CHAPTER 3

SSL 終了の設定

（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。

この章では、SSL 終了の仮想 SSL サーバとして Cisco ACE アプリケーション コントロール エンジン のコンテキストを設定するために必要な手順について説明します。この章の内容は、次のとおりです。

- SSL 終了の設定例

（注）　クライアントから ACE への SSL 接続が正常に開始されたことを確認するために、**show stats crypto server** コマンドの出力のハンドシェイク カウンタを監視できます（第 6 章「SSL 情報および統計情報の表示」を参照）。接続が成功するとハンドシェイク カウンタがインクリメントします。たとえば、SSLv3 Full Handshakes カウンタはハンドシェイクが正常に完了したことを示し、SSLv3 Resumed Handshakes カウンタはセッション ID を使用してハンドシェイクが正常に再開したことを示します。トラフィックが流れていると、これらのカウントがインクリメントします。障害が発生した場合は、アラートが送信され、受信カウンタもインクリメントします。

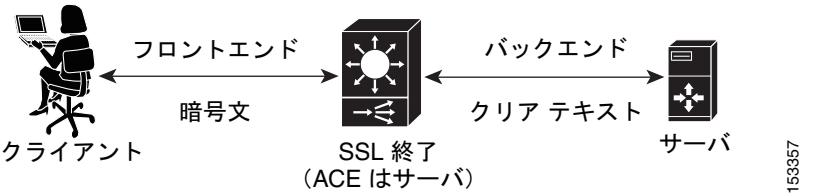
SSL 終了の概要

SSL 終了は、SSL プロキシ サーバとして動作する ACE がクライアントからの SSL 接続を終端し、続いて HTTP サーバと TCP 接続を確立するときに実行されます。ACE は、SSL 接続を終了すると、クライアントからの暗号文を復号化し、データをクリア テキストとして HTTP サーバに送信します。

図 3-1 に、ACE がクライアントとの SSL 接続を終端しているネットワーク接続を示します。

- クライアントと ACE の間：クライアントと、SSL プロキシ サーバとして動作する ACE との間の SSL 接続
- ACE とサーバの間：ACE と HTTP サーバとの間の TCP 接続

図 3-1　クライアントとの SSL 終了



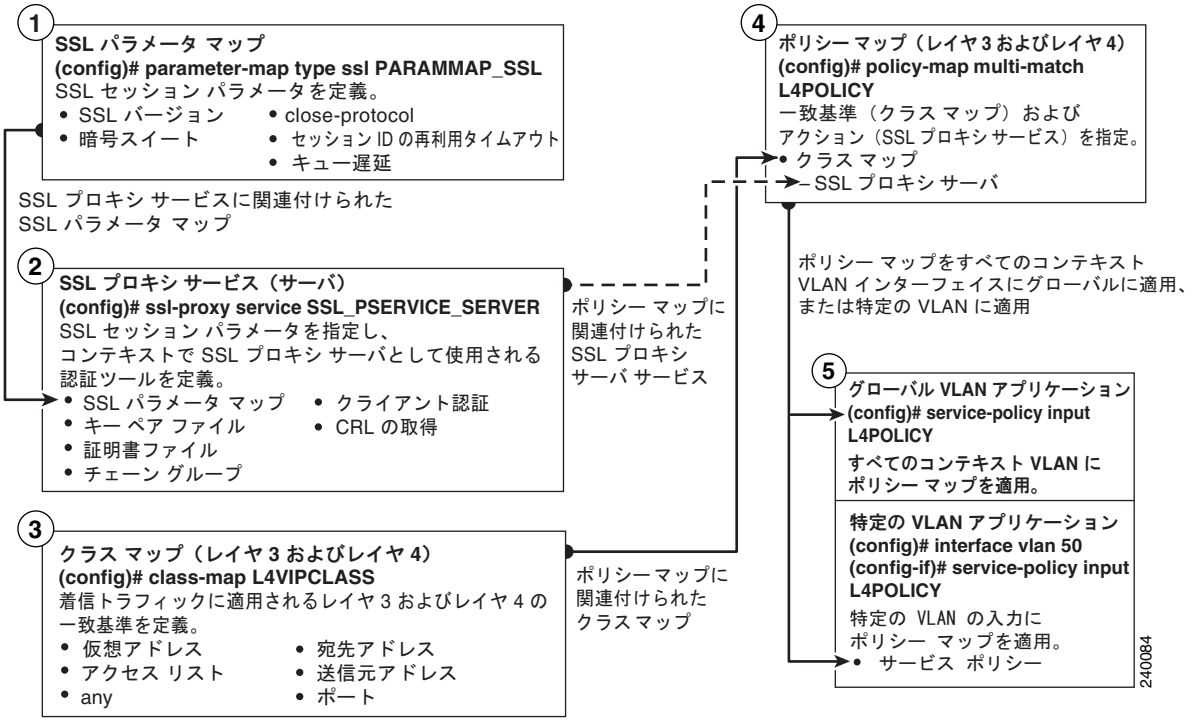
ACE は、パラメータ マップ、SSL プロキシ サービス、およびクラス マップを使用してポリシー マップを作成し、ポリシー マップによりクライアント、ACE、およびサーバの間の情報のフローが決まります。SSL 終了は、クライア

