SSL 開始の設定

この章では、クライアントからクリア テキストを受信した後、SSL サーバとの SSL 接続を開始するように CSS を設定するための手順を説明します。この章の主な内容は次のとおりです。

- SSL 開始の概要
- SSL 開始プロキシ リストの作成
- SSL 開始プロキシ リストへの説明の追加
- SSL 開始プロキシ リストでのバックエンド SSL サーバの定義
- SSL プロキシ リストのアクティブ化と使用中断
- SSL プロキシ リストの変更
- SSL 開始のサービスの設定
- SSL 開始のコンテンツ ルールの設定
- SSL 開始のトラブルシューティング

SSL 開始の概要

SSL モジュールを装着した CSS は、SSL 開始によって次の処理を実行できます。

- クライアントからのクリア テキストの受信
- コンテンツのロード バランシング
- クリア テキストの暗号化
- SSL サーバか、SSL 終了を設定した他の CSS への SSL 接続の開始（第 4 章「SSL 終了の設定」参照）

この機能は、サイト間でセキュアなデータ転送を行う手段として使用します。SSL 開始はサイト内でのクリア テキストの最大送信速度を実現し、同時にインターネット経由でのサイト間接続や SSL サーバとの接続で暗号化したテキスト送信の最大限のセキュリティを保証します。クリア テキスト接続からの SSL 接続を確立したい SSL サーバまたは CSS ごとに、それらにマップされる CSS で、SSL 開始サービスを設定する必要があります。このサービスは、SSL プロキシ リストを使用して、CSS 内のフローを正しく振り分けます。

図 6-1 に、SSL サーバへの SSL 開始フローを示します。

図 6-1　SSL サーバへの SSL 開始

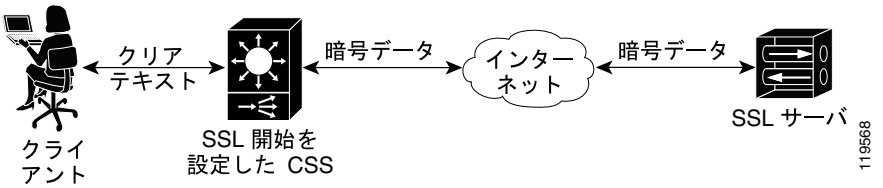
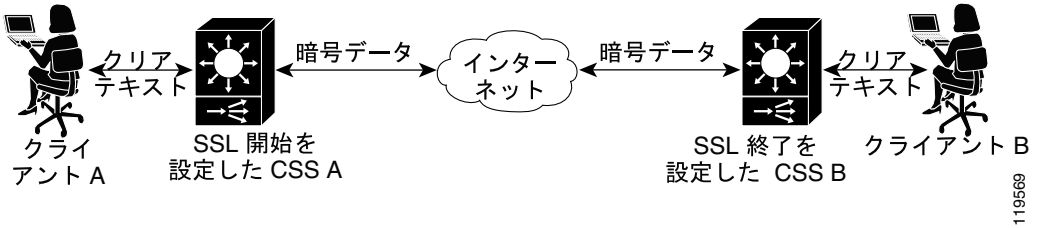


図 6-2 に、SSL 終了を設定した他の CSS との SSL 開始フローを示します。この場合、2 番目の CSS が仮想的なフロントエンド SSL サーバとして機能します。

図 6-2　SSL 終了を実行する 2 番目の CSS への SSL 開始



クライアント、SSL モジュール、およびサーバ間の SSL 情報の流れは、SSL プロキシ リストによって決定されます。SSL プロキシ リストは、インデックス エントリで識別される 1 つ以上のバックエンド SSL サーバ（CSS の SSL モジュールに作成する仮想サーバ）の定義で構成されており、CSS の SSL モジュールは、このバックエンド SSL サーバを使用して SSL サーバへの接続を開始します。1 つの SSL プロキシ リストには、最大で 256 のバックエンド SSL サーバを定義できます。

SSL プロキシ リストを作成してバックエンド SSL サーバを定義したら、リストをアクティブにする必要があります。続いて、そのプロキシ リストを開始サービスに追加すると、SSL 設定データの SSL モジュールへの転送が可能になります。開始サービスをアクティブ化すると、CSS が設定データを SSL モジュールに転送します。その後、各 SSL 開始サービスを SSL コンテンツ ルールに追加します。

SSL 開始プロキシ リストの作成

SSL 開始プロキシ リストは、SSL 開始サービスに関連付けられたバックエンド SSL サーバのグループです。SSL プロキシ リストの作成には、**ssl-proxy-list** コマンドを使用します。

ssl-proxy-list 設定モードには、ACL モード、ブート モード、グループ モード、rmon モード、および所有者設定モードを除く、ほとんどの設定モードからアクセスできます。また、このコマンドを ssl-proxy-list 設定モードから使用して、別の SSL プロキシ リストにアクセスすることもできます。SSL プロキシ リスト名は、1 ～ 31 文字のテキスト文字列を引用符で囲まずに入力します。

たとえば、SSL プロキシ リスト ssl_list1 を作成するには、次のコマンドを入力します。

```
(config)# ssl-proxy-list ssl_list1
Create ssl-list <ssl_list1>, [y/n]: y
```

SSL プロキシ リストを作成すると、CLI が ssl-proxy-list 設定モードに入ります。

```
(config-ssl-proxy-list[ssl_list1])#
```

既存の SSL プロキシ リストを削除するには、次のコマンドを入力します。

```
(config)# no ssl-proxy-list ssl_list1
Delete ssl-list <ssl_list1>, [y/n]: y
```

（注）　SSL サービスが使用中の場合、サービスに含まれるアクティブな SSL プロキシ リストは削除できません。この場合は、最初に SSL サービスを一時停止してから、特定の SSL プロキシ リストを削除する必要があります。

SSL 開始プロキシ リストへの説明の追加

SSL 開始プロキシ リストの説明を指定するには、**description** コマンドを使用します。説明は、スペースを含む 64 文字以内のテキスト文字列を引用符で囲んで入力します。

たとえば、ssl_list1 SSL プロキシ リストに説明を追加するには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# description “This is the SSL list
for www.brandnewproducts.com”
```

特定の SSL プロキシ リストの説明を削除するには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# no description
```

SSL 開始プロキシ リストでのバックエンド SSL サーバの定義

ここでは、SSL プロキシ リスト内にバックエンド SSL サーバの定義を 1 つ以上作成し、SSL 開始機能で使用できるようにする方法を説明します。SSL のプロキシ リスト モードで、**backend-server** コマンドを使用して CSS に SSL 開始バックエンド サーバを作成し、SSL プロキシ リスト内にインデックス エントリを作成します。このインデックスを使用して、バックエンド SSL サーバに関連する個々の SSL パラメータを設定します。CSS の SSL モジュールは、SSL プロキシ リストで定義されているバックエンド SSL サーバを使用して SSL サーバへの接続を開始します。SSL プロキシ リスト パラメータを設定する前に、バックエンド SSL サーバのインデックス番号を定義する必要があります。1 つの SSL プロキシ リストには、最大で 256 のバックエンド SSL 開始サーバを定義できます。

（注）　アクティブな SSL プロキシ リストには、変更を加えることはできません。変更を加える前に SSL プロキシ リストの使用を一時停止し、変更が終了したらリストを再度アクティブ化します。CCS は、そのプロキシ リストを使用して SSL サービスへ追加情報または変更内容を送信します。詳細については、「SSL プロキシ リストの変更」を参照してください。

