行**」をクリックします。
   d. *「証明書ストア・パスワード」*に *SYSTEM を入力し、「**続行**」をクリックします。メニューが再ロードされたら、「**アプリケーションの管理**」を展開します。
   e. 「**証明書割り当ての更新**」をクリックします。
   f. 「**サーバー**」を選択し、「**続行**」をクリックします。
   g. 「**マネージメント・セントラル・サーバー**」を選択し、「**証明書割り当ての更新**」をクリックします。これにより、証明書が使用するマネージメント・セントラル・サーバーに割り当てられます。
   h. 「**新規証明書の割り当て**」をクリックします。DCM は、「証明書割り当ての更新」ページを再ロードして、確認メッセージを表示します。
   i. 「**完了**」をクリックします。
   j. すべてのクライアント・アクセス・サーバーに証明書を割り当てます。
8. CA を PC のクライアントにダウンロードします。

ボブがマネージメント・セントラル・サーバーで SSL を使用可能にする前に、SSL 前提条件のプログラムをインストールし、システムにディジタル証明書をセットアップする必要があります。前提条件を満たした

ら、以下の手順を完了させて、マネージメント・セントラル・サーバーで SSL を使用可能にできます。

構成ステップ

ボブは、SSL によって、自分の PC からシステム A のマネージメント・セントラル・サーバーへの接続を保護するために、次のステップを完了する必要があります。

1. 『ステップ 1: System i ナビゲーター クライアントについて SSL を非アクティブにする』
2. 29 ページの『ステップ 2: マネージメント・セントラル・サーバーの認証レベルを設定する』
3. 29 ページの『ステップ 3: セントラル・システム上のマネージメント・セントラル・システムを再始動する』
4. 29 ページの『ステップ 4: System i ナビゲーター クライアントについて SSL をアクティブにする』
5. 29 ページの『オプショナル・ステップ: System i ナビゲーター クライアントについて SSL を非アクティブにする』

関連概念:

25 ページの『SSL の前提条件』
このトピックでは、IBM i でのシステム SSL の前提条件、およびいくつかの役に立つヒントを示しています。

関連情報:

DCM の構成

ディジタル証明書マネージャーの開始

構成の詳細: SSL によるマネージメント・セントラル・システムへのクライアント接続の保護

このトピックでは、SSL を使用してマネージメント・セントラル・サーバーへのクライアント接続を保護する拡張された構成ステップについて説明しています。

次の情報は、『シナリオ: SSL によるマネージメント・セントラル・サーバーへのクライアント接続の保護』に目を通していることを前提としています。

このシナリオでは、IBM i は、企業のローカル・エリア・ネットワーク (LAN) のセントラル・システムとして指定されています。ボブは、セントラル・システム (ここではシステム A と呼びます) 上のマネージメント・セントラル・サーバーを使用して、企業のネットワークのエンドポイントを管理しています。次の情報で、マネージメント・セントラル・サーバーに対する外部のクライアント接続を保護するために必要なステップを行う方法を説明します。ボブがシナリオの構成ステップを完了するのを追っていきます。

関連概念:

25 ページの『SSL の前提条件』
このトピックでは、IBM i でのシステム SSL の前提条件、およびいくつかの役に立つヒントを示しています。

30 ページの『シナリオ: SSL によるマネージメント・セントラル・サーバーへのすべての接続の保護』
このシナリオは、SSL を使用して IBM i とのすべての接続を保護する方法を説明しています。IBM i は System i ナビゲーターのマネジメント・セントラル・システムを使用して、セントラル・システムとして機能しています。

関連情報:

証明書のはじめてのセットアップ

ステップ 1: System i ナビゲーター クライアントについて SSL を非アクティブにする:

このステップは、System i ナビゲーター クライアントで SSL を使用可能にしてある場合のみ必要です。

1. System i ナビゲーターで、「**ユーザー接続**」を展開します。
2. システム A を右クリックし、「**プロパティー**」を選択します。
3. 「**セキュア・ソケット**」タブをクリックし、「**SSL (Secure Sockets Layer) を接続に使用**」を選択解除します。
4. System i ナビゲーターを終了し、再始動します。

パッドロックが、System i ナビゲーター のマネージメント・セントラル・コンテナーから見えなくなります。これは、接続が非セキュアであるということを示しています。このことは、ボブが、クライアントと企業のセントラル・システムの間に SSL で保護された接続を保持していないことを示しています。

ステップ 2: マネージメント・セントラル・サーバーの認証レベルを設定する:

1. System i ナビゲーター で、「**マネージメント・セントラル**」を右クリックし、「**プロパティー**」を選択します。
2. 「**セキュリティー**」タブをクリックし、「**Secure Sockets Layer (SSL) を使用**」を選択します。
3. 認証レベルで**いずれか**を選択します。(IBM i Access for Windows で使用可能です)
4. 「**OK**」をクリックして、この値をセントラル・システムに設定します。

ステップ 3: セントラル・システム上のマネージメント・セントラル・システムを再始動する:

1. System i ナビゲーターで、「**ユーザー接続**」を展開します。
2. システム A で、「**ネットワーク**」**-->**「**サーバー**」の順に展開し、「**TCP/IP**」を選択します。
3. 「**マネージメント・セントラル**」を右クリックし、「**停止**」を選択します。「セントラル・システム (central system)」ビューは縮小表示され、サーバーには接続されていないという内容のメッセージが表示されます。
4. マネージメント・セントラル・サーバーが停止したら、「**開始**」をクリックして、再始動します。

ステップ 4: System i ナビゲーター クライアントについて SSL をアクティブにする:

1. System i ナビゲーターで、「**ユーザー接続**」を展開します。
2. システム A を右クリックし、「**プロパティー**」を選択します。
3. 「**セキュア・ソケット**」タブをクリックし、「**SSL (Secure Sockets Layer) を接続に使用**」を選択します。
4. System i ナビゲーターを終了し、再始動します。