ントからのインバウンド トラフィック フローに含まれる宛先 IP アドレスに基づいているため、レイヤ 3 およびレイヤ 4 アプリケーションの 1 つです。この種類のアプリケーションの場合には、ACE がインバウンド トラフィックに適用するレイヤ 3 およびレイヤ 4 ポリシー マップをユーザが作成します。

SSL 終了のポリシー マップを設定する場合、ポリシー マップにパラメータ マップと SSL プロキシ サーバ サービスを関連付けて、SSL セッション パラメータや、証明書および RSA キー ペアなどのクライアント/サーバ認証ツールを定義します。また、ポリシー マップにクラス マップを関連付けて、仮想 SSL サーバ IP アドレスを定義します。このアドレスに、着信トラフィックの宛先 IP アドレスが一致する必要があります。一致する場合、ACE は、クライアントとネゴシエートして、SSL 接続を確立します。1 つのクラス マップに対して最大 250 の仮想 SSL サーバを定義できます。

図 3-2 に、SSL 終了で ACE によって使用されるレイヤ 3 およびレイヤ 4 のポリシー マップを作成して適用するプロセスの概要を示します。この図は、ポリシー マップ設定のさまざまなコンポーネントを互いに関連付ける方法も示します。

図 3-2　基本的な SSL 終了の設定のフロー図



ACE SSL 終了設定の前提条件

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。実サーバとサーバ ファームの設定時、実サーバをサーバ ファームに関連付けるときは、実サーバの適切なポート番号を割り当てるようにしてください。ポートを指定しなかった場合、ACE のデフォルトの動作によりインバウンド接続で使用された宛先ポートがアウトバウンド サーバ接続に割り当てられます。

たとえば、ACE への着信接続がセキュア クライアント HTTPS 接続であれば、通常はポート 443 が使用されます。実サーバにポート番号を割り当てないと、ACE はサーバへの接続にポート 443 を自動的に使用し、その結果、ACE はポー

ト 443 を通じたクリア テキスト HTTP 接続を確立します。この場合、通常はバックエンド サーバ接続用の発信宛先ポートとして 80、81、または 8080 を定義します。

SLB トラフィック ポリシー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 を設定したら、このガイドに記載されている SSL 終了用の SSL 設定要件を使用して、既存の SLB クラス マップおよびポリシー 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。

SSL 終了の設定のクイック スタート

表 3-1 は、SSL 終了用に ACE を設定するのに必要な手順の概要を示します。各手順には、その作業を完了するために必要な CLI コマンド、または手順の参照先が含まれています。各機能および CLI コマンドに関連付けられているすべてのオプションについての詳細は、表 3-1 以降のセクションを参照してください。

（注） このクイック スタートには、図 3-2 に示すパラメータ マップの作成手順は含まれません。ACE は、表 3-2 で説明するように、デフォルトのパラメータ マップ設定を使用します。

表 3-1　SSL 終了の設定のクイック スタート

作業およびコマンドの例

1. 複数のコンテキストで動作する場合は、CLI プロンプトを観察して、適切なコンテキストで動作しているかどうかを確認してください。必要に応じて、適切なコンテキストに直接ログインするか、または切り替えてください。
`host1/Admin# changeto C1`
`host1/C1#`
この表の残りの例では管理コンテキストを使用しています。コンテキスト作成の詳細については、『*Virtualization Guide, Cisco ACE Application Control Engine*』を参照してください。

