バックエンド SSL の設定

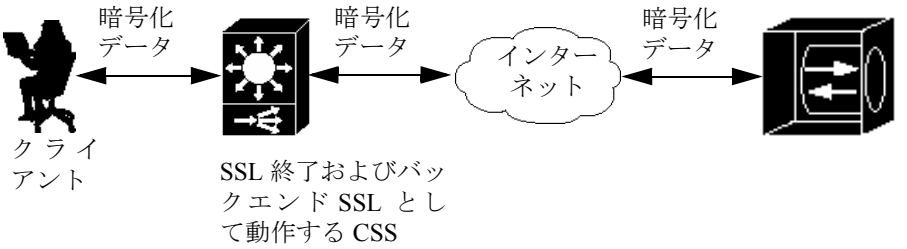
この章では、CSS でバックエンド SSL を設定するために必要な手順を説明します。この章の主な内容は次のとおりです。

バックエンド SSL の概要

バックエンド SSL を使用すると、CSS から SSL サーバとの接続を開始できます。さらに SSL 終了と共存させれば、クライアントと SSL サーバがセキュアなエンドツーエンド接続で結ばれます。

図 5-1 に、SSL 終了とバックエンド SSL が共存する構成を示します。

図 5-1　　SSL 終了とバックエンド SSL の共存



SSL モジュール、クライアント、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 サーバの定義で構成されており、この定義によって SSL サーバとの接続が開始されます。1 つの SSL プロキシ リストには、最大で 256 の仮想 SSL サーバまたはバックエンド 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 リストをサービスにすると、SSL 設定データの SSL モジュールへの転送を開始できます。サービスをアクティブ化すると、CSS はデータをこのモジュールに転送します。次に各 SSL サービスを SSL コンテンツ ルールに追加します。

SSL プロキシ リストの作成

SSL プロキシ リストは、同じ SSL サービスに関連付けられている、いくつかのバックエンド 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 リスト名には、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、その SSL プロキシ リストを削除する必要があります。

SSL プロキシ リストへの説明の追加

SSL 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 プロキシ リストでのバックエンド SSL サーバの設定

ここでは、SSL プロキシ リストに 1 つ以上のバックエンド SSL サーバを定義する方法について説明します。**backend-server** コマンドを使用して SSL プロキシ リスト内にインデックス エントリを作成し、このエントリにこのバックエンド 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 リストを使ってバックエンド 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 サーバのインデックス番号を指定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 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。

バックエンド サーバを SSL モジュールで使用するように定義するには、SSL プロキシ リストにそのバックエンド SSL サーバのエントリを作成し、設定を行う必要があります。サービスのアドレスに対応する IP アドレスと、サーバの IP アドレスを設定します。次に SSL プロキシ リストをアクティブ化します。

SSL プロキシ リストを設定してアクティブ化した後、そのリストをバックエンド SSL サービスに追加し、サービス タイプ **ssl-accel-backend** を割り当てます。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 プロキシ リスト内でのバックエンド SSL サーバのエントリの作成
- SSL バックエンド サーバの VIP アドレスの設定
- 仮想ポートの設定
- サーバの IP アドレスの設定
- 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
- SSL TCP 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

SSL プロキシ リスト内でのバックエンド SSL サーバのエントリの作成

SSL プロキシ リスト内にバックエンド SSL サーバの各パラメータを設定するには、その前にそのバックエンド SSL 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、SSL プロキシ リスト内でバックエンド SSL サーバに番号（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（VIP 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、次のように入力します。

```
(config-ssl-proxy-list[ssl_list3])# backend-server 1
```

SSL プロキシ リストからバックエンド サーバ 1 のエントリを削除するには、次のように入力します。

```
(config-ssl-proxy-list[ssl_list3])# no backend-server 1
```

バックエンド SSL サーバ タイプの設定

特に指定しない場合、バックエンド SSL サーバのタイプは **backend-ssl** になります。バックエンド SSL サーバがこのタイプの場合、CSS は次の処理を実行できます。

- クライアントから暗号化データを受信する。
- ロード バランシング用にデータを復号化する。
- データを再度暗号化し、その暗号化データを SSL 接続を通じて SSL サーバに送信する。

SSL 開始サーバを設定したが、それを同じプロキシ リスト内でバックエンド サーバとして設定し直す場合は、**backend-server** *number* **type backend-ssl** コマンドを使用します。