パッドロックは、System i ナビゲーター のマネージメント・セントラル・サーバーの横に表示されます。これは、SSL で接続がセキュアになっていることを示します。このことは、ボブが、彼のクライアントと彼の企業のセントラル・システムの間の SSL でのセキュアな接続をアクティブにするのに成功したということを示しています。

注: この手順は、1 つの PC とマネージメント・セントラル・システムの間の接続のみをセキュアにします。マネージメント・セントラル・サーバーへの他のクライアント接続や、エンドポイントからマネージメント・セントラル・サーバーへの接続は、セキュアになりません。他のクライアントをセキュアにするためには、前提条件を満たしていることを確認してから、『ステップ 4: System i ナビゲーター クライアントについて SSL をアクティブにする』を繰り返し行ってください。マネージメント・セントラル・サーバーとの他の接続を保護するには、『シナリオ: SSL によるマネージメント・セントラル・サーバーへのすべての接続の保護』を参照してください。

オプショナル・ステップ: System i ナビゲーター クライアントについて SSL を非アクティブにする:

ボブがオフィスで仕事をしていて、SSL 接続によって彼の PC のパフォーマンスに影響を与えたくない場合は、次のステップを実行することで簡単に SSL を非アクティブにすることができます。

1. System i ナビゲーターで、「**ユーザー接続**」を展開します。
2. システム A を右クリックし、「**プロパティー**」を選択します。
3. 「**セキュア・ソケット**」タブをクリックし、「**SSL (Secure Sockets Layer) を接続に使用**」を選択解除します。
4. System i ナビゲーターを終了し、再始動します。

シナリオ: SSL によるマネージメント・セントラル・サーバーへのすべての接続の保護

このシナリオは、SSL を使用して IBM i とのすべての接続を保護する方法を説明しています。IBM i は System i ナビゲーターのマネジメント・セントラル・システムを使用して、セントラル・システムとして機能しています。

状況:

ある企業が最近、複数の IBM i システムをリモート・ロケーション (エンドポイント) に置く広域ネットワーク (WAN) をセットアップしました。エンドポイントは、メイン・オフィスにある 1 台のシステム (セントラル・システム) によって中央管理されています。トムは、この企業のセキュリティー・スペシャリストです。トムは、企業のセントラル・システムのマネージメント・セントラル・サーバーと、すべての IBM i システムおよびクライアントとの間の接続を、すべてセキュアにするために、Secure Sockets Layer (SSL) を使用したいと思っています。

詳細:

トムは、SSL を使用することにより、マネージメント・セントラル・サーバーへのすべての接続を、**セキュアに**管理することができます。マネージメント・セントラル・サーバーで SSL を使用するには、トムは、セントラル・システムへのアクセスに使用する PC で、System i ナビゲーターを保護する必要があります。

トムは、マネージメント・セントラル・サーバー用に以下の 2 つの認証レベルを選択します。

サーバー認証
: サーバー証明書の認証を行います。クライアントは、クライアントが PC 上の System i ナビゲーター にあるか、セントラル・システム上のマネージメント・セントラル・サーバーにあるかを検証する必要があります。System i ナビゲーターがセントラル・システムに接続しているとき、PC は SSL クライアントであり、セントラル・システムで実行しているマネージメント・セントラル・サーバーは SSL サーバーです。エンドポイント・システムに接続する場合は、セントラル・システムは SSL クライアントとして機能します。エンドポイント・システムは SSL サーバーとして機能し、サーバーは、クライアントが信頼する認証局によって発行された証明書を提供することによって、クライアントに ID を証明しなければなりません。すべての SSL サーバーには、トラステッド CA から有効な証明書が発行される必要があります。

クライアントおよびサーバー認証
: セントラル・システム証明書とエンドポイント・システム証明書の両方の認証を行います。この認証は、サーバー認証レベルよりも高いセキュリティー・レベルです。他のアプリケーションでは、この認証はクライアント認証と呼ばれています。その場合、クライアントは有効な信頼できる証明書を提供する必要があります。セントラル・システム (SSL クライアント) がエンドポイント・シ