SSL プロキシ リスト内にバックエンド サーバを定義した後、SSL サービスのアドレスに対応する IP アドレスと、SSL 開始サーバの IP アドレスに対応するサーバ IP アドレスを設定します。必要に応じて、他のオプションのプロキシ リスト パラメータも設定し、SSL プロキシ リストをアクティブ化します。バックエンド サーバを SSL 開始用に動作させるには、バックエンド サーバのタイプを **initiation** に設定する必要があります。

SSL プロキシ リストを設定してアクティブ化した後、リストを SSL 開始サービスに追加します。サービスをアクティブ化すると、CSS は設定データを SSL モジュールに転送するようになります。

以降では、次の項目について説明します。

- SSL 開始プロキシ リスト内でのバックエンド サーバのエントリの作成
- SSL 開始サーバとしてのバックエンド サーバの設定

- SSL 開始サーバの IP アドレスの設定
- SSL 開始サーバのポートの設定
- SSL サーバ IP アドレスの設定
- SSL サーバ ポートの設定
- SSL バージョンの設定
- 利用可能な暗号スイートの設定
- SSL セッション キャッシュ タイムアウトの設定
- SSL セッションのハンドシェイク再ネゴシエーションの設定
- TCP 仮想クライアント接続タイムアウト値の設定
- SSL モジュールでの TCP サーバ側接続タイムアウト値の設定
- SSL TCP 接続における確認応答遅延の変更
- クライアント側接続の Nagle アルゴリズムの指定
- SSL TCP 接続の TCP バッファリングの指定
- クライアント証明書とキーの設定
- サーバ認証用の CA 証明書の設定

SSL 開始プロキシ リスト内でのバックエンド サーバのエントリの作成

SSL 開始プロキシ リスト パラメータを設定するには、その前に SSL プロキシ リスト内にバックエンド サーバのエントリを作成する必要があります。SSL プロキシ リスト内にバックエンド サーバのエントリを作成するには、**backend-server** *number* コマンドを使用します。このコマンドを実行すると、デフォルトでは **backend-ssl** タイプのバックエンド サーバのエントリが SSL プロキシ リスト内に作成され、指定した番号（インデックス エントリ）が割り当てられます。バックエンド SSL サーバに関連付ける個々の SSL パラメータ（IP アドレス、証明書名、キー ペアなど）は、この番号を使って設定します。1 ～ 256 の数値を入力します。

たとえば、プロキシ リスト内にバックエンド サーバ 1 のエントリを作成するには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1
```

バックエンド サーバ 1 のエントリと、そのサーバに設定したすべてのパラメータをプロキシ リスト ssl_list1 から削除するには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# no backend-server 1
```

SSL 開始サーバとしてのバックエンド サーバの設定

バックエンド サーバを SSL 開始用に使用するには、**ssl-init** タイプのサービスに使用する開始サーバとして設定する必要があります。デフォルトでは、バックエンド サーバは、**ssl-accel-backend** タイプのサービスで使用する **backend-ssl** タイプのサーバとなります。

バックエンド サーバを **ssl-init** タイプのサービスに使用する開始サーバ（クライアントからクリア テキストのトラフィックを受け付け、SSL サーバとの間で SSL トラフィックを開始）として設定するには、**backend-server** *number* **type** コマンドを使用します。このコマンドのシンタックスは次のとおりです。

backend-server *number* **type** {**backend-ssl**|**initiation**}

たとえば、プロキシ リスト ssl_list1 内にバックエンド サーバ 1 を、SSL 開始サーバとして設定するには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 type initiation
```

すべてのバックエンド サーバ パラメータを設定せずに SSL 開始サーバをバックエンド SSL サーバとして設定し直すには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 type backend-ssl
```

SSL 開始サーバの IP アドレスの設定

SSL 開始サーバの IP アドレスを設定するには、**backend-server** *number* **ip address** コマンドを使用します。この IP アドレスは、SSL 開始サービスの IP アドレスに対応します。

たとえば、バックエンド SSL 開始サーバ 1 に IP アドレス 192.168.2.3 を設定するには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 ip address
192.168.2.3
```

SSL 開始サーバ 1 から IP アドレスを削除するには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# no backend-server 1 ip address
```

（注）　**active** コマンドを実行したときに SSL 開始バックエンド サーバの IP アドレスが設定されていないと、次のメッセージが表示され、リストはアクティブ化されません。

```
%% Error in backend-server 1: SSL-server/Backend-server must have valid
IP address
```

SSL 開始サーバのポートの設定

デフォルトでは、バックエンド SSL 開始サーバのポートはポート 80 です。このポートで SSL 開始サービスからのクリア テキスト データ トラフィックを受け付け、SSL モジュールに送信します。SSL 開始サーバに異なるポートを設定するには、**backend-server** *number* **port** コマンドを使用します。1 ～ 65535 のポート番号を入力します。

たとえば、ポート番号を 1200 に設定するには、次のように入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 port 1200
```

ポート番号をデフォルトの 80 に戻すには、次のように入力します。

```
(config-ssl-proxy-list[ssl_list1])# no backend-server 1 port
```

SSL サーバ IP アドレスの設定

このサーバの IP アドレスは、実際の SSL サーバに設定した IP アドレスと同じになります。実際の SSL サーバの IP アドレスを設定するには、**backend-server** *number* **server-ip** コマンドを使用します。

たとえば、サーバ IP アドレス 192.168.2.3 を設定するには、次のように入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 server-ip
192.168.2.3
```

プロキシ リストから実際のサーバの IP アドレスを削除するには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# no backend-server 1 server-ip
```

（注）　**active** コマンドを実行したときに実サーバの IP アドレスが設定されていないと、次のメッセージが表示され、リストはアクティブ化されません。

```
%% Error in backend-server 1: SSL-server/Backend-server must have valid
IP address
```

SSL サーバ ポートの設定

デフォルトでは、実際の SSL サーバのポート番号は 443 です。SSL サーバのサーバ ポートを変更するには、**backend-server** *number* **server-port** コマンドを使用します。1 ～ 65535 のポート番号を入力します。

たとえば、サーバ ポート番号を 155 に設定するには、次のように入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 server-port 155
```

ポート番号をデフォルトの 443 に戻すには、次のように入力します。

```
(config-ssl-proxy-list[ssl_list1])# no backend-server 1 server-port
```

SSL バージョンの設定

SSL モジュールは、実際の SSL サーバへの接続を開始します。サーバに送られる ClientHello メッセージ内のバージョンは、サポートする最新のバージョンです。

デフォルトでは、SSL バージョンは SSL バージョン 3 と TLS バージョン 1 です。SSL モジュールは、SSL バージョン 3 のヘッダーと、TLS バージョン 1 のメッセージの ClientHello を送信します。

バックエンド サーバがサポートする SSL のバージョンを指定するには、**backend-server** *number* **version** コマンドを使用します。

- **ssl3** ：SSL バージョン 3
- **tls1**：TLS バージョン 1
- **ssl-tls**：SSL バージョン 3 と TLS バージョン 1。SSL モジュールは、SSL バージョン 3 のヘッダー、TLS バージョン 1 のメッセージの ClientHello を送信します。

たとえば、SSL バージョン 3 を設定するには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 version ssl3
```

デフォルトの SSL バージョンに戻すには、次のように入力します。