たとえば、SSL プロキシ リスト ssl_list3 内で SSL 開始サーバ 1 をバックエンド SSL サーバとして設定し直すときは、次のように入力します。

```
(config-ssl-proxy-list[ssl_list3])# backend-server 1 type backend-ssl
```

SSL 開始についての詳細は、第 6 章「SSL 開始の設定」を参照してください。

SSL バックエンド サーバの VIP アドレスの設定

バックエンド サーバの VIP 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 VIP アドレスは、サービスのアドレスです。

たとえば、バックエンド サーバ 1 に VIP 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 ip address
192.168.2.3
```

バックエンド サーバから VIP 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 VIP アドレスが設定されていない場合は、次のエラー 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。

```
SSL-server/Backend-server must have valid IP Address
```

仮想ポートの設定

バックエンド サーバのデフォルトの仮想ポートはポート 80 です。仮想ポートを介して、SSL モジュールから CSS にクリア テキストのデータ トラフィックが転送されます。SSL バックエンド サーバに異なる仮想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
```

サーバの IP アドレスの設定

サーバの IP アドレスはバックエンド SSL サーバに設定した IP アドレスと同じになります。バックエンド 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
```

バックエンド サーバから 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 server-ip
```

（注）　**active** コマンドを実行したときにサーバ IP アドレスが設定されていない場合、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。

```
SSL-server/Backend-server must have valid IP address
```

サーバ ポートの設定

バックエンド SSL サーバのデフォルトのポート番号は 443 です。SSL バックエンド 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

バックエンド サーバに向けて、SSL モジュールは SSL 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、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、ヘッダーが SSL バージョン 3 でメッセージが TLS バージョン 1 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、ヘッダーが SSL バージョン 3 でメッセージが TLS バージョン 1 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 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、ハードウェア アクセラレーションにより、サポートされたすべての暗号スイートは有効になっています。

SSL モジュールでサポートされるすべての暗号スイートと対応する値については、表 4-1 を参照してください。この値は、SSL バージョン 3.0 と TLS バージョン 1.0 で定義された値と同じです。この表にはまた、ソフトウェアの他のバージョンに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、そのバックエンド サーバについて、すべての暗号スイートが再度有効になります。このオプションは、特定の暗号を設定していない場合にだけ有効です。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）。

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 サーバに設定した暗号スイートを 1 つ以上削除するには、次のように入力します。

```
(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 セッションが再確立されます。

デフォルトでは、バックエンド SSL サーバによる、データ交換後の SSL 再ハンドシェイクは無効です（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、データ交換の後の再ハンドシェイクは無効になります。次に例を示します。

```
(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ると、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])# back-end-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。この 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。

クライアントとの TCP 接続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

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。クライアントと CSS モジュール間の 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されるこのタイムアウト値は、**ssl-server** *number* **tcp virtual 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）として、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 SSL 接続と TCP 接続のどちらについても、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、ACK を CSS がクライアントから受信したときに開始され、TCP 3 ウェイ ハンドシェイクを終了するまで行われます。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）として、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。デフォルトでは 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くと、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 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 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。このタイムアウト値は、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し、**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）として、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 SSL 接続と TCP 接続のどちらについても、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す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
```

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
```

SSL TCP 接続の Nagle アルゴリズムの指定

TCP Nagle アルゴリズムは、クライアントと SSL モジュール間またはバックエンド サーバと SSL モジュール間の 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。この処理では、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）で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 モジュール間の TCP 接続の Nagle アルゴリズムを無効にするには、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**

サーバと SSL モジュール間の 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 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 server nagle enable|disable**

サーバと SSL モジュール間の 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

ネットワークの速度が遅くて輻輳が発生している場合、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 20 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、次のように入力します。

```
(config-ssl-proxy-list[ssl_list1])# no backend-server 20 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 20 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、次のように入力します。

```
(config-ssl-proxy-list[ssl_list1])# no backend-server 20 tcp
buffer-share tx
```

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化する前に、仮想 SSL サーバまたはバックエンド SSL サーバの定義を 1 つ以上、リスト内に作成してください（この章で前述した「SSL プロキシ リストでの仮想 SSL サーバの設定」または「SSL TCP 接続の Nagle アルゴリズムの指定」参照）。

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、適切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。たとえば、次のように入力します。

```
(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。この章で後述する「バックエンド SSL のサービスの設定」を参照してください。

リスト内の仮想サーバまたはバックエンド SSL サーバを表示するには、**show ssl-proxy-list** 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行う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てしまいます。

バックエンド SSL 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、SSL のコンテンツ要求を転送できます。