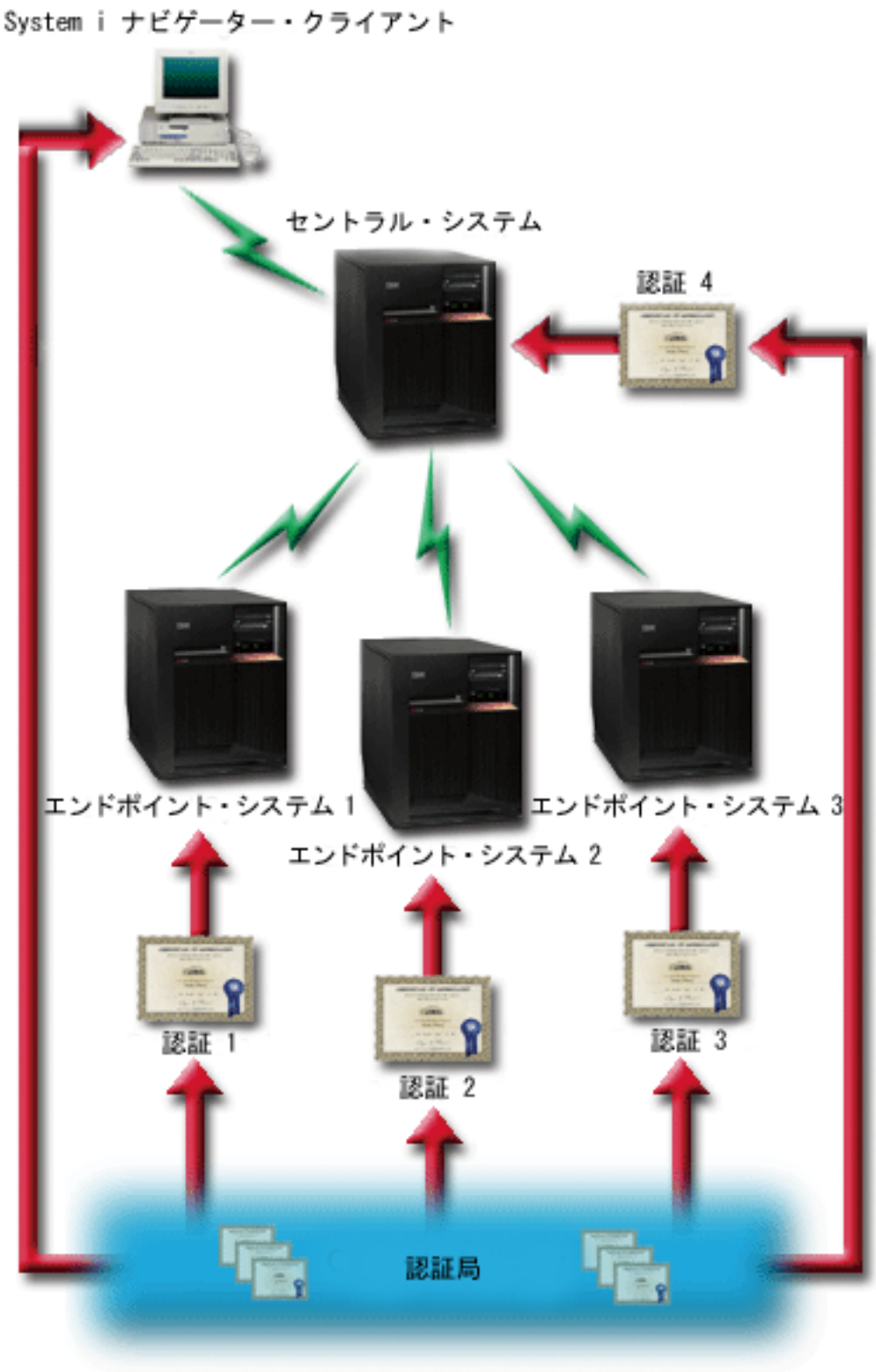
ステム (SSL サーバー) との接続を確立しようとすると、セントラル・システムとエンドポイント・システムは、互いの証明書の CA 認証性を認証します。

注: クライアントおよびサーバー認証は、2 つの IBM i システム間でのみ行われます。クライアントが PC の場合、クライアント認証はサーバーによって実行されません。

他のアプリケーションと異なり、マネージメント・セントラルは、トラステッド・グループ妥当性検査リストと呼ばれる妥当性検査リストを通して認証を提供します。一般に、妥当性検査リストには、ユーザーを識別する情報 (たとえば、ユーザー ID) と認証情報 (たとえば、パスワード、個人識別番号、ディジタル証明書) が保管されています。この認証情報は暗号化されています。

大半のアプリケーションでは通常、サーバー認証とクライアント認証の両方を使用可能にすることを指定しません。これは、サーバー認証が、ほとんど常に SSL セッションが使用可能になっている間に発生するためです。多くのアプリケーションには、クライアント認証の構成のオプションがあります。セントラル・システムがネットワークで果たす役割は 2 つあるので、マネージメント・セントラルでは、クライアント認証ではなく、「サーバーおよびクライアント認証」という用語を使用しています。PC ユーザーがセントラル・システムに接続している場合は、セントラル・システムはサーバーとして機能します。しかし、セントラル・システムがエンドポイント・システムに接続する場合、セントラル・システムはクライアントとして機能します。次の図は、セントラル・システムがネットワークでサーバーおよびクライアントとして機能する様子を示したものです。

注: この図では、認証局に関連付けられた証明書は、セントラル・システム、およびすべてのエンドポイント・システム上の鍵データベースに保管する必要があります。認証局は、PC、セントラル・システム、すべてのエンドポイントにある必要があります。



前提条件および前提事項

トムは、マネージメント・セントラル・サーバーへのすべての接続を保護するために、以下の管理タスクおよび構成タスクを行う必要があります。

1. システム A を SSL の前提条件に合わせます。
2. セントラル・システムおよびすべてのエンドポイント・システムは、OS/400® V5R2、または i5/OS V5R3 以降で稼働します。

   注: IBM i 5.4 以降の OS/400 V5R1 システムへの接続は許可されません。
3. PC クライアントは IBM i Access for Windows V5R3 以降の System i ナビゲーター で稼働します。
4. IBM i システムの認証局 (CA) を取得します。
5. システム A 用に CA によって署名された証明書を作成します。
6. CA および証明書をシステム A に送信し、それらを鍵データベースにインポートします。
7. マネージメント・セントラル・アプリケーション ID およびすべての IBM i システムのアプリケーション ID を証明書に割り当てます。 TCP セントラル・サーバー、データベース・サーバー、データ待ち行列サーバー、ファイル・サーバー、ネットワーク・プリント・サーバー、リモート・コマンド・サーバーおよびサインオン・サーバーは、すべて IBM i システムです。
   a. マネージメント・セントラル・サーバーで IBM ディジタル証明書マネージャーを開始します。トムが証明書を取得または作成する必要がある場合、あるいは証明書システムをセットアップまたは変更する必要がある場合には、それをこの時点で行います。
   b. 「**証明書ストアの選択**」を選択します。
   c. 「***SYSTEM**」を選択し、「**続行**」をクリックします。
   d. 「**証明書ストア・パスワード**」に *SYSTEM を入力し、「**続行**」をクリックします。メニューが再ロードされたら、「**アプリケーションの管理**」を展開します。
   e. 「**証明書割り当ての更新**」をクリックします。
   f. 「**サーバー**」を選択し、「**続行**」をクリックします。
   g. 「**マネージメント・セントラル・サーバー**」を選択し、「**証明書割り当ての更新**」をクリックします。これにより、使用するマネージメント・セントラル・システムに証明書が割り当てられます。
   h. アプリケーションに割り当てる証明書を選択し、「**新規証明書の割り当て**」をクリックします。DCM は、「**証明書割り当ての更新**」ページを再ロードして、確認メッセージを表示します。
   i. 「**キャンセル**」をクリックし、アプリケーションのリストに戻ります。
   j. すべての IBM i システムについて、この手順を繰り返します。
8. CA を System i ナビゲーターの PC クライアントにダウンロードします。

