CHAPTER 1

概要

（注） この章の情報は、特に記載のない限り、ACE モジュールと ACE アプライアンスの両方に適用されます。この章で説明する機能は、特に記載のない限り、IPv4 と IPv6 に適用されます。

Secure Sockets Layer（SSL）は、インターネットの暗号化テクノロジーを提供するアプリケーション層プロトコルです。SSL は、プライバシー、認証、およびデータ整合性を組み合わせることで、クライアントとサーバの間のデータ転送のセキュリティを確保します。SSL は、このアプリケーション レベルのセキュリティの実現に、証明書および秘密キーと公開キーのキー交換ペアを使用します。

この章の内容は、次のとおりです。

SSL 暗号化の概要

Cisco ACE アプリケーション コントロール エンジン は特別なセットの SSL コマンドを使用して、クライアントとサーバ間の SSL 暗号化機能を実行します。SSL 機能には、サーバ認証、秘密キーおよび公開キー生成、証明書管理、データ パケット暗号化および復号化が含まれます。

ACE は、SSL バージョン 3.0 および Transport Layer Security（TLS）バージョン 1.0 をサポートします。ACE は、ハイブリッド 2/3 hello メッセージと呼ばれる SSL バージョン 2.0 ClientHello メッセージを理解して受け取るため、デュアルバージョンのクライアントによる ACE との通信が可能になります。クライアントがバージョン 2.0 ClientHello で SSL バージョン 3.0 を示している場合、ACE は、クライアントが SSL バージョン 3.0 をサポートできると理解して、バージョン 3.0 ServerHello メッセージを返します。

（注）　クライアントが SSL バージョン 2.0 だけをサポートする場合、ACE はネットワーク トラフィックを渡すことができません。

ACE の一般的な SSL セッションでは、セキュア接続を確立して維持するために暗号方式が必要です。暗号スイートは、キー交換、認証、メッセージ認証コード（MAC）を実行するために ACE が必要とする暗号化アルゴリズムを提供します。サポートされる暗号スイートの詳細については、第 3 章 「SSL 終了の設定」の「暗号スイートの追加」の項を参照してください。

ここでは、ACE で実装される SSL 暗号化の概要について説明します。次のトピックが含まれます。

- SSL 公開キー インフラストラクチャ
- SSL ハンドシェイク

SSL 公開キー インフラストラクチャ

SSL は、公開キー インフラストラクチャ（PKI）の認証、暗号化、およびデータの整合性を提供します。PKI はデバイス間のセキュアな情報交換を確立する一連のポリシーおよび手順です。非対称暗号化で使用される PKI は、3 種類の基本要素で特徴付けられます。

- 機密性
- 認証
- メッセージ整合性

これらの 3 つの要素は、社内イントラネットからインターネット ベースの e- ビジネス アプリケーションまで、事実上すべてのタイプの電子商取引を構築することを目的として、e- コマースおよび信頼できる環境を導入するためのセキュアなシステムを提供します。

機密性

*機密性*によって、意図しないユーザがデータを参照できないようにします。PKI では、機密性はさまざまな方法でデータを暗号化することによって実現されます。SSL では特に、2 つのエンドポイントでのみ認識されている 1 つ以上の対称キーを使用して、大量のデータが暗号化されます。対称キーは通常、1 つのエンドポイントによって生成されるため、もう一方のエンドポイントに安全に転送する必要があります。ACE では、ACE とその SSL ピア間の対称キーの安全な転送のために、*キー交換*メカニズムの使用がサポートされます。

キー交換では、一方のデバイスが対称キーを生成し、これを自身の SSL ピアに転送する前に非対称暗号化スキームを使用して暗号化します。非対称暗号化では、各デバイスに公開キーと秘密キーで構成される一意のキー ペアが存在する必要があります。2 つのキーは数学的に関連付けられます。公開キー / 秘密キーを使用して暗号化されるデータは、対応する秘密キー / 公開キーを使用した場合だけ復号化できます。デバイスは自身の SSL ピアと公開キーを共有しますが、秘密キーは公開せずにおく必要があります。