（注）　CSS 内の SSL モジュールごとに 1 つのアクティブな SSL サービス（1 つのスロットにつき 1 つの SSL サービス）がサポートされます。1 つのスロットに対して複数の SSL サービスを設定できますが、アクティブにできるのは一度に 1 つの SSL サービスだけです。

このタイプのサービスをバックエンド 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。
- バックエンド サービスのキープアライブ タイプ（なし、ICMP、TCP、SSL、ネームド、スクリプト化、または 暗号化 HTTP）を設定する。暗号化 HTTP キープアライブは固定または非固定にできます。

  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。
- SSL プロキシ リストに、このタイプのサービスのバックエンド サーバを設定する。

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

ここでは、次の内容について説明します。

- SSL サービスの作成
- バックエンド SSL サービス タイプの設定

- バックエンド SSL サーバの SSL プロキシ リストへの追加
- バックエンド サービスのキープアライブ タイプ設定
- バックエンド SSL サービスの IP アドレスの設定
- バックエンド SSL サービスのポート番号の設定
- SSL サービスのアクティブ化
- SSL サービスの一時停止

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。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。

SLL 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、新しく作成されたサービスのモードに変わります。

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

バックエンド SSL サービス タイプの設定

バックエンド SSL サービスには、**ssl-accel-backend** サービス タイプを設定する必要があります。バックエンド SSL 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-accel-backend
```

バックエンド SSL サーバの SSL プロキシ リストへの追加

SSL プロキシ リストにバックエンド SSL サーバを設定したら、アクティブ リストを SSL サービスに追加し、CSS がバックエンド SSL サーバからの SSL のコンテンツ要求を処理する方法を定義します。バックエンド SSL サービスの設定方法は、サービス タイプを **ssl-accel-backend** 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、このタイプのサービスでは、バックエンド サーバを SSL プロキシ リストに定義する必要があります。

SSL プロキシ リストには、バックエンド SSL サービスの各種パラメータを定義します。SSL 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 リストにバックエンド サーバを設定する方法の詳細については、この章で前述した「バックエンド SSL サービス タイプの設定」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（この章の「SSL プロキシ リストの作成」参照）を入力します。

たとえば、SSL プロキシ リスト *ssl list3* をバックエンド SSL サービスに追加するには、次のように入力します。

```
(config-service[server1])# add ssl-proxy-list ssllist3
```

バックエンド サービスの 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[server1])# remove ssl-proxy-list ssllist3
```

バックエンド サービスのキープアライブ タイプ設定

ssl-accel-backend 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 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。

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、サーバから返されるフル SSL ハンドシェイクとデータを検証できます。バックエンド SSL サーバの場合、キープアライブは HTTP GET または HEAD を実行して、CSS 内のすべての SSL モジュールに対しブロードキャストします。キープアライブは SSL モジュールを選択すると、このモジュールにキープアライブ メッセージを送信し、キープアライブが失敗するまで続行します。その後、キープアライブは CSS 内の他の SSL モジュールを選択します。

（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、バックエンド 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。キープアライブによるモジュールの選択は、サービス タイプの設定によって異なります。すべてのサービスが ssl-accel-backend サービス タイプのバックエンド サービスである場合、キープアライブはすべてのモジュールを使用できます。設定に ssl-init サービス タイプの SSL 開始サービスが 1 つ含まれている場合、キープアライブはその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、バックエンド 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、バックエンド 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バックエンド SSL 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、別のバックエンド SSL 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、オリジナルのバックエンド 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他の値にします。次に、2 台めのバックエンド 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。このモジュールが実際の 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、バックエンド SSL 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。IP アドレスが 10.10.10.10 で ポート番号が 80 の HTTP GET 非固定キープアライブが SSL モジュールに送信されます。サービス s1 はバックエンド SSL サーバで稼動しているため、プロキシ リスト p1 が CSS 内のすべての SSL モジュールへブロードキャストされます。したがって、暗号化キープアライブはどの SSL モジュールも選択できます。キープアライブは、最初のキープアライブが失敗するまで 1 つのモジュールにとどまり、その後は CSS 内の他の SSL モジュールへ切り替えます。

```
!********************** SSL PROXY LIST ***********************
ssl-proxy-list p1
  backend-server 1
  backend-server 1 IP address 10.10.10.10
  backend-server 1 server-IP 10.10.10.10
  active

!************************** SERVICE **************************
service s1
  IP address 10.10.10.10
  port 80
  type ssl-accel-backend
  add ssl-proxy-list p1
  keepalive type http non-persistent encrypt
  keepalive method get
  keepalive uri "/index.html"
  active
```