構成ステップ:

トムがマネージメント・セントラル・サーバーで SSL を使用可能にするためには、まずその前に前提条件のプログラムをインストールし、セントラル・システムにディジタル証明書をセットアップする必要があります。続行する前に、このシナリオの前提条件と前提事項を参照してください。前提条件を満たしたら、以下の手順を完了させて、マネージメント・セントラル・サーバーですべての接続を保護できます。

注: SSL が System i ナビゲーター で使用可能になっている場合、トムは SSL をマネージメント・セントラル・サーバーで使用可能にする前に、SSL を使用不可にする必要があります。 SSL が System i ナビゲーターで使用可能であり、マネージメント・セントラル・サーバーでは使用可能でない場合は、System i ナビゲーターがセントラル・システムと接続しようとしても、失敗します。

1. 35 ページの『ステップ 1: サーバー認証用にセントラル・システムを構成する』
2. 35 ページの『ステップ 2: サーバー認証用にエンドポイント・システムを構成する』
3. 36 ページの『ステップ 3: セントラル・システム上のマネージメント・セントラル・システムを再始動する』

4. 36 ページの『ステップ 4: すべてのエンドポイント・システム上のマネージメント・セントラル・システムを再始動する』
5. 36 ページの『ステップ 5: System i ナビゲーター クライアントについて SSL をアクティブにする』
6. 37 ページの『ステップ 6: クライアント認証用にセントラル・システムを構成する』
7. 37 ページの『ステップ 7: クライアント認証用にエンドポイント・システムを構成する』
8. 37 ページの『ステップ 8: 妥当性検査リストをエンドポイント・システムにコピーする』
9. 38 ページの『ステップ 9: セントラル・システム上のマネージメント・セントラル・システムを再始動する』
10. 38 ページの『ステップ 10: すべてのエンドポイント・システム上のマネージメント・セントラル・システムを再始動する』

関連概念:

25 ページの『SSL の前提条件』
このトピックでは、IBM i でのシステム SSL の前提条件、およびいくつかの役に立つヒントを示しています。

25 ページの『SSL によるアプリケーション・セキュリティー』
IBM i の多数のアプリケーションを SSL で保護することができます。詳しくは、各アプリケーションの資料を参照してください。

関連タスク:

28 ページの『構成の詳細: SSL によるマネージメント・セントラル・システムへのクライアント接続の保護』
このトピックでは、SSL を使用してマネージメント・セントラル・サーバーへのクライアント接続を保護する拡張された構成ステップについて説明しています。

『構成の詳細: SSL を使用したマネージメント・セントラル・システムへのすべての接続の保護』
このトピックでは、SSL を使用したマネージメント・セントラル・サーバーへのすべての接続の保護に関する詳細について説明しています。

関連情報:

DCM の構成

証明書のはじめてのセットアップ

構成の詳細: SSL を使用したマネージメント・セントラル・システムへのすべての接続の保護

このトピックでは、SSL を使用したマネージメント・セントラル・サーバーへのすべての接続の保護に関する詳細について説明しています。

次の情報は、『シナリオ: SSL によるマネージメント・セントラル・サーバーへのすべてのクライアント接続の保護』に目を通していることを前提としています。

ここで、マネージメント・セントラル・サーバーに対するすべての接続をセキュアにするのに必要なステップを実行する方法を理解します。トムがシナリオを完了するのを追っていきます。

トムがマネージメント・セントラル・システムで SSL を使用可能にするためには、まずその前に前提条件のプログラムをインストールし、IBM i にディジタル証明書をセットアップする必要があります。前提条件を満たしたら、以下の手順を完了させて、マネージメント・セントラル・サーバーですべての接続を保護できます。

注: SSL が System i ナビゲーター で使用可能になっている場合、トムは SSL をマネージメント・セントラル・サーバーで使用可能にする前に、SSL を使用不可にする必要があります。 SSL が System i ナビゲーター で使用可能であり、マネージメント・セントラル・サーバーでは使用可能でない場合は、System i ナビゲーター がセントラル・システムと接続しようとしても、失敗します。

トムは SSL を使用することで、セントラル・システムとエンドポイント・システムの間の伝送、および System i ナビゲーター クライアントとセントラル・システムの間の伝送をセキュアにすることができます。SSL では、証明書の移送と認証、およびデータの暗号化を行うことができます。SSL 接続が可能なのは、SSL が使用可能なセントラル・システムと SSL が使用可能なエンドポイント・システムの間だけです。トムは、クライアントの認証を構成する前に、サーバーの認証を構成する必要があります。

関連概念:

25 ページの『SSL の前提条件』
このトピックでは、IBM i でのシステム SSL の前提条件、およびいくつかの役に立つヒントを示しています。

30 ページの『シナリオ: SSL によるマネージメント・セントラル・サーバーへのすべての接続の保護』
このシナリオは、SSL を使用して IBM i とのすべての接続を保護する方法を説明しています。IBM i は System i ナビゲーターのマネジメント・セントラル・システムを使用して、セントラル・システムとして機能しています。

関連情報:

証明書のはじめてのセットアップ

ステップ 1: サーバー認証用にセントラル・システムを構成する:

1. System i ナビゲーター で、「**マネージメント・セントラル**」を右クリックし、「**プロパティー**」を選択します。
2. 「**セキュリティー**」タブをクリックし、「**Secure Sockets Layer (SSL) を使用**」を選択します。
3. 認証レベルとして「**サーバー**」を選択します。
4. 「**OK**」をクリックして、この値をセントラル・システムに設定します。

   注: 指示があるまで、マネージメント・セントラル・サーバーを再始動**しないでください**。ここでサーバーを再始動した場合、エンドポイント・サーバーに接続できません。SSL を活動化してサーバーを再始動する前に、さらに構成タスクを完了する必要があります。最初に、比較および更新タスクで、エンドポイント・システムへ SSL 構成を伝搬する必要があります。