2. コンフィギュレーション モードを開始します。
`host1/Admin# config`
`host1/Admin(config)#`

3. SSL プロキシ サーバ サービスを作成して、SSL サーバとして動作する ACE がポリシー マップに適用するハンドシェイク パラメータを定義します。
`host1/Admin(config)# ssl-proxy service SSL_PSERVICE_SERVER`
`host1/Admin(config-ssl-proxy)#`

4. 証明書および対応する RSA キー ペアを定義して、SSL プロキシ サーバ サービスを設定します。
`host1/Admin(config-ssl-proxy)# key MYRSAKEY_SERVER`
`host1/Admin(config-ssl-proxy)# cert MYCERT_SERVER`
`host1/Admin(config-ssl-proxy)# exit`
`host1/Admin(config)#`

表 3-1　SSL 終了の設定のクイック スタート （続き）

作業およびコマンドの例

5. レイヤ 3 およびレイヤ 4 クラス マップを作成して、必要に応じて入力トラフィック一致基準を設定します。
`host1/Admin(config)# class-map L4VIPCLASS`
`host1/Admin(config-cmap)# match virtual-address 2001:DB8:1::24 tcp any`
`or`
`host1/Admin(config-cmap)# match virtual-address 192.168.10.24 tcp any`
`host1/Admin(config-cmap)# exit`
`host1/Admin(config)#`

6. ポリシー マップを作成し、手順 5 で作成したクラス マップと関連付けます。
`host1/Admin(config)# policy-map multi-match L4POLICY`
`host1/Admin(config-pmap)# class L4VIPCLASS`
`host1/Admin(config-pmap-c)#`

7. 手順 3 で作成した SSL プロキシ サーバ サービスをポリシー マップと関連付けます。
`host1/Admin(config-pmap-c)# ssl-proxy server SSL_PSERVICE_SERVER`
`host1/Admin(config-pmap-c)# exit`
`host1/Admin(config-pmap)# exit`
`host1/Admin(config)#`

8. 次のように、目的のインターフェイスの入力トラフィックにポリシー マップを適用します。
コンテキストのすべての VLAN にポリシー マップをグローバルに適用します。
`host1/Admin(config)# service-policy input L4POLICY`
特定の VLAN にポリシー マップを適用します。
`host1/Admin(config)# interface vlan 50`
`host1/Admin(config-if)# service-policy input L4POLICY`

表 3-1　SSL 終了の設定のクイック スタート （続き）

作業およびコマンドの例
9. 実行コンフィギュレーションを表示して、追加した情報が正しく設定されているか確認します。 `host1/Admin(config-if)# do show running-config`
10. （任意）スタートアップ コンフィギュレーションに実行コンフィギュレーションをコピーして、変更をフラッシュ メモリに保存します。 `host1/Admin(config-if)# do copy running-config startup-config`

SSL パラメータ マップの作成および定義

SSL パラメータ マップでは、ACE が SSL プロキシ サービスに適用する SSL セッション パラメータを定義します。SSL パラメータ マップを作成すると、同じ SSL セッション パラメータを異なるプロキシ サービスに適用できます。表 3-2 では、各 SSL セッション パラメータとそれぞれのデフォルト値について説明します。

表 3-2　SSL パラメータ マップの SSL セッション パラメータ

SSL セッション パラメータ	説明	デフォルトの値 / 動作
Cipher suites	SSL ハンドシェイク時に ACE がサポートする暗号スイートを定義します（ACE がサポートする使用可能な暗号スイートのリストについては、表 3-3 を参照してください）。	ACE は、使用可能な暗号スイートをすべてサポートします。
Authentication-failure	クライアント認証が ACE で有効な場合、このパラメータを使用すると、ACE はクライアント認証の失敗が検出されたときに、SSL 終了設定のフロントエンド接続の確立を続行できます。	ACE は、クライアント認証の失敗が発生すると、SSL ハンドシェイクを終了します。