非対称暗号化のセキュリティは、オーナー以外のいかなる当事者にも秘密キーが知られていないという事実に全面的に依存しています。何らかの理由でこのキーのセキュリティが損なわれた場合、不正な Web ユーザ（または Web サイト）によって、対称キーとデータ転送全体を含むストリームが復号化される可能性があります。最も一般的に使用されるキー交換アルゴリズムは、Rivest Shamir Adelman（RSA）アルゴリズムです。

SSL の場合、受信者の秘密キーが転送を復号化できる唯一のキーとなるようにするために、送信者は受信者の公開キーで対称キーを暗号化します。

認証

*認証*は、交換において 1 つ以上のデバイスが、他のデバイスの ID を確認できるようにします。たとえば、クライアントが e- コマース Web サイトに接続しているとします。クレジット カード番号などの機密情報を送信する前に、クライアントはサーバが正当な e- コマース Web サイトであることを確認します。トランザクションを開始する前に、クライアントとサーバの両方で相互認証が必要な場合があります。2 つの銀行間の金融トランザクションでは、クライアントおよびサーバの両方で相手のアイデンティティを認識している必要があります。SSL はデジタル証明書を使用して、この認証を容易にします。

デジタル証明書とは、サーバの ID をクライアントに証明する、またはオプションでクライアントの ID をサーバに証明するデジタル識別の一形式のことです。証明書は、識別情報が正しいことと、証明書に埋め込まれた公開キーが実際にそのクライアントまたはサーバに属することを確認します。

認証局（CA）は、セキュリティを保障するために、公開キーと秘密キーの暗号化を使用する PKI コンテキストでデジタル証明書を発行します。CA は、認証を検証するために証明書に署名する信頼できる認証局です。デジタル証明書には、次の情報が含まれます。

- オーナーの詳細（証明書サブジェクト）
- CA の詳細（証明書の発行元）
- 証明書サブジェクトの公開キー
- 証明書の有効性と有効期限
- 証明書に関連付けられた権限

証明書の発行元として、CA は秘密キーを使用して証明書に署名します。証明書を受け取ると、クライアントは発行元の公開キーを使用して証明書の署名を復号化および確認します。この手順により、証明書が権限のある機関によって実際に発行され署名されていることを確認します。

公開キー証明書は*証明書の階層*をサポートします。CA は、署名付き証明書を発行する責任を共有する子機関の階層を作成します。階層の最上位にある CA は*ルート認証局*と呼ばれます。階層の各レベルがその下のレベルを認証し、*証明書チェーン*と呼ばれる信頼関係の階層を作成します。このプロセスにより、証明書を確認しているエンティティは、必要に応じて、ルート認証局まで遡って証明書を追跡し、その証明書によって信頼される階層内の CA を特定できます。

証明書は、期限切れになるまで、または CA によって取り消されるまで有効です。CA が証明書を取り消すと、その CA が以前に発行して無効になったすべての証明書が含まれる証明書失効リスト（CRL）にその証明書が追加されます。

ACE に接続されたクライアントまたはサーバは、同じ CA から、または信頼関係の階層に含まれる異なる CA（例：A は B を信頼し、B は C を信頼している場合、A は C を信頼している）から発行された信頼できる証明書を備えている必要があります。

メッセージ整合性

*メッセージ整合性*は、メッセージ受信者に対し、メッセージの内容が送信中に改ざんされていないことを保証します。メッセージ整合性を保障するために、SSL はデータを送信する前にメッセージ ダイジェストを適用します。メッセージ ダイジェストは、任意の長さのメッセージを取得し、そのメッセージの特性を示す固定長の文字列を出力します。