ステップ 2: サーバー認証用にエンドポイント・システムを構成する:
トムは、セントラル・システムでサーバー認証を構成した後に、すべてのエンドポイント・システムにサーバー認証を構成する必要があります。次のタスクを実行します。

1. 「**マネージメント・セントラル**」を展開します。
2. エンドポイント・システムのシステム値を比較および更新します。
   a. 「**エンドポイント・システム**」において、「セントラル・システム」を右クリックし、「**インベントリー**」 > 「**収集**」の順に選択します。
   b. セントラル・システムで使用しているシステム値のインベントリーを収集するために、「収集」ダイアログ・ボックスで「**システム値**」オプションをチェックします。他のオプションを選択解除します。「OK」をクリックし、インベントリー・タスクが完了するまで待機します。
   c. 「**システム・グループ**」 > 「**新規システム・グループ**」の順に右クリックします。
   d. SSL を使用して、接続するすべてのエンドポイント・システムを含む新規のシステム・グループを定義します。この新規システム・グループに「トラステッド・グループ」という名を付けます。

e. 新規グループ「トラステッド・グループ」を表示するには、システム・グループのリストを展開します。

f. 収集が完了した後に、新規のシステム・グループを右クリックして、「**システム値**」 > 「**比較および更新**」と選択します。

g. 「**モデル・システム**」フィールドにセントラル・システムが表示されていることを確認します。

h. 「**カテゴリー**」フィールドで、「**マネージメント・セントラル**」を選択します。

i. 「**Secure Sockets Layer (SSL) を使用**」が「**はい**」に設定されているかを確認し、「**更新**」を選択して、この値を「トラステッド・グループ」に伝搬します。

j. 「**SSL 認証レベル**」が「**サーバー**」に設定されているかを確認して、「**更新**」を選択し、この値を「トラステッド・グループ」に伝搬します。

注: これらの値を設定していない場合は、『ステップ 1: サーバー認証用にセントラル・システムを構成する』を完了してください。

k. 「**OK**」をクリックします。「**比較および更新**」で処理が完了するまで待機してから、次のステップに進んでください。

ステップ 3: セントラル・システム上のマネージメント・セントラル・システムを再始動する:

1. System i ナビゲーターで、「**ユーザー接続**」を展開します。
2. セントラル・システムを展開します。
3. 「**ネットワーク**」 > 「**サーバー**」の順に展開し、「**TCP/IP**」を選択します。
4. 「**マネージメント・セントラル**」を右クリックし、「**停止**」を選択します。「セントラル・システム (central system)」ビューは縮小表示され、サーバーには接続されていないという内容のメッセージが表示されます。
5. マネージメント・セントラル・サーバーが停止したら、「**開始**」をクリックして、再始動します。

ステップ 4: すべてのエンドポイント・システム上のマネージメント・セントラル・システムを再始動する:

1. System i ナビゲーターで、「**ユーザー接続**」を展開します。
2. 再始動するエンドポイント・システムを展開します。
3. 「**ネットワーク**」 > 「**サーバー**」の順に展開し、「**TCP/IP**」を選択します。
4. 「**マネージメント・セントラル**」を右クリックし、「**停止**」を選択します。
5. マネージメント・セントラル・サーバーが停止したら、「**開始**」をクリックして、再始動します。
6. それぞれのエンドポイント・システムについて、この手順を繰り返します。

ステップ 5: System i ナビゲーター クライアントについて SSL をアクティブにする:

1. System i ナビゲーターで、「**ユーザー接続**」を展開します。
2. セントラル・システムを右クリックし、「**プロパティー**」を選択します。
3. 「**セキュア・ソケット**」タブをクリックし、「**SSL (Secure Sockets Layer) を接続に使用**」を選択します。
4. System i ナビゲーターを終了し、再始動します。

注: これらのステップを完了した後、サーバー認証がセントラル・システムおよびエンドポイント・システムで構成されます。同様に、クライアント認証のセントラル・システムおよびエンドポイント・システムをオプションで構成することができます。セントラル・システムおよびエンドポイント・システムでクライアント認証を使用可能にする場合は、ステップ 6 から 10 までを完了してください。

ステップ 6: クライアント認証用にセントラル・システムを構成する:
これで、トムはサーバー認証用の構成を終了したので、以下のオプションのクライアント認証手順の実行を選択できます。クライアント認証では、エンドポイント・システムとセントラル・システムの両方について、認証局とトラステッド・グループの妥当性検査を行います。セントラル・システム (SSL クライアント) が SSL を使用してエンドポイント・システム (SSL サーバー) に接続しようとした場合、セントラル・システムとエンドポイント・システムは、サーバー認証とクライアント認証により互いの証明書を認証します。また、これは、認証局 (CA) とトラステッド・グループの認証と呼ばれます。

注: サーバーの認証を構成するまで、クライアントの認証の構成は完了できません。サーバー認証を構成していない場合は、前に戻って構成してください。

1. System i ナビゲーター で、「**マネージメント・セントラル**」を右クリックし、「**プロパティー**」を選択します。
2. 「**セキュリティー**」タブをクリックし、「**Secure Sockets Layer (SSL) を使用**」を選択します。
3. 認証レベルの「**クライアントおよびサーバー**」を選択します。
4. 「**OK**」をクリックして、この値をセントラル・システムに設定します。

   注: 指示があるまで、マネージメント・セントラル・サーバーを再始動**しないでください**。ここでサーバーを再始動した場合、エンドポイント・サーバーに接続できません。SSL を活動化してサーバーを再始動する前に、さらに構成タスクを完了する必要があります。最初に、比較および更新タスクで、エンドポイント・システムへ SSL 構成を伝搬する必要があります。