表 3-2　SSL パラメータ マップの SSL セッション パラメータ （続き）

SSL セッション パラメータ	説明	デフォルトの値 / 動作
CDP-errors ignore	**crl best-effort** コマンドが ACE で設定されている場合、このパラメータによって、ACE は CDP エラーによる認証の失敗を無視できます。	Disabled
Close-protocol	ACE が終了通知メッセージを実行する方法を定義します。	**none**：ACE は、セッションを終了するときに終了通知アラート メッセージを自身のピアに送信しますが、そのピアからの応答は想定しません。
Purpose-check disabled	このコマンドが設定されると、ACE は認証時に証明書に対する目的確認を実行できません。	Enabled
Rehandshake	再ハンドシェイクを有効化すると、ACE は、SSL HelloRequest メッセージをピアに送信して SSL ハンドシェイク ネゴシエーションを再開できます。	Disabled
Version	SSL ハンドシェイク時に ACE でサポートされる SSL および TLS のバージョンを定義します。	ACE がサポートするバージョンは、SSL3 と TLS1 です。
Queue delay time	クライアント用としてサーバからのパケット データを暗号化する前に ACE が保持する時間を定義します。	Disabled
Session cache timeout	ACE が新しい SSL セッションを確立するために新しい SSL ハンドシェイクが必要になるまでの、SSL セッション ID の有効期間を定義します。	Disabled
Expired CRL	CRL が期限切れになった場合に、ACE が受け取ったすべてのクライアント認証を拒否するかどうかを定義します。	Disabled

（注）　SSL プロキシ サービスが SSL セッション パラメータのデフォルト値を使用するようにする場合、SSL パラメータ マップを作成したり、プロキシ サービスと関連付けたりする必要はありません。SSL プロキシ サービスにパラメータ マップを関連付けないと、ACE は、表 3-2 にリストされているセッション パラメータのデフォルト値を自動的にプロキシ サービスに適用します。

SSL パラメータ マップを作成するには、**parameter-map type ssl** コマンドをコンフィギュレーション モードで使用します。

このコマンドの構文は次のとおりです。

parameter-map type ssl *parammap_name*

parammap_name 引数は、SSL パラメータ マップの名前です。最大 64 文字の英数字からなる文字列を引用符で囲まずに入力します。

たとえば、SSL パラメータ マップ PARAMMAP_SSL を作成するには、次のように入力します。

```
host1/Admin(config)# parameter-map type ssl PARAMMAP_SSL
```

SSL プロキシ パラメータ マップを作成したあと、CLI はパラメータ マップ SSL コンフィギュレーション モードになります。

```
host1/Admin(config-parammap-ssl)#
```

SSL セッション パラメータを定義せずにパラメータ マップ SSL コンフィギュレーション モードを終了すると、ACE は、表 3-2 にリストされているデフォルト値を使用してパラメータ マップを設定します。

既存の SSL パラメータ マップを削除するには、次のように入力します。

```
host1/Admin(config)# no parameter-map type ssl PARAMMAP_SSL
```

ここでは、次の内容について説明します。

- SSL パラメータ マップの説明の定義
- 暗号スイートの追加
- クライアント認証失敗時の SSL セッションのセットアップの続行
- CDP エラーによる認証の失敗を無視する ACE 設定
- close-protocol 動作の定義
- 証明書での目的確認の無効化

- SSL セッションの再ハンドシェイクの有効化
- SSL および TLS のバージョンの定義
- SSL キュー遅延の設定
- SSL セッション キャッシュ タイムアウトの設定
- 期限切れの CRL クライアント証明書の拒否

SSL パラメータ マップの説明の定義

SSL パラメータ マップ コンフィギュレーション モードで **description** コマンドを使用して、SSL パラメータ マップの簡単な説明を記述できます。このコマンドの構文は次のとおりです。

description *text*

text 引数には、スペースを含め最大 240 文字の英数字からなる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 パラメータ マップの説明を指定するには、次のコマンドを入力します。

```
host1/Admin(config)# parameter-map type ssl PARAMMAP_SSL
host1/Admin(config-parammap-conn)# description SSL parameter map
```

SSL パラメータ マップから説明を削除するには、次のように入力します。

```
host1/Admin(config-parammap-conn)# no description
```

暗号スイートの追加

SSL プロトコルは、以下のような操作で使用する、さまざまな暗号化アルゴリズムをサポートします。

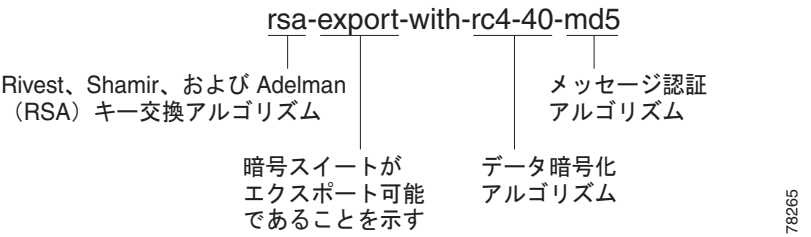
- サーバとクライアントを相互に認証する
- 証明書を送信する
- セッション キーを確立する