メッセージ ダイジェストの重要な特性は、反転が非常に困難であることです。メッセージを送信する前にそのメッセージのダイジェストを付けるだけでは、整合性を確保するために十分ではありません。攻撃者がメッセージを変更して、その内容に応じてダイジェストを変更する可能性があります。

SSL ピア間で交換される各メッセージは、SHA または MD5 などのハッシュ アルゴリズムを使用して計算できるメッセージ認証コード（MAC）によって保護されています。MAC は、秘密の値、送信される実際のデータ、およびシーケンス番号を含む、複数データのハッシュ値です。シークレット値は書き込みセッション キーです。シーケンス番号は 32 ビット カウンタ値です。このデータは、MAC を取得するために、ハッシュ アルゴリズムによって処理されます。メッセージの受信時に、受信者は読み取りセッション キーと予測されるシーケンス番号を使用して MAC を確認し、受信データに対してハッシュを計算します。2 つのハッシュ値が一致しない場合、データ ストリームは何らかの形で変更されています。

SSL ハンドシェイク

クライアントおよびサーバは、SSL ハンドシェイク プロトコルを使用して、2 つのデバイス間で SSL セッションを確立します。ハンドシェイク時に、クライアントおよびサーバは、安全なセッション中に使用する SSL パラメータをネゴシエートします。図 1-1 に、SSL ハンドシェイク時に発生するクライアントおよびサーバのアクションを示します。

（注） ACE では、SSL やその他の終端（プロキシ）接続がアクティブ コンテキストからスタンバイ コンテキストに複製されることはありません。

図 1-1　SSL ハンドシェイク

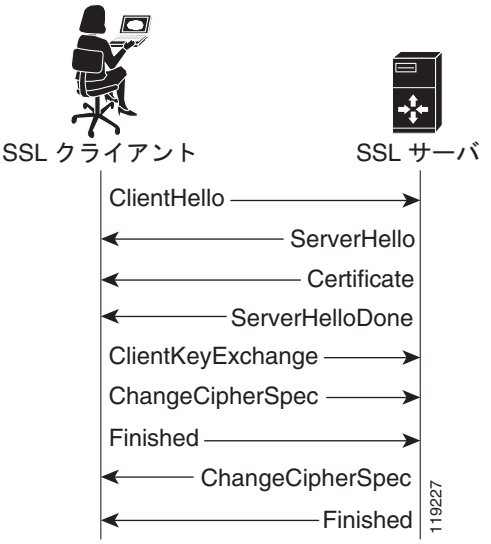


表 1-1 では、SSL ハンドシェイク時にクライアントとサーバ間で実行されるアクションについて説明します。

表 1-1　SSL ハンドシェイク アクション

ステップ	メッセージ	アクション
1	ClientHello	クライアントが SSL セッション中にが使用する SSL パラメータを提案する ClientHello メッセージを送信してハンドシェイクを開始します。
2	ServerHello	サーバは SSL セッション時に使用する SSL パラメータを含む ServerHello メッセージで応答します。
3	Certificate	サーバは、クライアントに公開キー証明書を送信します。
4	ServerHelloDone	サーバは SSL ネゴシエーションの一部を完了します。
5	ClientKeyExchange	クライアントは、サーバの公開キーを使用して暗号化されたセッション キー情報を送信します。

表 1-1　SSL ハンドシェイク アクション （続き）

ステップ	メッセージ	アクション
6	ChangeCipherSpec	クライアントは、サーバが将来送信するメッセージすべてに対して、ネゴシエートされた SSL パラメータをアクティブ化するようにサーバに指示します。
7	Finished	クライアントは、SSL ネゴシエーションが成功したことを確認するようにサーバに指示します。
8	ChangeCipherSpec	サーバは、クライアントが将来送信するメッセージすべてに対して、ネゴシエートされた SSL パラメータをアクティブ化するようにクライアントに指示します。
9	Finished	サーバは、SSL ネゴシエーションが成功したことを確認するようにクライアントに指示します。