```
(config-ssl-proxy-list[ssl_list1])# no backend-server 1 version
```

利用可能な暗号スイートの設定

バックエンド SSL 開始サーバが使用する暗号スイートを 1 つ以上設定するには、**backend-server** *number* **cipher** コマンドを使用します。デフォルトでは、サポートされた、ハードウェア アクセラレーションによる暗号スイートはすべて有効になっています。

SSL モジュールがサポートするすべての暗号スイートと、対応する暗号スイートの値のリストについては、第 4 章「SSL 終了の設定」の「暗号スイートの指定」の項にある表 4-1 を参照してください。これらの値は、SSL バージョン 3.0 および TLS バージョン 1.0 用に定義されている値と同じです。表 4-1 には、CSS ソフトウェアでエクスポートできる暗号スイートもリストされています。

デフォルトの設定を使用するか **all-cipher-suite** オプションを選択した場合、表 4-1 にリストした、rsa-with-rc4-128-md5 から始まる順序で暗号スイートが送信されます。

（注）　**all-cipher-suites** オプションでは、バックエンド サーバのすべての暗号スイートを再度有効にできます。このオプションが機能するのは、特定の暗号を設定していない場合だけです。再び **all-cipher-suites** オプションを使用するには、設定したすべての暗号を明示的に削除する必要があります。

たとえば、暗号 rsa-with-rc4-128-md5 を設定するには、次のように入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 cipher
rsa-with-rc4-128-md5
```

使用する暗号スイートをネゴシエートするとき、SSL モジュールはリスト中の最も重みの高い暗号から順にサーバに送信します。

デフォルトでは、設定されたすべての暗号スイートの重みは 1 ですが、暗号スイートに重みを割り当てることができます（最大の重みは 10）。

（注）　2 つ以上の暗号に同じ重みが設定されている場合（重みが設定されていない場合は、値が 1 となる）、これらの暗号は ClientHello に、実行設定ファイル内と同じ順序で示されます。

たとえば、重みを 10 に設定するには、次のように入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 cipher
rsa-with-rc4-128-md5 weight 10
```

バックエンド SSL 開始サーバの暗号スイートを 1 つ以上削除するには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# no backend-server 1 cipher
rsa-with-rc4-128-md5
```

SSL セッション キャッシュ タイムアウトの設定

SSL では、クライアントとサーバが完全なキー交換を行い、新しいマスター秘密キーが確立されるたびに、新しいセッション ID が作成されます。セッション キャッシュ タイムアウトを有効にすると、そのクライアントの以降の接続でそのマスター キーを再利用できます。キャッシュ タイムアウトを無効にすると、SSL モジュールへのそれぞれの新しい接続で完全な SSL ハンドシェイクを行う必要があります。**backend-server** *number* **session-cache** コマンドを使用して、以前に設定した秘密キーでバックエンド SSL サーバとの接続を再開できるように SSL モジュールを設定します。

デフォルトでは、300 秒（5 分）のタイムアウトでキャッシュ タイムアウトが有効になります。タイムアウト値は、0 ～ 72000（0 秒～ 20 時間）の範囲で設定できます。タイムアウト値 0 で、セッション キャッシュの再使用は無効になります。

たとえば、SSL セッションのキャッシュ タイムアウトに 500 秒を指定するには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 session-cache 500
```

セッション キャッシュ ID 再使用をデフォルトの 300 秒にリセットするには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# no backend-server 1 session-cache
```

セッション キャッシュ ID 再使用を無効にするには、0 秒のタイムアウト値を入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 session-cache 0
```

SSL セッションのハンドシェイク再ネゴシエーションの設定

SSL セッション ハンドシェイク コマンドを実行すると、SSL HelloRequest メッセージがクライアントへ送信され、SSL ハンドシェイク ネゴシエーションが再開されます。SSL 再ハンドシェイクは、長時間にわたって接続を維持しているときに、CSS とバックエンド SSL サーバ間の SSL セッションを再確立してセキュリティを保証する手段として役立ちます。

backend-server *number* **handshake data** *kbytes* コマンドを使用すると、CSS とバックエンド SSL サーバ間での一定量のデータの交換の後、SSL 再ハンドシェイクが強制的に行われます。この後、CSS は SSL ハンドシェイク メッセージを送信し、SSL セッションを再確立します。

デフォルトでは、データ交換後は、そのデータに基づくバックエンド SSL サーバの SSL 再ハンドシェイクが無効になります（0 に設定）。データは KB 単位で、0 ～ 512000 の範囲で設定します。

たとえば、SSL セッションの再ハンドシェイクまでのデータ量を 500 KB に設定するには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 handshake data
500
```

再ハンドシェイクのデータ量を 0 にリセットすると、データ交換に基づく再ハンドシェイクは無効になります。たとえば、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# no backend-server 1 handshake data
```

最大タイムアウト値を指定するには、**backend-server** *number* **handshake timeout** *seconds* コマンドを使用します。このタイムアウト値が経過すると、CSS は SSL ハンドシェイク メッセージを送信して SSL セッションを再確立します。タイムアウト値を設定すれば、指定した秒数の経過後に SSL セッションは新しいセッション キーを再取り決めするようになります。SSL 再ハンドシェイクに設定する値は、レイヤ 5 コンテンツ ルールで **advanced-balance ssl** ロード バランシング方式を使用する場合に、クライアントを接続先サーバに固定するために使用される SSL セッション ID が適切に調整されるかどうかの重要な要素になります。

デフォルトでは、バックエンド SSL サーバの SSL 再ハンドシェイクのタイムアウトは無効です（0 に設定）。タイムアウト値は、0 ～ 72000（0 秒～ 20 時間）の範囲で設定できます。

たとえば、SSL セッションの再ハンドシェイク タイムアウトを 30 秒に設定するには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 handshake timeout
30
```

タイムアウトを 0 に戻すには、次のように入力します。これにより、バックエンド サーバの再ハンドシェイク期間は無効になります。

```
(config-ssl-proxy-list[ssl_list1])# no backend-server 1 handshake
timeout
```

TCP 仮想クライアント接続タイムアウト値の設定

クライアントと SSL モジュール間の TCP 接続は、指定した時間が経過すると終了します。TCP タイムアウト機能を使用すると、SSL モジュールとクライアント間の TCP 接続を、より柔軟に管理できます。

クライアントと TCP 接続するためのパラメータを設定する方法については、次の項を参照してください。

- 仮想クライアント接続の TCP SYN タイムアウト値の指定
- 仮想クライアント接続の TCP 無活動タイムアウト値の指定
- クライアント側接続の Nagle アルゴリズムの指定

仮想クライアント接続の TCP SYN タイムアウト値の指定

CSS の SYN タイマーは、TCP 3 ウェイ ハンドシェイクを終了する手段として、CSS による SYN/ACK の送信とクライアントによる ACK の応答の間の時間差をカウントします。このタイムアウト値は、データの転送前に TCP 3 ウェイ ハンドシェイクが正常に完了しなかったクライアントとの TCP 接続を終了するために使用し、**ssl-server** *number* **tcp virtual syn-timeout** *seconds* コマンドで指定します。

TCP SYN のタイムアウト値（秒単位）には、0（TCP SYN タイムアウトは無効）～ 3600（1 時間）を入力します。デフォルトは 30 秒です。このコマンドに 0 を設定するとタイマーは非アクティブになり、切断された TCP 接続は再送信タイマーによってやがて終了します。