サポートする SSL バージョン、使用できる暗号化強度に関する企業ポリシー、SSL 対応ソフトウェアの輸出に関する政府規制などのさまざまな要因によって、クライアントとサーバがサポートする暗号スイート（暗号セット）が異なる可能

性があります。また、SSL ハンドシェイク プロトコルによって、サーバとクライアントが相互認証、証明書の送信、およびセッション キーの確立に使用する暗号スイートをどのようにネゴシエートするかが決まります。

図 3-3 に示すように、暗号スイートは、キー交換アルゴリズム、データ暗号化アルゴリズム、メッセージ認証（ハッシュ）アルゴリズムという、3 種類のアルゴリズムで構成されます。

図 3-3　暗号スイートのアルゴリズム



（注） 輸出可能な暗号スイートは、米国のソフトウェア製品輸出規制で定義されている他の暗号スイート（128 ビット暗号化の 3DES や RC4 など）ほどの強度はない暗号スイートです。輸出可能な暗号スイートは、米国からほとんどの国に輸出され、輸出可能な製品に強力な暗号化を提供します。

安全なセッション中に ACE がサポートする暗号スイートを定義するには、SSL パラメータ マップ コンフィギュレーション モードで **cipher** コマンドを使用します。ユーザが選択する暗号スイートはユーザの環境およびセキュリティ要件によって異なり、ACE にロードした証明書とキーに関連付けられている必要があります。

（注） デフォルトでは、ACE は、表 3-3 にリストされているすべての暗号スイートをサポートします。このデフォルト設定は、特定の暗号を使用して SSL パラメータ マップを設定しない場合にのみ有効です。すべての暗号スイートを使用する設定に戻すには、コマンドの **no** 形式を使用して、定義した暗号をすべてパラメータ マップから削除する必要があります。

このコマンドの構文は次のとおりです。

cipher *cipher_name* [**priority** *cipher_priority*]

次のキーワードと引数があります。

- *cipher_name*：ACE がサポートするように指定する暗号スイートの名前です。表 3-3 のリストは、ACE がサポートする暗号スイートを示しています。この表から、サポートされている暗号スイートの 1 つを入力します。
- **priority**：（任意）暗号スイートにプライオリティ レベルを割り当てます。プライオリティ レベルは、最も高い 10 から最も少ない 1 までで、暗号スイートの優先順位を表します。デフォルトでは、すべての設定された暗号スイートに 1 のプライオリティ レベルが付けられます。ACE は、どの暗号スイートを使用するかをネゴシエートする際に、最も高いプライオリティ レベルで設定されている暗号スイートに基づいて、クライアント リストから選択します。より高いプライオリティ レベルは、指定された暗号スイートに偏ります。SSL 終了アプリケーションの場合、ACE は、クライアントの ClientHello ハンドシェイク メッセージの暗号スイートに一致するプライオリティ レベルを使用します。SSL 開始アプリケーションの場合、プライオリティ レベルは、ACE がサーバへの ClientHello ハンドシェイク メッセージに暗号スイートを配置する順序を表します。
- *cipher_priority*：暗号スイートの優先度レベルです。1 ～ 10 の整数を入力します。デフォルトは 1 です。

たとえば、プライオリティ レベル 2 で rsa_with _aes_128_cbc_sha の暗号スイートを追加するには、次のように入力します。

```
host1/Admin(config)# parameter-map type ssl PARAMMAP_SSL
host1/Admin(config-parammap-ssl)# cipher rsa_with_aes_128_cbc_sha
priority 2
```

SSL パラメータ マップに含める各暗号スイートについて、**cipher** コマンドを繰り返します。

SSL パラメータ マップから暗号スイートを削除するには、次のように入力します。

```
host1/Admin(config-parammap-ssl)# no cipher rsa_with_aes_128_cbc_sha
```

表 3-3 は、ACE がサポートする使用可能な暗号化スイートを示します。また、サポートされている暗号スイートのうち、ACE からエクスポート可能なものを示します。この表では、各暗号スイートに必要な認証証明書および暗号キーも示します。