ACE SSL 機能

表 1-2 では ACE の SSL 機能について説明します。

表 1-2　ACE SSL 機能

SSL 機能	ACE でサポートされる機能タイプまたは仕様
SSL バージョン	• SSL バージョン 3.0 および Transport Layer Security（TLS）バージョン 1.0 • SSL バージョン 2.0 ClientHello メッセージ（ハイブリッド 2/3 hello）
公開キー交換アルゴリズム	RSA：512 ビット、768 ビット、1024 ビット、1536 ビット、2048 ビット
暗号化タイプ	• データ暗号規格（DES） • Triple-Strength データ暗号規格（3DES） • RC4 • AES

表 1-2　ACE SSL 機能 （続き）

SSL 機能	ACE でサポートされる機能タイプまたは仕様
ハッシュ タイプ	• SSL MAC-MD5 • SSL MAC-SHA1 • SHA-224 • SHA-256 • SHA-384 • SHA-512
暗号スイート	• RSA_WITH_RC4_128_MD5 • RSA_WITH_RC4_128_SHA • RSA_WITH_DES_CBC_SHA • RSA_WITH_3DES_EDE_CBC_SHA • RSA_EXPORT_WITH_RC4_40_MD5 • RSA_EXPORT_WITH_DES40_CBC_SHA • RSA_EXPORT1024_WITH_RC4_56_MD5 • RSA_EXPORT1024_WITH_DES_CBC_SHA • RSA_EXPORT1024_WITH_RC4_56_SHA • RSA_WITH_AES_128_CBC_SHA • RSA_WITH_AES_256_CBC_SHA
デジタル証明書	認証局（CA）の主要なデジタル証明書すべてをサポートします。次のものが含まれます。 • VeriSign • Entrust • Netscape iPlanet • Windows 2000 Certificate Server • Thawte • Equifax • Genuity
証明書の最大数	4096

表 1-2　ACE SSL 機能 （続き）

SSL 機能	ACE でサポートされる機能タイプまたは仕様
証明書ファイルまたはキー ファイルの最大サイズ	（ACE モジュール のみ）32 KB （ACE アプライアンス のみ）無制限（フラッシュ ディスクの容量は超えられません）
キー ペアの最大数	4096
同時 SSL 接続の最大数	（ACE モジュール のみ）250,000 （ACE アプライアンス のみ）100,000
秒あたりの SSL トランザクション（TPS）の最大数	（ACE モジュール のみ）1000 TPS（デフォルト）。30,000 TPS には、オプションのバンドル ライセンスが必要です。 （ACE アプライアンス のみ）100 TPS（デフォルト）。 オプションのバンドル ライセンスを使用して、TPC の最大数を増やすことができます。 ACE ライセンス オプションの詳細については、『*Administration Guide, Cisco ACE Application Control Engine*』のマニュアルを参照してください。
（ACE モジュール のみ）SSL 帯域幅の最大量	6 Gbps
（ACE アプライアンス のみ）SSL スループット	1000 Mbps
（ACE アプライアンス のみ）SSL バルク暗号化	1 Gbps