（注）　接続タイマーは、新しい TCP 接続についても、常に再送信終了時間より短く設定する必要があります。

TCP SYN タイムアウトの値を 100 秒に設定するには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 tcp virtual
syn-timeout 100
```

タイムアウトを無効にするには、値を 0 に設定します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 tcp virtual
syn-timeout 0
```

タイマーをデフォルトの 30 秒に戻すには、次のように入力します。

```
(config-ssl-proxy-list[ssl_list1])# no backend-server 1 tcp virtual
syn-timeout
```

仮想クライアント接続の TCP 無活動タイムアウト値の指定

TCP 無活動タイムアウトのカウントは、まず、TCP 3 ウェイ ハンドシェイクを終了するための、ACK をクライアントから受信したときに開始されます。この無活動タイマーは、そのトラフィック フローの SYN タイマーが停止した時点で再開されます。このタイムアウト値は、クライアントと SSL モジュール間の TCP 接続がほとんど、またはまったく活動していないときに、その接続を終了させるために使用し、**backend-server** *number* **tcp virtual inactivity-timeout** *seconds* コマンドを使用して指定します。

TCP 無活動タイムアウト値（秒単位）には、0（TCP 無活動タイムアウトは無効）～ 3600（1 時間）の値を入力します。デフォルトは 240 秒に設定されています。

このタイマー値は、再送のデフォルトのパラメータに基づき、60 秒（1 分）を超える値にする必要があります。

たとえば、仮想クライアント接続の TCP 無活動タイムアウトを 100 秒に設定するには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 tcp virtual
inactivity-timeout 100
```

タイムアウトを無効にするには、値を 0 に設定します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 tcp virtual
inactivity-timeout 0
```

タイマーをデフォルトの 240 秒に戻すには、次のように入力します。

```
(config-ssl-proxy-list[ssl_list1])# no backend-server 1 tcp virtual
inactivity-timeout
```

SSL TCP 接続における確認応答遅延の変更

クライアントまたはサーバ接続におけるデフォルトの確認応答遅延時間は 200 ミリ秒（Ms）です。次のコマンドを実行することで、確認応答遅延の SSL TCP タイマーの長さを無効にしたり調整したりできます。

backend-server *server-num* **tcp virtual**|**server ack-delay** *value*

value 変数は、確認応答遅延のタイマーの長さをミリ秒（Ms）で指定します。デフォルト値は 200 です。0 ～ 10000 の値を入力します。0 を指定すると、クライアントから SSL トラフィックを受信する際の確認応答遅延は無効になります。タイマーを無効にすると、SSL セッション キャッシュ（セッション ID の再使用）の使用によりセッションのパフォーマンスが向上します。

たとえば、クライアントと SSL モジュールとの間の TCP 接続に 400 ミリ秒の確認応答遅延を設定するには、次のように入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 20 tcp virtual
ack-delay 400
```

サーバと SSL モジュールとの間の TCP 接続に 400 ミリ秒の確認応答遅延を設定するには、次のように入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 20 tcp server
ack-delay 400
```

クライアント側接続の Nagle アルゴリズムの指定

TCP Nagle アルゴリズムでは、クライアントと SSL モジュールの間の TCP 接続で転送されるサイズの小さい多数のバッファ メッセージを自動的に連結します。この処理で個々の TCP 接続で送信されるパケット数が減少し、CSS のスループットが向上します。ただし、Nagle アルゴリズムと TCP 遅延応答確認の相互作用によっては、TCP 接続の遅延時間が長くなることもあります。TCP 接続（クリア テキストまたは SSL）に許容範囲を超える遅延が発生するようであれば、Nagle アルゴリズムを無効にしてください。

クライアントと SSL モジュールの間の TCP 接続の Nagle アルゴリズムを無効にする、または再度有効にするには、**backend-server** *number* **tcp virtual nagle** コマンドを使用します。このコマンドのシンタックスは次のとおりです。

backend-server *number* **tcp virtual nagle enable|disable**

クライアントと SSL モジュールの間の TCP 接続の Nagle アルゴリズムを無効にするには、次のように入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 tcp virtual nagle
disable
```

クライアントと SSL モジュールの間の TCP 接続の Nagle アルゴリズムを再度有効にするには、次のように入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 tcp virtual nagle
enable
```

SSL モジュールでの TCP サーバ側接続タイムアウト値の設定

SSL モジュールとサーバ間の TCP 接続は、指定した時間が経過すると終了します。この TCP タイムアウト機能を使用すると、CSS SSL モジュールとサーバ間の TCP 接続を、より柔軟に制御できます。

サーバとの TCP 接続のタイムアウト値を設定する方法については、次の項を参照してください。

- サーバ側接続の TCP SYN タイムアウト値の指定
- サーバ側接続の TCP 無活動タイムアウト値の指定
- サーバ側接続の Nagle アルゴリズムの指定

サーバ側接続の TCP SYN タイムアウト値の指定

TCP SYN タイマーは、CSS が SYN を送信してバックエンドの TCP 接続を開始したタイミングと、サーバが SYN/ACK で応答したタイミングの時間差をカウントします。サーバとの TCP 接続で TCP 3 ウェイ ハンドシェイクが正常に完了しなかったときに、その接続をデータ転送前に終了させるために使用するこのタイムアウト値は、**backend-server** *number* **tcp server syn-timeout** *seconds* コマンドを使用して指定します。

TCP SYN のタイムアウト値（秒単位）には、0（TCP SYN タイムアウトは無効）～ 3600（1 時間）の値を入力します。デフォルトは 30 秒です。このコマンドに 0 を設定するとタイマーは非アクティブになり、切断された TCP 接続は再送信タイマーによってやがて終了します。

（注）　接続タイマーは、新しい TCP 接続についても、常に再送信終了時間より短く設定する必要があります。

たとえば、サーバ側接続の TCP SYN タイムアウトを 100 秒に設定するには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 tcp server
syn-timeout 100
```

タイムアウトを無効にするには、値を 0 に設定します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 tcp server
syn-timeout 0
```

タイマーをデフォルトの 30 秒に戻すには、次のように入力します。

```
(config-ssl-proxy-list[ssl_list1])# no backend-server 1 tcp server
syn-timeout
```

サーバ側接続の TCP 無活動タイムアウト値の指定

TCP 無活動タイムアウトのカウントは、CSS がサーバから SYN/ACK を受信した時点から開始されます。この無活動タイマーは、そのトラフィック フローの SYN タイマーが停止した時点で再開されます。サーバとの TCP 接続がほとんど、またはまったく活動していないときに、その接続を終了させるために使用されるこのタイムアウト値は、**backend-server** *number* **tcp server inactivity-timeout** *seconds* コマンドを使用して指定します。

TCP 無活動タイムアウト値（秒単位）には、0（TCP 無活動タイムアウトは無効）～ 3600（1 時間）の値を入力します。デフォルトは 240 秒です。

たとえば、サーバ側接続の TCP 無活動タイムアウトを 100 秒に設定するには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 tcp server
inactivity-timeout 100
```

タイムアウトを無効にするには、値を 0 に設定します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 tcp server
inactivity-timeout 0
```

タイマーをデフォルトの 240 秒に戻すには、次のように入力します。

```
(config-ssl-proxy-list[ssl_list1])# no backend-server 1 tcp server
inactivity-timeout
```