表 3-3 にリストされている暗号スイートを ACE が暗黙的にサポートするデフォルト設定を使用する場合、または各暗号スイートを等しいプライオリティで明示的に定義し、クライアント接続が複数の暗号を使用する場合、ACE は、暗号スイートを表に示されるのと同じ順序で RSA_WITH_RC4_128_MD5 からピアに送信します。

注意　タイトルに「export」が含まれている暗号スイートは、米国以外での使用を目的としており、キー サイズが制限さている暗号化アルゴリズムを持ちます。

表 3-3　ACE でサポートされる SSL 暗号スイート

暗号スイート	輸出可能	使用される認証証明書	使用するキー交換アルゴリズム
RSA_WITH_RC4_128_MD5	No	RSA 証明書	RSA キー交換
RSA_WITH_RC4_128_SHA	No	RSA 証明書	RSA キー交換
RSA_WITH_DES_CBC_SHA	No	RSA 証明書	RSA キー交換
RSA_WITH_3DES_EDE_CBC_SHA	No	RSA 証明書	RSA キー交換
RSA_WITH_AES_128_CBC_SHA	No	RSA 証明書	RSA キー交換
RSA_WITH_AES_256_CBC_SHA	No	RSA 証明書	RSA キー交換
RSA_EXPORT_WITH_RC4_40_MD5	Yes	RSA 証明書	RSA キー交換
RSA_EXPORT1024_WITH_RC4_56_MD5	Yes	RSA 証明書	RSA キー交換
RSA_EXPORT_WITH_DES40_CBC_SHA	Yes	RSA 証明書	RSA キー交換
RSA_EXPORT1024_WITH_DES_CBC_SHA	Yes	RSA 証明書	RSA キー交換
RSA_EXPORT1024_WITH_RC4_56_SHA	Yes	RSA 証明書	RSA キー交換

クライアント認証失敗時の SSL セッションのセットアップの続行

デフォルトでは、クライアント認証が有効な場合、SSL 終了設定でのフロントエンド接続のセットアップ時に次のクライアント認証失敗のいずれかが ACE で発生すると、ACE は SSL ハンドシェイクを終了します。

- Certificate is not yet valid

- Certificate has expired
- Unable to get issuer certificate
- Certificate is revoked
- No client certificate is sent
- Certificate signature failure
- CRL is not available during the revocation check
- CRL is expired during revocation check
- All other certificate errors

パラメータ マップ SSL コンフィギュレーション モードで **authentication-failure** コマンドを使用して、これらのエラーを無視して SSL ハンドシェイクを継続するか、ハンドシェイクの完了後に HTTP リダイレクトを実行するように ACE を設定できます。

このコマンドの構文は次のとおりです。

authentication-failure {**ignore** | **redirect** *reason* {**serverfarm** *serverfarm_name* | **url** *URL_string* {**301**|**302**}}}

キーワード、引数、およびオプションは次のとおりです。

- **ignore**：SSL ハンドシェイク時にすべての証明書の失敗を無視し、証明書の失敗があったとしても ACE は SSL 接続を確立できます。**authentication-failure ignore** コマンドを 1 つ以上の **authentication-failure redirect** コマンドと組み合わせると、ACE は指定された個々のエラーをリダイレクトし、残りのエラーを無視します。

  authentication-failure redirect any コマンドは **authentication-failure ignore** コマンドと設定できません。

- **redirect** *reason*：ハンドシェイクの完了後の証明書の失敗に対して指定された *reason* 引数が ACE で発生した場合に、指定したサーバ フォームまたは URL にリダイレクトを実行します。

  失敗の理由が複数の場合、それぞれの理由に対して **authentication-failure redirect** コマンドを設定します。

  複数の失敗によりリダイレクトが行われる場合、ACE は、発生した最初の失敗に対してリダイレクトを実行します。失敗が修正されると、ACE は、次に発生した失敗にリダイレクトを実行します。