ACE SSL 機能

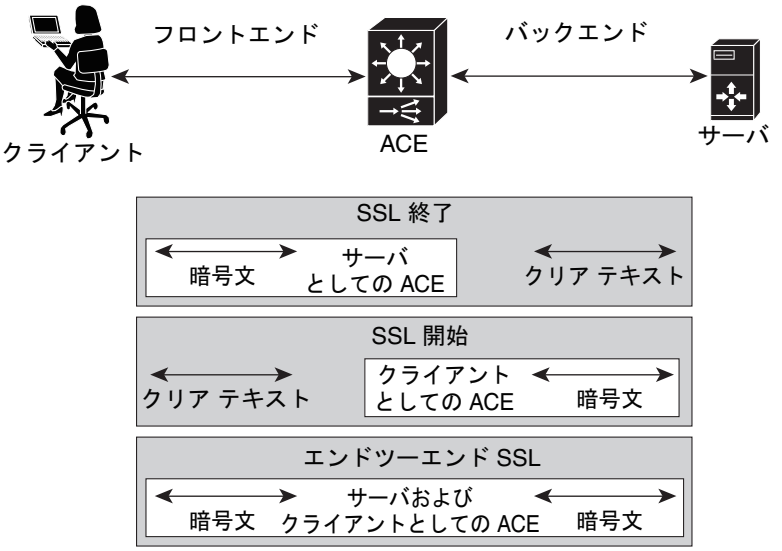
ACE は、クライアントとサーバ間のすべてのユーザ認証、公開 / 秘密キー生成、証明書管理、パケットの暗号化および復号化を行います。

複数のコンテキスト（仮想 ACE デバイス）に ACE を分割できます。証明書およびキー ファイルによって各コンテキストを設定します。コンテキストは SSL ピアとの SSL セッションを確立する必要があります。ACE は、作成される各コンテキストに関連付けられた証明書とキーを格納するために、フラッシュ メモリ内に安全なストレージ エリアを作成します。

ACE とその SSL ピアとの間の SSL セッションを確立して維持するために、ACE は、受信するトラフィックにポリシー マップを適用します。トラフィック特性が特定のポリシー マップの属性に一致する場合、ACE はポリシー マップに関連付けられたアクションを実行します。

ポリシー マップの定義方法に応じて、SSL セッション中にクライアントまたはサーバとして動作するように ACE を設定できます。図 1-2 では、ACE を使用した 3 種類の基本 SSL 設定を示します。ここでは、クライアントとサーバ間のデータの暗号化と復号化に ACE が使用されます。

図 1-2　　ACE SSL アプリケーション



次の項では、3 つの ACE SSL アプリケーションの概要を示します。

- SSL 終了
- SSL 開始
- エンドツーエンド SSL

SSL 終了

*SSL 終了*は、クライアントと通信する SSL サーバとして ACE が動作する、フロントエンド アプリケーションの ACE コンテキストの設定を示します。ACE とクライアント間のフローを定義するためにレイヤ 3 およびレイヤ 4 ポリシー マップを作成すると、ACE は、Web ブラウザ（クライアント）と HTTP 接続（サーバ）間のセキュリティ サービスを追加することにより、仮想 SSL サーバとして動作します。クライアントからのすべてのインバウンド SSL フローは ACE で終わります。

接続が終了したら、ACE はクライアントからの暗号文を復号化し、そのデータをクリア テキストとして HTTP サーバに送信します。SSL 終了のための ACE の設定の詳細については、第 3 章 「SSL 終了の設定」を参照してください。

SSL 開始

*SSL 開始*は、SSL サーバと通信するクライアントとして ACE が動作する、バックエンド アプリケーションの ACE コンテキストの設定を示します。ACE と SSL サーバ間のフローを定義するためにレイヤ 7 ポリシー マップを作成すると、ACE はクライアントとして機能し、ACE とサーバ間の SSL セッションを開始します。SSL 開始により、ACE はクライアントからクリア テキストを受信することが可能になり、その後、SSL サーバとの SSL セッションを確立し、SSL サーバ接続にクライアント接続を組み合わせることができます。ACE は、クライアントから受け取るクリア テキストを暗号化して、そのデータを暗号文として SSL サーバに送信します。SSL サーバは、SSL 終了（仮想 SSL サーバ）または実 SSL サーバ（Web サーバ）のいずれかに設定された ACE にすることができます。