サーバ側接続の Nagle アルゴリズムの指定

TCP Nagle アルゴリズムは、バックエンド サーバと SSL モジュールの間の TCP 接続で転送されるサイズの小さい多数のバッファ メッセージを自動的に連結します。この処理で個々の TCP 接続で送信されるパケット数が減少し、CSS のスループットが向上します。ただし、Nagle アルゴリズムと TCP 遅延応答確認の相互作用によって、TCP 接続の遅延時間が長くなることもあります。TCP 接続（クリア テキストまたは SSL）に許容範囲を超える遅延が発生するようであれば、Nagle アルゴリズムを無効にしてください。

サーバと SSL モジュールの間の TCP 接続の Nagle アルゴリズムを無効、または再度有効にするには、**backend-server** *number* **tcp server nagle** コマンドを使用します。このコマンドのシンタックスは次のとおりです。

backend-server *number* **tcp server nagle enable|disable**

サーバと SSL モジュールの間の TCP 接続の Nagle アルゴリズムを無効にするには、次のように入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 tcp server nagle
disable
```

サーバと SSL モジュールの間の TCP 接続の Nagle アルゴリズムを再度有効にするには、次のように入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 tcp server nagle
enable
```

SSL TCP 接続の TCP バッファリングの指定

輻輳によりネットワークに許容範囲を超える遅延が発生する場合、特定の TCP 接続に対して、TCP ウィンドウがシャットダウンされて 0 に設定されるまでに CSS によってバッファリングされるデータ量（バッファ サイズ）を増やすことができます。特定の TCP 接続でバッファリングされるクライアントまたはサーバからのデータ量を設定するには、**backend-server** *number* **tcp buffer-share** コマンドを使用します。このコマンドのシンタックスは次のとおりです。

backend-server *number* **tcp buffer-share rx** *number1*|**tx** *number2*

特定の接続でバッファリングできるクライアント トラフィックからのデータ量（バイト数）を設定するには、**rx** *number1* キーワードと変数を使用します。デフォルトのバッファ サイズは 32768 です。バッファ サイズには 16400 ～ 262144 の値を指定できます。たとえば、値を 65535 に設定するには、次のように入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 tcp buffer-share
rx 65536
```

バッファ サイズをデフォルトの 32768 にリセットするには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# no backend-server 1 tcp
buffer-share rx
```

特定の接続でバッファリングできるサーバからクライアントへのデータ量（バイト数）を設定するには、**tx** *number2* キーワードと変数を使用します。デフォルトのバッファ サイズは 65536 です。バッファ サイズには 16400 ～ 262144 の値を指定できます。たとえば、値を 131072 に設定するには、次のように入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 tcp buffer-share
tx 131072
```

バッファ サイズをデフォルトの 65536 にリセットするには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# no backend-server 1 tcp
buffer-share tx
```

クライアント証明書とキーの設定

SSL サーバでは、多くの場合、データ転送前にクライアント認証が必要です。このようなサーバに対してクライアント（この場合は SSL モジュール）が自身の認証を設定するには、CSS にクライアント証明書とキーを設定する必要があります。

クライアント証明書とキー ペアを取得するには、認証局（CA）に連絡してください。CA でクライアント証明書とキー ペアの用意ができたら、これらを CSS にインポートする必要があります。証明書とキー ペアのインポートについては、第 3 章「SSL 証明書とキーの設定」の「証明書と秘密キーのインポートまたはエクスポート」の項を参照してください。証明書とキー ペアをインポートしたら、これらをファイル名と関連付ける必要があります。証明書とキー ペアをファイル名に関連付ける方法については、第 3 章「SSL 証明書とキーの設定」の「証明書ファイルおよび秘密キー ファイルと名前との関連付け」の項を参照してください。

CSS にクライアント証明書とキーが設定されていない場合に、クライアント証明書を必要とする SSL サーバへの接続を SSL モジュールが開始すると、CSS では、Requested Client Certificate Not Sent カウンタの値が増加します。

（注）　必要なクライアント証明書が SSL サーバに渡されない場合は、接続が終了することがあります。

ここでは、クライアント証明書とキーの設定方法について説明します。

RSA 証明書名の設定

バックエンド サーバの RSA 証明書を設定するには、**backend-server** *number* **rsacert** *name* コマンドを使用します。証明書は、あらかじめ SCM にロードしておく必要があります。指定した証明書名が存在しない場合、CSS のログにエラー メッセージが出力されます。RSA 証明書の名前は、1 ～ 31 文字のテキスト文字列を引用符で囲まずに入力します。

たとえば、RSA 証明書名を myrsacert に設定するには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 rsacert myrsacert
```

RSA 証明書を SSL プロキシ リストから削除するには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# no backend-server 1 rsacert
```

RSA キー名の設定

バックエンド サーバの RSA キー名を設定するには、**backend-server** *number* **rsakey** *name* コマンドを使用します。キー ペアは、あらかじめ SCM にロードしておく必要があります。指定したキー ペアが存在しない場合、CSS のログにエラー メッセージが出力されます。RSA キーペアの名前は、1 ～ 31 文字のテキスト文字列を引用符で囲まずに入力します。

たとえば、RSA キー ペア名を myrsakey に設定するには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 rsakey myrsakey
```

RSA キー ペアを SSL プロキシ リストから削除するには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# no backend-server 1 rsakey
```

Diffie Hellman パラメータの設定

バックエンド サーバの Diffie-Hellman（DH）パラメータ ファイルを設定するには、**backend-server** *number* **dhparam** *name* コマンドを使用します。DH パラメータ ファイルは、あらかじめ SCM にロードしておく必要があります。パラメータ ファイルが存在しない場合、CSS のログにエラー メッセージが出力されます。DH パラメータ ファイルの名前は、1 ～ 31 文字のテキスト文字列を引用符で囲まずに入力します。

たとえば、DH パラメータ ファイル名を dhparamfile2 に設定するには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 dhparam
dhparamfile2
```

設定した DH パラメータ ファイルを SSL プロキシ リストから削除するには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# no backend-server 1 dhparam
```

DSA 証明書名の設定

バックエンド サーバの DSA 証明書を設定するには、**backend-server** *number* **dsacert** *name* コマンドを使用します。証明書は、あらかじめ SCM にロードしておく必要があります。指定した証明書名が存在しない場合、CSS のログにエラー メッセージが出力されます。DSA 証明書の名前は、1 ～ 31 文字のテキスト文字列を引用符で囲まずに入力します。

たとえば、DSA 証明書名を mydsacert に設定するには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 dsacert mydsacert
```

DSA 証明書を SSL プロキシ リストから削除するには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# no backend-server 1 dsacert
```

DSA キー ファイル名の設定

バックエンド サーバの DSA キー名を設定するには、**backend-server** *number* **dsakey** *name* コマンドを使用します。キー ペアは、あらかじめ SCM にロードしておく必要があります。指定したキー ペアが存在しない場合、CSS のログにエラー メッセージが出力されます。DSA キーペアの名前は、1 ～ 31 文字のテキスト文字列を引用符で囲まずに入力します。

たとえば、DSA キー ペア名を mydsakey に設定するには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 dsakey mydsakey
```

DSA キー ペアを SSL プロキシ リストから削除するには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# no backend-server 1 dsakey
```

サーバ認証用の CA 証明書の設定

CSS に特定の CA の公開キーがある場合は、サーバ証明書がその CA によって署名されたものかどうかがわかります。CSS は、CA 証明書からその CA の公開キーを取得します。SSL 開始プロキシ リスト内に CA 証明書名が設定されている場合、CSS はその証明書に含まれている公開キーを使用して、そのサーバ証明書内の CA のデジタル署名を検証できます。SSL 開始プロキシ リスト内で CA 証明書を定義すると、CSS 側ではサーバ証明書の検証が必要なものと認識します。