次の例では、暗号化キープアライブがセッション ID の再使用をサポートするバックエンド サーバではなく別のハンドシェイクを使用する場合、他のバックエンド サーバをオリジナルのバックエンド サーバと同じ設定にする必要があることを示しています。ただし、ポート番号とセッション キャッシュ タイムアウト値は別の値にします。その後で、暗号化キープアライブのポートを、新しいバックエンド サーバが使用するポート番号と一致する設定にします。

IP アドレスが 10.10.10.10 で ポート番号が 81 の HTTP GET 非固定キープアライブが SSL モジュールに送信されます。この IP アドレスとポート番号は、リスト p1 内のバックエンド サーバ 2 の設定と一致します。バックエンド サーバ 1 のアドレスとポートの情報は、実際のバックエンド サーバのアドレスとポート番号であることに注意してください。このように、SSL モジュールは、フル SSL ハンドシェイクを実行することなくキープアライブ プローブを実サーバへ送信し続けます。なぜなら、セッション キャッシュのタイムアウトがゼロに設定されているからです。

```
!********************** SSL PROXY LIST **********************
ssl-proxy-list p1
  backend-server 1
  backend-server 1 IP address 10.10.10.10
  backend-server 1 port 80
  backend-server 1 server-IP 10.10.10.10
  backend-server 1 server-port 443
  backend-server 1 session-cache 500
  backend-server 2
  backend-server 2 IP address 10.10.10.10
  backend-server 2 port 81
  backend-server 2 server-IP 10.10.10.10
  backend-server 2 server-port 443
  backend-server 2 session-cache 0
  active

!************************** SERVICE **************************
service s1
  IP address 10.10.10.10
  port 80
  type ssl-accel-backend
  add ssl-proxy-list p1
  keepalive type http non-persistent encrypt
  keepalive port 81
  keepalive method get
  keepalive uri "/index.html"
  active
```

バックエンド SSL サービスの IP アドレスの設定

バックエンド SSL サービスの IP アドレスは、SSL プロキシ リストに設定されたバックエンド サーバの IP アドレスと一致させる必要があります。

たとえば、バックエンド SSL サービスに IP アドレス 10.11.21.13 を設定するには、次のように入力します。

```
(config-service[server1])# ip address 10.11.21.13
```

バックエンド SSL 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
```

バックエンド SSL サービスのポート番号の設定

CSS はポート番号を使用してクリア テキスト データを SSL モジュールに戻し、そこで再び暗号化が行われます。デフォルトでは、CSS はサービスに関連付けられているバックエンド コンテンツ ルールのポート番号、ポート 80 を使用します。このポート番号がバックエンド HTTP-SSL コンテンツ ルールと違う場合は、**port** コマンドを使用して設定します。

1 ～ 65535 の整数でポート番号を入力します。設定するポート番号は、SSL プロキシ リストでバックエンド サーバに設定した仮想ポート番号と一致させる必要があります。

たとえば、ポート番号 55 を設定するには、次のように入力します。

```
(config-service[server1])# port 55
```

バックエンド コンテンツ ルールのポート番号をリセットするには、次のように入力します。

```
(config-service[server1])# no port
```

SSL サービスのアクティブ化

SSL プロキシ リストの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。

- 仮想 SSL サーバの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-accel** 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-accel type service
  ```

- バックエンド SSL サーバの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-accel-backend** 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-accel type service
  ```

- サービスに追加した SSL プロキシ リストを、サービスをアクティブ化する前にアクティブにします。リストが一時停止されると、次のエラー メッセージがログに記録され、その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コンテンツ要求のためにサービス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
```

バックエンド SSL のコンテンツ ルールの設定

CSS がコンテンツへの SSL 要求を転送するように設定するには、コンテンツ ルールにバックエンド サービスを適用します。SSL コンテンツ ルールをアクティブ化して、コンテンツが実際に存在する場所、コンテンツ要求の送信先（SSL サービス）、および使用するロード バランシング方式を定義するまで、ネットワーク トラフィックは SSL モジュールに送信されません。

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、設定する VIP アドレスとポートを、SSL プロキシ リストの仮想 SSL サーバのエントリで設定した暗号化スイート パラメータ（「暗号スイートの指定」参照）と、必ず一致させてください。

バックエンド サーバでは、レイヤ 5 クッキーまたは URL ルールを指定できます。このルールの情報で、使用するスティッキ サーバを見つけたり、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。