SSL サーバからのアウトバウンド フローでは、ACE は、サーバからの暗号文を復号化して、クリア テキストをクライアントに返します。

SSL 開始のための ACE の設定の詳細については、第 4 章 「SSL 開始の設定」を参照してください。

エンドツーエンド SSL

エンドツーエンド SSL は、SSL 終了と SSL 開始の両方のための ACE コンテキストの設定を示します。クライアント、ACE、および SSL サーバ間でセキュアな SSL チャネルを確立することを必要とするアプリケーションがある場合、エンドツーエンド SSL 用に ACE を設定できます。たとえば、銀行間のトランザクションでは、クライアントとサーバ間で交換される財務情報を保護するためにエンドツーエンド SSL が必要です。

また、エンドツーエンド SSL では、ACE は、ロード バランシングおよびセキュリティ情報をデータに挿入することもできます。ACE は、受け取った暗号文を復号化して、ロード バランシングおよびファイアウォール情報をクリア テキストに挿入します。次に、ACE はこのデータを再び暗号化して、指定された宛先に暗号文を渡します。

エンドツーエンド SSL 開始のための ACE の設定の詳細については、第 5 章「エンドツーエンド SSL の設定」を参照してください。

ACE SSL 設定の前提条件

SSL オペレーション用に ACE を設定する前に、まずサーバ ロード バランシング（SLB）用に設定する必要があります。SLB の設定プロセス中に、次の設定オブジェクトを作成します。

- レイヤ 7 クラス マップ
- レイヤ 3 およびレイヤ 4 クラス マップ
- レイヤ 7 ポリシー マップ
- レイヤ 3 およびレイヤ 4 ポリシー マップ

SLB を設定してから、このマニュアルで説明した SSL 終了、SSL 開始、またはエンドツーエンド SSL 用の SSL 設定要件に合わせて、既存の SLB クラス マップとポリシー マップを変更します。

SLB 用に ACE を設定するには、『*Server Load-Balancing Guide, Cisco ACE Application Control Engine*』を参照してください。

大きい SSL レコードと小さい MSS での SSL の動作

SSL レコードのサイズが大きく、TCP 最大セグメント サイズ（MSS）が小さい組み合わせでは、TCP ウィンドウ サイズがゼロになることが原因で、ACE SSL スループットが非常に遅くなる可能性があります。この動作は、次の SSL 設定のいずれかで発生する可能性があります。

- SSL 終了
- SSL 開始
- エンドツーエンド SSL（SSL 終了と SSL 開始の組み合わせ）

この動作の理由は、データを暗号化または復号化する前に、ACE がバッファされた SSL レコードすべてを受信する必要があるためです。ローカル インターフェイス MSS が過度に低く設定されている場合、ACE がレコード全体を受信する前に、大きい SSL レコードがすべての利用可能なバッファを使用してしまい、TCP ウィンドウ サイズがゼロであることがアドバタイズされ、送信者は追加パケットの送信を停止します。**set tcp buffer-share** コマンドの値も過度に低く設定されていると、問題は悪化し、短時間でウィンドウ サイズがゼロになります。実際の決定要因は、パスの最大伝送単位（PMTU）、またはパケットのパス上の最少 MTU です。

SSL を ACE と使用する場合は、デフォルトのバッファ共有値（32768）より小さい値を設定しないようにします。デフォルト値では十分でない場合は、大きい SSL レコードや小さい MSS に合わせてより多くのバッファを使用できるようにするために、**set tcp buffer-share** の値を増やすことができます。

たとえば、160 MSS（200 MTU）の場合、最大サイズの SSL レコードを作成するために、103 バッファ（16,400 バイトの最大 SSL レコード サイズを 160 で除算）が必要になります。TCP に関する限り、103 バッファでは 57,736 バイトを保持できます。したがって、バッファ共有値は、SSL 接続を継続するために少なくとも 57736 である必要があります。異なる MSS と MTU サイズには、異なるバッファ共有の値を設定する必要があります。

set tcp buffer-share コマンドの詳細については、『*Security Guide, Cisco ACE Application Control Engine*』を参照してください。