（注）　デフォルトでは、SSL サーバに認証は行われません。

SSL 開始プロキシ リストで CA 証明書を設定する前に、CSS に証明書をインポートして、その証明書をファイル名に関連付ける必要があります。CA 証明書のインポートについては、第 3 章「SSL 証明書とキーの設定」の「証明書と秘密キーのインポートまたはエクスポート」の項を参照してください。証明書とファイル名の関連付けについては、第 3 章「SSL 証明書とキーの設定」の「証明書ファイルおよび秘密キー ファイルと名前との関連付け」の項を参照してください。

SSL モジュール（クライアント）での SSL サーバ認証を有効にするには、1 ～ 4 個の CA 証明書を SSL 開始プロキシ リストに設定する必要があります。5 個以上の CA 証明書を設定しようとすると、次のエラー メッセージが表示されます。

```
%% Max number of CA Certificates configured on server.
```

プロキシ リストに CA 証明書を設定するには、**cacert** コマンドを使用します。このコマンドのシンタックスは次のとおりです。

backend-server *number* **cacert** {*name*}

name 変数には、あらかじめ CA 証明書に関連付けておいたファイル名を指定します。1 ～ 31 文字の名前を入力します。CA 証明書は、あらかじめ SCM にロードしておく必要があります。各 SSL 開始プロキシ リストには、最大で 4 個の CA 証明書を定義できます。CSS では、CA 証明書は設定した順に、サーバ証明書の検証に使用されます。

たとえば、SSL 開始サーバ 1 の ssl_list1 プロキシ リストに CA 証明書 mycert1 を設定するには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# backend-server 1 cacert mycert1
```

SSL プロキシ リストから CA 証明書を削除するには、このコマンドの先頭に **no** を入力します。たとえば、バックエンド SSL 開始サーバ 1 の ssl_list1 プロキシ リストから CA 証明書 mycert1 を削除するには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# no backend-server 1 cacert mycert1
```

SSL プロキシ リストのアクティブ化と使用中断

SSL プロキシ リストをアクティブにするには、バックエンド SSL サーバを 1 つ以上、**initiation** タイプとしてリスト内に定義する必要があります。「SSL 開始プロキシ リストでのバックエンド SSL サーバの定義」を参照してください。

CSS は SSL プロキシ リストをチェックして、必要なコンポーネントがすべて設定されているか確認します。このときに、証明書とキー ペアの相互確認も行います。確認に失敗した場合、証明書名は受け入れられず、エラー メッセージ「`Certificate and key pair do not match`」がログに記録され、SSL プロキシ リストはアクティブ化されません。設定されているキー ペアを削除するか、有効な証明書を設定する必要があります。

新しいまたは変更された SSL プロキシ リストをアクティブ化するには、**active** コマンドを使用します。次に例を示します。

```
(config-ssl-proxy-list[ssl_list1])# active
```

SSL プロキシ リストは、アクティブ化するとサービスに追加できます。「SSL 開始のサービスの設定」を参照してください。

プロキシ リスト内で定義しているバックエンド SSL サーバを表示するには、**show ssl-proxy-list** コマンドを使用します（第 7 章「SSL の設定情報および統計情報の表示」参照）。

アクティブな SSL プロキシ リストの使用を一時停止するには、**suspend** コマンドを使用します。

SSL プロキシ リストを一時停止するには、次のコマンドを入力します。

```
(config-ssl-proxy-list[ssl_list1])# suspend
```

SSL プロキシ リストの変更

SSL プロキシ リストは、リストがアクティブになっているときには変更できません。変更を加える前に SSL プロキシ リストの使用を一時停止し、変更が終了したらリストを再度アクティブ化します。

プロキシ リストを変更すると、SSL サービスを一時停止したり、そのリストを使用して再度アクティブ化したりする必要はありません。SSL モジュールにより、次のことが実行されます。

- プロキシ リストを使用して SSL サービスへ追加情報または変更内容を送信する。
- 変更または削除された SSL 関連サービスの接続を解除する。SSL モジュールがこれらの接続に関するパケットを受信すると、SSL モジュールは TCP RST を送信します。

注意　フロントエンド サーバとバックエンド サーバをフローに使用している場合は、エンドツーエンド接続を動作させるために両方のサーバがアクティブである必要があります。SSL プロキシ リストを変更する場合、サービスがまだアクティブなうちはリストからバックエンド サーバを削除しないでください。SSL プロキシ リストを再びアクティブ化する際、エンドツーエンド接続が失敗します。

SSL 開始のサービスの設定

SSL プロキシ リストは複数の SSL サービスに属すことができます（サービスにつき 1 つの SSL プロキシ リストを指定）。また、1 つの SSL サービスが複数のコンテンツ ルールに属すことも可能です。サービスをコンテンツ ルールに適用すると、クリア テキストのコンテンツ要求を暗号化する SSL モジュールに転送できます。

このタイプのサービスを SSL 開始コンテンツ ルールに追加するには、次の作業を行う必要があります。

- サービスに IP アドレスを設定する。
- SSL 開始サービスのキープアライブ タイプ（なし、ICMP、TCP、ネームド、スクリプト化または SSL）を設定する。暗号化された固定または非固定の HTTP キープアライブを設定することで、サーバから返されたフル SSL ハンドシェイクとデータを検証することもできます。

  TCP または SSL キープアライブ タイプを設定した場合、そのサービスを適切に動作させるためにキープアライブ ポートを適切に設定する必要があります。
- SSL プロキシ リストに **ssl-init** サービス タイプの SSL 開始バックエンド サーバを設定する。

（注）　CSS は装着された SSL モジュールにつき、複数のアクティブな **ssl-init** タイプの SSL サービスをサポートします。

ここでは、次の内容について説明します。

- SSL サービスのアクティブ化
- SSL サービスの一時停止

（注）　サービス ポートを設定しないと、コンテンツ ルールと同じポート番号が使用されます。

SSL サービスの作成

SSL モジュールで使用するサービスを作成する際には、そのサービスを CSS の SSL サービスとして指定し、CSS にそれを認識させる必要があります。SSL 開始コンテンツ ルールで使用する複数の SSL サービスを作成できます。サービスの作成についての詳細は、『*Cisco Content Services Switch Content Load-Balancing Configuration Guide*』を参照してください。

SSL サービス名は 1 ～ 31 文字で入力します。

サービス *ssl_serv1* を作成するには、次のコマンドを入力します。

```
(config)# service ssl_serv1
Create service <ssl_serv1>, [y/n]: y
```

CSS がサービス モードに変わります。

```
(config-service[ssl_serv1])#
```

SSL サービス タイプの設定

SSL 開始サービスには、**ssl-init** サービス タイプを設定する必要があります。SSL 開始サービスのサービス タイプを設定するには、次のコマンドを実行します。

```
(config-service[server1]# type ssl-init
```

SSL 開始サービスの IP アドレスの設定

SSL 開始サービスの IP アドレスは、SSL プロキシ リストでバックエンド SSL 開始サーバ用に設定した IP アドレス（**backend server-ip** アドレス）と一致させる必要があります。

たとえば、SSL 開始サーバに IP アドレス 192.168.21.7 を設定するには、次のコマンドを入力します。

```
(config-service[server1])# ip address 192.168.21.7
```