ステップ 7: クライアント認証用にエンドポイント・システムを構成する:
エンドポイント・システムのシステム値を比較および更新します。

1. 「**マネージメント・セントラル**」を展開します。
2. エンドポイント・システムのシステム値を比較および更新します。
   a. 「**エンドポイント・システム**」において、「セントラル・システム」を右クリックし、「**インベントリー**」 > 「**収集**」の順に選択します。
   b. セントラル・システムで使用しているシステム値のインベントリーを収集するために、「収集」ダイアログ・ボックスで「**システム値**」オプションをチェックします。他のオプションを選択解除します。「OK」をクリックし、インベントリー・タスクが完了するまで待機します。
   c. 収集が完了した後に、「トラステッド・グループ」を右クリックして、「**システム値**」 > 「**比較および更新**」と選択します。
   d. 「**モデル・システム**」フィールドにセントラル・システムが表示されていることを確認します。
   e. 「**カテゴリー**」フィールドで、「**マネージメント・セントラル**」を選択します。
   f. 「**Secure Sockets Layer (SSL) を使用**」が「**はい**」に設定されているかを確認し、「**更新**」を選択して、この値を「トラステッド・グループ」に伝搬します。
   g. 「**SSL 認証レベル**」が「**クライアントおよびサーバー**」に設定されているかを確認して、「**更新**」を選択し、この値を「トラステッド・グループ」に伝搬します。

      注: これらの値を設定していない場合は、『ステップ 6: クライアント認証用にセントラル・システムを構成する』を完了してください。
   h. 「**OK**」をクリックします。「**比較および更新**」で処理が完了するまで待機してから、次のステップに進んでください。

ステップ 8: 妥当性検査リストをエンドポイント・システムにコピーする:
このタスクは、ご使用のセントラル・システムがIBM i V5R3 以上であることを前提しています。OS/400 V5R2 以前のシステムでは、QYPSVLDL.VLDL は QMGTC2.LIB ではなく QUSRSYS.LIB にありました。

したがって、ご使用のシステムが V5R3 よりも前の場合、妥当性検査リストをそのシステムに送信して、QMGTC2.LIB ではなく QUSRSYS.LIB にセットする必要があります。V5R3 以上のシステムの場合、以下のステップを続行してください。

1. System i ナビゲーターで、「**マネージメント・セントラル**」 > 「**定義**」の順に展開します。
2. 「**パッケージ**」を右クリックし、「**新規定義**」を選択します。
3. 「**新規定義**」ウィンドウで、以下のものについての作業を行います。
   a. **名前:** 定義名を入力する。
   b. **ソース・システム:** セントラル・システム名を選択する。
   c. **選択されているファイルとフォルダー:** フィールド内をクリックし、/QSYS.LIB/QMGTC2.LIB/QYPSVLDL.VLDL と入力する。
4. 「**オプション**」タブをクリックし、「**既存のファイルを送信中のファイルで置き換える**」を選択します。
5. 「**拡張**」をクリックします。
6. 「**拡張オプション**」ウィンドウで、「**はい**」を指定して、復元時にオブジェクトの違いが許されるようにし、「**ターゲット・リリース**」をエンドポイントの最初のリリースに変更します。
7. 「**OK**」をクリックして、定義のリストを最新表示し、新規のパッケージを表示します。
8. 新規パッケージを右クリックし、「**送信**」を選択します。
9. 「**送信**」ダイアログ・ボックスで、「**使用可能なシステムおよびグループ**」リストから、「**システム・グループ**」->「**トラステッド・グループ**」を展開します。このグループは、 35 ページの『ステップ 2: サーバー認証用にエンドポイント・システムを構成する』で定義されたものです。

   注: セントラル・システムは常にソース・システムであるため、「**送信**」タスクは、セントラル・システムでは常に失敗します。「**送信**」タスクは、すべてのエンドポイント・システムで正常に完了するはずです。
10. 「**トラステッド・グループ**」に、IBM i V5R3 よりも前のシステムがある場合、手動でそれらのシステムに進み、QYPSVLDL.VLDL オブジェクトを QMGTC2.LIB から QUSRSYS.LIB に移動します。QUSRSYS.LIB にすでに QYPSVLDL.VLDL のバージョンがある場合、それを削除し、QMGTC2.LIB の新規のバージョンと置き換えます。

ステップ 9: セントラル・システム上のマネージメント・セントラル・システムを再始動する:

1. System i ナビゲーターで、「**ユーザー接続**」を展開します。
2. セントラル・システムを展開します。
3. 「**ネットワーク**」 > 「**サーバー**」の順に展開し、「**TCP/IP**」を選択します。
4. 「**マネージメント・セントラル**」を右クリックし、「**停止**」を選択します。「セントラル・システム (central system)」ビューは縮小表示され、サーバーには接続されていないという内容のメッセージが表示されます。
5. マネージメント・セントラル・サーバーが停止したら、「**開始**」をクリックして、再始動します。

ステップ 10: すべてのエンドポイント・システム上のマネージメント・セントラル・システムを再始動する:

注: それぞれのエンドポイント・システムについて、この手順を繰り返します。

1. System i ナビゲーターで、「**ユーザー接続**」を展開します。
2. 再始動するエンドポイント・システムを展開します。
3. 「**ネットワーク**」 > 「**サーバー**」の順に展開し、「**TCP/IP**」を選択します。

4. 「**マネージメント・セントラル**」を右クリックし、「**停止**」を選択します。
5. マネージメント・セントラル・サーバーが停止したら、「**開始**」をクリックして、再始動します。

SSL のトラブルシューティング

これはごく基本のトラブルシューティング情報であり、 IBM i プラットフォームが SSL を使用する際に発生する可能性のある問題のリストを削減することを目的としています。