SSL 開始サービスの IP アドレスを削除するには、次のコマンドを入力します。

```
(config-service[server1])# no ip address
```

SSL 開始サービスへの SSL プロキシ リストの追加

SSL 開始サーバ用に SSL プロキシ リストを設定した後、アクティブなリストを 1 つ以上の SSL 開始サービスに追加して、バックエンド SSL 開始サーバからの SSL コンテンツ要求の処理方法を定義します。SSL 開始サービスの設定方法は、サービス タイプを **ssl-init** に設定する点を除いて、ローカル サービスの設定方法と同じです。また、SSL 開始サービスでは SSL プロキシ リストにバックエンド サーバを定義する必要があります。

SSL プロキシ リストには、SSL 開始サービスのパラメータがあります。プロキシ リストをサービスに追加するには、**add ssl-proxy-list** コマンドを使用します。SSL プロキシ リストに SSL 開始を設定する方法については、「SSL 開始プロキシ リストの作成」を参照してください。

サービスに追加する作成済みの SSL プロキシ リストの名前（「SSL 開始プロキシ リストの作成」参照）を入力します。

たとえば、SSL 開始サービスを設定した SSL プロキシ リスト *ssl_list1* を追加するには、次のコマンドを入力します。

```
(config-service[ssl_serv1])# add ssl-proxy-list ssl_list1
```

SSL 開始サーバの SSL プロキシ リストを削除するには、次のコマンドを入力します。

```
(config-service[ssl_serv1])# remove ssl-proxy-list ssl_list1
```

SSL モジュール スロットの指定

CSS 11501 では統合型の SSL モジュールを 1 つサポートします。CSS 11503 と CSS 11506 では、複数の SSL モジュール（CSS 11503 では最大 2 つ、CSS 11506 では最大 4 つ）をサポートします。SSL サービスでは、SSL プロキシ リストと仮想 SSL サーバを特定の SSL モジュールに関連付けるために、SSL モジュール スロット番号が必要です。SSL モジュールが装着されている CSS シャーシのスロットを指定するには、**slot** コマンドを使用します。

有効なスロット番号は次のとおりです。

- CSS 11501：2
- CSS 11503：2 と 3
- CSS 11506：2 ～ 6

スロット 1 は、SCM 用に予約されています。

（注）　CSS は装着された SSL モジュールにつき、複数のアクティブな **ssl-init** タイプの SSL サービスをサポートします。

たとえば、CSS シャーシのスロット 3 にある SSL モジュールを指定するには、次のコマンドを入力します。

```
(config-service[ssl_serv1])# slot 3
```

SSL 開始サービスのキープアライブ タイプの設定

ssl-init タイプのサービスでは、キープアライブを使用して、SSL サーバの状態を定期的にチェックできます。このサービスに設定された IP アドレスに、CSS からキープアライブが送信されます。

キープアライブを設定するには、サービス設定モードで **keepalive type** コマンドを使用します。このサービス設定モードのコマンドのシンタックスは次のとおりです。

(config-service[server1])# **keepalive type** *type*

type 変数には、次のキープアライブ タイプのいずれかを入力します。

- **icmp**：ICMP エコー メッセージ（ping）。これがデフォルトのキープアライブ タイプです。
- **none**：キープアライブ メッセージをこのサービスに送信しない。
- **ssl**：このサービスの SSL HELLO キープアライブ。CSS は、クライアント HELLO を送信して SSL サーバに接続します。サーバから HELLO を受信すると、CSS は TCP RST を送信して接続を閉じます。
- **tcp**：3 ウェイ ハンドシェイクとリセット（SYN、SYN-ACK、ACK、RST-ACK）を通じてサービスの利用可能状況を判定する TCP セッション
- **http** {**non-persistent**} **encrypt**：暗号化された固定または非固定の HTTP キープアライブ。サーバから返されたフル SSL ハンドシェイクおよびデータを検証します。

（注）　TCP、SSL、または暗号化キープアライブを設定した場合、そのサービスを適切に動作させるためにキープアライブ ポートを適切に設定する必要があります。

ICMP、SSL、TCP、および他の CSS のキープアライブの詳細については、『*Cisco Content Services Switch Content Load-Balancing Configuration Guide*』を参照してください。暗号化 HTTP キープアライブの詳細については、次の項を参照してください。

暗号化 HTTP キープアライブの設定

暗号化 HTTP キープアライブにより、サーバから返されたフル SSL ハンドシェイクとデータを検証できます。このキープアライブは、SSL モジュールに対して HTTP GET または HEAD を実行します。その後で、SSL モジュールが設定済みサーバへ接続します。

SSL 開始サービスの場合、暗号化キープアライブは、サービス用に設定されたスロット内の SSL モジュールを使用します。SSL モジュール スロットの設定については、「SSL モジュール スロットの指定」を参照してください。

設定されたキープアライブの IP アドレスとポートは、SSL プロキシ リスト内の SSL 開始サーバと同じである必要があります。

（注）　SSL プロキシ リストには、最大 256 の SSL バックエンド サーバまたは開始サーバが含まれます。したがって、CSS 上の暗号化キープアライブの総数は 256 までです。

（注）　暗号化 HTTP でキープアライブを設定する場合、設定済みのバンクエンド サーバ IP アドレスと実サーバ IP アドレスが同じであることを確認してください。

サービスまたはグローバル暗号化キープアライブを設定できます。このサービス設定モードのコマンドのシンタックスは次のとおりです。

(config-service)# **keepalive type http** {**non-persistent**} **encrypt**

たとえば、SSL 開始サービスに対し、暗号化 HTTP HEAD 非固定キープアライブをサービスとして設定するには、次のように入力します。

```
(config-service [ssl_serv1])# keepalive http non-persistent encrypt
```

グローバル暗号化キープアライブのシンタックスは次のとおりです。

(config-keepalive)# **type http** {**non-persistent**} **encrypt**

グローバル キープアライブの場合、キープアライブに対するサービスやスロットの情報はありません。グローバル暗号化キープアライブでの SSL モジュールの選択は、SSL 開始サービスのスロットを使用します。

（注）　暗号化グローバル キープアライブを正常に動作させるには、同じ SSL プロキシ リストを使用するサービスに追加します。

たとえば、SSL 開始サービスに対し、暗号化 HTTP HEAD 非固定キープアライブをサービスとして設定するには、次のように入力します。

```
(config-keepalive [SSL1])# http non-persistent encrypt
```

その後で、SSL 開始サービスのサービスにグローバル キープアライブを割り当てます。

セッション ID の再使用をサポートするように SSL 開始サーバが設定されていて、キープアライブを使用してフル SSL ハンドシェイクを実行する場合には、別の SSL 開始サーバを設定する必要があります。この 2 台めのサーバは、オリジナルの SSL サーバと同じ設定にする必要があります。ただし、ポート番号とセッション キャッシュ タイムアウトの値は別の値にします。次に、2 台めのサーバのポート番号で暗号化キープアライブを設定します。暗号化キープアライブは、SSL モジュールを介してこのサーバへ送信されます。このモジュールが SSL バックエンド サーバへ接続すると、キープアライブがフル SSL ハンドシェイクを実行します。