ただし、トラブルシューティングに関する包括的な情報源ではなく、共通の問題解決に役立つ手引きである点にご注意ください。

以下の内容に当てはまることを確認します。

- IBM i プラットフォーム上での SSL の前提条件を満たしている。
- 使用している認証局および証明書は有効であり、有効期限が切れていない。

前述の内容がご使用のシステムに当てはまることを確認しても、依然として SSL 関連の問題がある場合は、オプションで以下を試行してください。

- エラーに関する詳細については、サーバーのジョブ・ログにある SSL のエラー・コードをエラー・テーブルで相互参照することができます。たとえば、このテーブルではサーバーのジョブ・ログに示された -93 は、定数 SSL_ERROR_SSL_NOT_AVAILABLE にマップされます。
  - 負の戻りコード (コード番号の前にあるダッシュで表される) は、SSL_ API を使用していることを表します。
  - 正の戻りコードは、GSKit API を使用していることを表します。プログラマーは、プログラム内で gsk_strerror() API または SSL_Strerror() API をコーディングして、エラーの戻りコードの要旨を取得することができます。一部のアプリケーションはこの API を使用し、メッセージをこのセンテンスを含むジョブ・ログへ出力します。

  より詳細な情報が必要な場合は、 この表にあるメッセージ ID を IBM i に表示して、 このエラーの潜在的な原因とリカバリーを示すことができます。これらのエラー・コードに関するその他の説明は、エラーを戻した個々のセキュア・ソケット API 内で見つかる場合もあります。
- 現在のセキュア・セッションにおける最後の証明書検証エラーに関する追加情報は、gsk_attribute_get_numeric_value() で GSK_LAST_VALIDATION_ERROR 属性を使用することにより検索できます。 gsk_secure_soc_init() または gsk_secure_soc_startInit() がエラーを返した場合、この属性で、より詳細なエラー情報が提供されることがあります。
- 以下の 2 つのヘッダー・ファイルには、テーブルに存在するものと同じシステム SSL の戻りコードの定数名が存在しますが、相互参照のためのメッセージ ID は存在しません。
  - QSYSINC/H.GSKSSL
  - QSYSINC/H.QSOSSL

  システム SSL の戻りコードの名前はこれらの 2 つのファイル内では変化しませんが、それぞれの戻りコードには複数の固有のエラーが関連する場合があります。

関連概念:

25 ページの『SSL の前提条件』
このトピックでは、IBM i でのシステム SSL の前提条件、およびいくつかの役に立つヒントを示しています。

関連情報:

セキュア・ソケット API のエラー・コード・メッセージ

SSL の関連情報

この情報を使用して、Secure Sockets Layer (SSL) の使用に関連する他のリソースおよび情報を学習します。

Web サイト

- RFC 5246:「Transport Layer Security (TLS) プロトコル・バージョン 1.2 (英語)」 (http://www.ietf.org/rfc/rfc5246.txt)

  TLS プロトコル・バージョン 1.2 について詳細に説明しています。

- RFC 4346:「Transport Layer Security (TLS) プロトコル・バージョン 1.1 (英語)」 (http://www.ietf.org/rfc/rfc4346.txt)

  TLS プロトコル・バージョン 1.1 について詳細に説明しています。

- RFC 2246:「TLS プロトコル・バージョン 1.0 (英語)」 (http://www.ietf.org/rfc/rfc2246.txt)

  TLS プロトコル・バージョン 1.0 について詳細に説明しています。

- RFC2818:「HTTP Over TLS (英語)」 (http://www.ietf.org/rfc/rfc2818.txt)

  TLS を使用してインターネットで HTTP 接続をセキュアにする方法について説明しています。

- RFC 5746:「Transport Layer Security (TLS) 再ネゴシエーション表示拡張 (英語)」 (http://www.ietf.org/rfc/rfc5746.txt)

  TLS 再ネゴシエーション表示拡張を定義しています。

- RFC 6066:「Transport Layer Security (TLS) 拡張: 拡張の定義 (英語)」 (http://www.ietf.org/rfc/rfc6066.txt)

  TLS 拡張を定義しています。

- RFC 2560:「X.509 インターネット PKI Online Certificate Status Protocol (OCSP) (英語)」 (http://www.ietf.org/rfc/rfc2560.txt)

  デジタル証明書の失効状況を判別するために使用する OCSP を定義しています。

- RFC 4492:「Transport Layer Security (TLS) の楕円曲線暗号 (ECC) 暗号スイート (英語)」 (http://www.ietf.org/rfc/rfc4492.txt)

  TLS の新しい鍵交換アルゴリズムを定義しています。

その他の情報

- SSL および Java セキュア・ソケット拡張機能
- IBM Toolbox for Java

特記事項

本書は米国 IBM が提供する製品およびサービスについて作成したものです。

本書に記載の製品、サービス、または機能が日本においては提供されていない場合があります。日本で利用可能な製品、サービス、および機能については、日本 IBM の営業担当員にお尋ねください。本書で IBM 製品、プログラム、またはサービスに言及していても、その IBM 製品、プログラム、またはサービスのみが使用可能であることを意味するものではありません。これらに代えて、IBM の知的所有権を侵害することのない、機能的に同等の製品、プログラム、またはサービスを使用することができます。ただし、IBM 以外の製品とプログラムの操作またはサービスの評価および検証は、お客様の責任で行っていただきます。

IBM は、本書に記載されている内容に関して特許権 (特許出願中のものを含む) を保有している場合があります。本書の提供は、お客様にこれらの特許権について実施権を許諾することを意味するものではありません。実施権についてのお問い合わせは、書面にて下記宛先にお送りください。

〒103-8510
東京都中央区日本橋箱崎町19番21号
日本アイ・ビー・エム株式会社
法務・知的財産
知的財産権ライセンス渉外

以下の保証は、国または地域の法律に沿わない場合は、適用されません。 IBM およびその直接または間接の子会社は、本書を特定物として現存するままの状態で提供し、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任を負わないものとします。国または地域によっては、法律の強行規定により、保証責任の制限が禁じられる場合、強行規定の制限を受けるものとします。

この情報には、技術的に不適切な記述や誤植を含む場合があります。本書は定期的に見直され、必要な変更は本書の次版に組み込まれます。IBM は予告なしに、随時、この文書に記載されている製品またはプログラムに対して、改良または変更を行うことがあります。

本書において IBM 以外の Web サイトに言及している場合がありますが、便宜のため記載しただけであり、決してそれらの Web サイトを推奨するものではありません。それらの Web サイトにある資料は、この IBM 製品の資料の一部ではありません。それらの Web サイトは、お客様の責任でご使用ください。

IBM は、お客様が提供するいかなる情報も、お客様に対してなんら義務も負うことのない、自ら適切と信ずる方法で、使用もしくは配布することができるものとします。

本プログラムのライセンス保持者で、(i) 独自に作成したプログラムとその他のプログラム (本プログラムを含む) との間での情報交換、および (ii) 交換された情報の相互利用を可能にすることを目的として、本プログラムに関する情報を必要とする方は、下記に連絡してください。

IBM Corporation
Software Interoperability Coordinator, Department YBWA
3605 Highway 52 N
Rochester, MN 55901
U.S.A.

本プログラムに関する上記の情報は、適切な使用条件の下で使用することができますが、有償の場合もあります。

本書で説明されているライセンス・プログラムまたはその他のライセンス資料は、IBM 所定のプログラム契約の契約条項、IBM プログラムのご使用条件、またはそれと同等の条項に基づいて、IBM より提供されます。

この文書に含まれるいかなるパフォーマンス・データも、管理環境下で決定されたものです。そのため、他の操作環境で得られた結果は、異なる可能性があります。一部の測定が、開発レベルのシステムで行われた可能性がありますが、その測定値が、一般に利用可能なシステムのものと同じである保証はありません。さらに、一部の測定値が、推定値である可能性があります。実際の結果は、異なる可能性があります。お客様は、お客様の特定の環境に適したデータを確かめる必要があります。

IBM 以外の製品に関する情報は、その製品の供給者、出版物、もしくはその他の公に利用可能なソースから入手したものです。 IBM は、それらの製品のテストは行っておりません。したがって、他社製品に関する実行性、互換性、またはその他の要求については確証できません。 IBM 以外の製品の性能に関する質問は、それらの製品の供給者にお願いします。

IBM の将来の方向または意向に関する記述については、予告なしに変更または撤回される場合があり、単に目標を示しているものです。

本書はプランニング目的としてのみ記述されています。記述内容は製品が使用可能になる前に変更になる場合があります。

本書には、日常の業務処理で用いられるデータや報告書の例が含まれています。より具体性を与えるために、それらの例には、個人、企業、ブランド、あるいは製品などの名前が含まれている場合があります。これらの名称はすべて架空のものであり、名称や住所が類似する企業が実在しているとしても、それは偶然にすぎません。

著作権使用許諾:

本書には、様々なオペレーティング・プラットフォームでのプログラミング手法を例示するサンプル・アプリケーション・プログラムがソース言語で掲載されています。お客様は、サンプル・プログラムが書かれているオペレーティング・プラットフォームのアプリケーション・プログラミング・インターフェースに準拠したアプリケーション・プログラムの開発、使用、販売、配布を目的として、いかなる形式においても、IBM に対価を支払うことなくこれを複製し、改変し、配布することができます。このサンプル・プログラムは、あらゆる条件下における完全なテストを経ていません。従って IBM は、これらのサンプル・プログラムについて信頼性、利便性もしくは機能性があることをほのめかしたり、保証することはできません。これらのサンプル・プログラムは特定物として現存するままの状態で提供されるものであり、いかなる保証も提供されません。 IBM は、お客様の当該サンプル・プログラムの使用から生ずるいかなる損害に対しても一切の責任を負いません。

それぞれの複製物、サンプル・プログラムのいかなる部分、またはすべての派生的創作物にも、次のように、著作権表示を入れていただく必要があります。

© (お客様の会社名) (西暦年). このコードの一部は、IBM Corp. のサンプル・プログラムから取られています。

© Copyright IBM Corp. _年を入れる_.

商標

IBM、IBM ロゴおよび ibm.com は、世界の多くの国で登録された International Business Machines Corporation の商標です。他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。現時点での IBM の商標リストについては、『www.ibm.com/legal/copytrade.shtml』 をご覧ください。

Adobe、Adobe ロゴ、PostScript、PostScript ロゴは、Adobe Systems Incorporated の米国およびその他の国における登録商標または商標です。

Java およびすべての Java 関連の商標およびロゴは Oracle やその関連会社の米国およびその他の国における商標または登録商標です。

他の製品名およびサービス名等は、それぞれ IBM または各社の商標である場合があります。

使用条件

これらの資料は、以下の条件に同意していただける場合に限りご使用いただけます。

個人使用: これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、非商業的な個人による使用目的に限り複製することができます。ただし、IBM の明示的な承諾をえずに、これらの資料またはその一部について、二次的著作物を作成したり、配布 (頒布、送信を含む) または表示 (上映を含む) することはできません。

商業的使用: これらの資料は、すべての著作権表示その他の所有権表示をしていただくことを条件に、お客様の企業内に限り、複製、配布、および表示することができます。 ただし、IBM の明示的な承諾をえずにこれらの資料の二次的著作物を作成したり、お客様の企業外で資料またはその一部を複製、配布、または表示することはできません。

ここで明示的に許可されているもの以外に、資料や資料内に含まれる情報、データ、ソフトウェア、またはその他の知的所有権に対するいかなる許可、ライセンス、または権利を明示的にも黙示的にも付与するものではありません。

資料の使用が IBM の利益を損なうと判断された場合や、上記の条件が適切に守られていないと判断された場合、IBM はいつでも自らの判断により、ここで与えた許可を撤回できるものとさせていただきます。

お客様がこの情報をダウンロード、輸出、または再輸出する際には、米国のすべての輸出入関連法規を含む、すべての関連法規を遵守するものとします。

IBM は、これらの資料の内容についていかなる保証もしません。これらの資料は、特定物として現存するままの状態で提供され、商品性の保証、特定目的適合性の保証および法律上の瑕疵担保責任を含むすべての明示もしくは黙示の保証責任なしで提供されます。

プログラム番号: 5770-SS1

日本アイ・ビー・エム株式会社
〒103-8510 東京都中央区日本橋箱崎町19-21