暗号化キープアライブの設定例

次に示す設定は、SSL 開始サーバに対する暗号化 HTTP 非固定キープアライブの例です。キープアライブが宛先アドレス 192.168.7.10 とポート 80 で SSL モジュール 3 へ送信されます。

```
!********************** SSL PROXY LIST **********************
ssl-proxy-list SSLInit_list
  backend-server 1
  backend-server 1 ip address 192.168.7.10
  backend-server 1 server-ip 192.168.7.10
  backend-server 1 type initiation
  active

!************************* SERVICE **************************

service DC-SSL1
  ip address 192.168.7.10
  port 80
  type ssl-init
  slot 2
  keepalive type http non-persistent encrypt
  keepalive method head
  keepalive uri "/index.html"
  active
```

次に示す設定は、SSL 開始サーバに対するグローバル暗号化 HTTP 非固定キープアライブの例です。グローバル キープアライブが宛先アドレス 192.168.7.10 とポート 80 で SSL モジュールの 2 または 3 のいずれかに送信されます。サーバがダウンした場合、このキープアライブに関連付けられたすべてのサーバもダウンします。

```
!************************** KEEPALIVE ************************
keepalive SSL1
  IP address 192.168.7.10
  type http non-persistent encrypt
  method head
  uri "/index.html"
  active

!********************** SSL PROXY LIST ***********************
ssl-proxy-list SSLInit_list
  backend-server 1
  backend-server 1 ip address 192.168.7.10
  backend-server 1 server-ip 192.168.7.10
  backend-server 1 type initiation
  backend-server 2
  backend-server 2 ip address 192.168.7.20
  backend-server 2 server-ip 192.168.7.20
  backend-server 2 type initiation
  backend-server 3
  backend-server 3 ip address 192.168.7.30
  backend-server 3 server-ip 192.168.7.30
  backend-server 3 type initiation
  backend-server 4
  backend-server 4 ip address 192.168.7.40
  backend-server 4 server-ip 192.168.7.40
  backend-server 4 type initiation
  active
```

```
!************************* SERVICE ************************

service DC1
  type ssl-init
  ip address 192.168.7.10
  protocol tcp
  port 80
  slot 2
  keepalive named SSL1
active

service DC2
  type ssl-init
  ip address 192.168.7.20
  protocol tcp
  port 80
  slot 3
  keepalive named SSL1
  active

service DC3
  type ssl-init
  ip address 192.168.7.30
  protocol tcp
  port 80
  slot 2
  keepalive named SSL1
  active

service DC4
  type ssl-init
  ip address 192.168.7.40
  protocol tcp
  port 80
  slot 3
  keepalive named SSL1
  active
```

SSL セッションの ID キャッシュ サイズ

キャッシュ サイズは、1 つの SSL モジュールの専用のセッション キャッシュに格納できる SSL セッション ID の最大数で表します。**ssl-init** タイプのサービスについては、SSL セッション キャッシュ サイズが 4096 エントリに固定されており、設定できません。

SSL サービスのアクティブ化

SSL プロキシ リスト サービスを設定したら、**active** コマンドを使用してそのサービスをアクティブ化します。サービスをアクティブ化すると、そのサービスは、クライアントとサーバ間で行われる SSL コンテンツ要求のロード バランシングのためのリソース プールに格納されます。

SSL サービスをアクティブ化する前に、次の作業を行います。

- SSL 開始サーバで、サービスをアクティブ化する前に SSL プロキシ リストを **ssl-init** タイプのサービスに追加します。**active** コマンドを入力したときにリストが設定されていないと、次のエラー メッセージがログに記録され、サービスはアクティブ化されません。

  ```
  Must add at least one ssl-proxy-list to an ssl-init type service
  ```

- サービスをアクティブ化する前に、サービスに追加した SSL プロキシ リストをアクティブ化します。リストが一時停止していると、次のエラー メッセージがログに記録され、サービスはアクティブ化されません。

  ```
  No ssl-lists on service, service not activated
  ```

サービスをアクティブ化する準備ができると、各 SSL プロキシ リストの適切な SSL 設定データが特定の SSL モジュールへ転送され、サービスがアクティブ化されます。転送時にエラーが発生すると、対応するエラーがログに記録され、サービスはアクティブ化されません。

サービス *ssl_serv1* をアクティブ化するには、次のコマンドを入力します。

```
(config-service[ssl_serv1])# active
```

SSL サービスの一時停止

SSL サービスを一時停止し、SSL コンテンツ要求のロード バランシングを行うためのリソース プールからそのサービスを削除する場合には、**suspend** コマンドを使用します。SSL サービスを一時停止しても既存のコンテンツ フローには影響ありませんが、新たな接続を確立してサービスのコンテンツにアクセスすることはできなくなります。

サービス *ssl_serv1* を一時停止するには、次のコマンドを入力します。

```
(config-service[ssl_serv1])# suspend
```

SSL 開始のコンテンツ ルールの設定

CSS でクライアントのクリア テキストのコンテンツ要求を暗号化するには、コンテンツ ルールに SSL 開始サービスを適用します。コンテンツ ルールでは、次の内容を定義します。

- コンテンツの物理的な格納場所
- コンテンツ要求を送信する場所（SSL 開始サービス）
- 使用するロード バランシング方式

HTTP サーバまたはバックエンド SSL サーバのコンテンツ ルールでは、各サービスに設定した IP アドレスが SSL プロキシ リスト内の SSL 開始サーバに設定した IP アドレスと一致する必要があります（「SSL 開始サーバの IP アドレスの設定」参照）。

SSL 開始のコンテンツ ルールには、レイヤ 5 のクッキーまたは URL ルールを指定できます。CSS はこのルールに含まれる情報を基に、使用するスティッキ サーバを検索したり、新しいクライアント要求に対して新しいサーバのロードバランスを取ったりすることができます。

レイヤ 5 スティッキとコンテンツ ルールの詳細については、『*Cisco Content Services Switch Content Load-Balancing Configuration Guide*』を参照してください。

SSL 開始のトラブルシューティング

SSL 開始設定のトラブルシューティングについては、次の説明を参照してください。

SSL プロキシ リストに関する問題が発生した場合は、次のように対処します。

- バックエンド サーバが initiation タイプに設定されていることを確認します。「SSL 開始サーバとしてのバックエンド サーバの設定」を参照してください。
- SSL プロキシ リストが **ssl-init** タイプのサービスに追加されていることと、そのサービスがアクティブになっていることを確認します。「SSL 開始のサービスの設定」を参照してください。
- SSL サービスがコンテンツ ルールに追加されていることと、そのコンテンツ ルールがアクティブになっていることを確認します。「SSL 開始のコンテンツ ルールの設定」を参照してください。

クライアント証明書に関する問題が発生した場合は、次のように対処します。

- SSL プロキシ リスト内の適切なバックエンド サーバの定義にクライアント証明書とキーが設定されていることを確認します。「クライアント証明書とキーの設定」を参照してください。
- バックエンド サーバを使用するサービスと同じタイプのサービスに SSL プロキシ リストが追加されていることを確認します。SSL 開始には **type ssl-init** コマンド、バックエンド SSL には **type ssl-accel-backend** コマンドを使用します。「SSL 開始のサービスの設定」を参照してください。
- SSL サービスがコンテンツ ルールに追加されていることと、そのコンテンツ ルールがアクティブになっていることを確認します。「SSL 開始のコンテンツ ルールの設定」を参照してください。
- SSL　サーバがクライアント証明書を要求するように設定されていることを確認します。
- バックエンド接続でスニファを使用し、サーバからクライアト証明書の要求が出されることと、CSS からその証明書が送信されることを確認します。