  reason 引数には、次のキーワードのいずれかを入力します。

  - **cert-not-yet-valid**：リダイレクトとまだ有効な失敗ではない証明書を関連付けます。
  - **cert-expired**：リダイレクトと期限切れの証明書の失敗を関連付けます。
  - **unknown-issuer**：リダイレクトと発行元が不明な証明書の失敗を関連付けます。
  - **cert-revoked**：リダイレクトと失効した証明書の失敗を関連付けます。
  - **no-client-cert**：リダイレクトとクライアント認証がない失敗を関連付けます。
  - **crl-not-available**：リダイレクトと使用できない CRL の失敗を関連付けます。
  - **crl-has-expired**：リダイレクトと期限切れの CRL の失敗を関連付けます。
  - **cert-has-signature-failure**：リダイレクトと証明書の署名の失敗を関連付けます。
  - **cert-other-error**：リダイレクトと他のすべての証明書の失敗を関連付けます。
  - **any**：リダイレクトと任意の証明書の失敗を関連付けます。リダイレクトの個別の理由で **authentication-failure redirect any** コマンドを設定できます。設定すると、ACE は **any** の理由を使用する前に個別の理由の 1 つへの一致を試みます。**authentication-failure redirect any** コマンドは **authentication-failure ignore** コマンドと設定できません。

- **serverfarm** *serverfarm_name*：ロード バランシング用に設定されたサーバ ファームの名前を指定します。ホストまたはリダイレクト サーバ ファームを設定できます。

  ホスト サーバ ファームを設定する場合、ソーリー サーバとして実サーバを含めます。実サーバは HTTP サーバである必要があり、ポート番号を指定する必要があります。リダイレクトの失敗が発生すると、ACE は別の VIP を含めないで、実サーバにクライアント接続を直接転送します。

  SSL リダイレクトの追加中にサーバ ファーム ID を使い切ると、ACE は次のメッセージを表示します。

  ```
  "Number of sfarms in the config have reached the maximum limit!"
  ```

- **url** *URL_string*：リダイレクトのスタティック URL パスを指定します。最大 255 文字の文字列をスペースを入れずに入力します。

- **301|302**：クライアントに送信されるリダイレクト コードを指定します。次のいずれか 1 つを入力します。
  - **301**：新しい場所に完全に移動するリソースのステータス コード。
  - **302**：新しい場所に一時的に移動するリソースのステータス コード。

たとえば、クライアント認証の失敗を無視するには、次のように入力します。

```
host1/Admin(config)# parameter-map type ssl SSL_PARAMMAP_SSL
host1/Admin(config-parammap-ssl)# authentication-failure ignore
```

任意のクライアント認証の失敗が発生した場合にサーバ ファームにリダイレクトを実行するには、次のように入力します。

```
host1/Admin(config-parammap-ssl)# authentication-failure redirect any
serverfarm SFARM2
```

不明発行元の失敗が発生した場合に 302 ステータス コードでスタティック URL にリダイレクトを実行するには、次のように入力します。

```
host1/Admin(config-parammap-ssl)# authentication-failure redirect
unknown-issuer url http://www.eng.com 302
```

クライアント認証失敗時に SSL ハンドシェイクを終了するデフォルト動作をリセットするには、このコマンドの **no** 形式を使用します。

```
host1/Admin(config-parammap-ssl)# no authentication-failure redirect
unknown-issuer
```

次の設定例は、クライアント認証の失敗のタイプに応じて、サーバ ファームまたは直接 URL に HTTP リダイレクトを実行するように ACE に指示します。

IPv6 の例

```
crypto authgroup AUTH-GROUP1

access-list EVERYONE line 8 extended permit ip anyv6 anyv6

rserver redirect EXPIRED
  webhost-redirection https://%h/support/expiredclientcert.html 302
  inservice
rserver redirect INVALID
  webhost-redirection https://%h/support/invalidclientcert.html 302
  inservice
rserver host SERVER1
  ip address 2001:DB8:1::10
  inservice
rserver host SERVER2
```

```
  ip address 2001:DB8:1::20
  inservice

serverfarm redirect EXPIRED-CERT
  rserver EXPIRED
    inservice
serverfarm redirect INVALID-CERT
  rserver INVALID
    inservice
serverfarm host WEB
  rserver SERVER1 80
    inservice
  rserver SERVER2 80
    inservice

parameter-map type ssl SSLPARAM
  authentication-failure redirect cert-not-yet-valid serverfarm
INVALID-CERT
  authentication-failure redirect cert-expired serverfarm EXPIRED-CERT
  authentication-failure redirect unknown-issuer url
https://www.example.com/NewCertRequest.html 302

ssl-proxy service SSLTERM-CLIENTAUTH
  ssl advanced-options SSLPARAM

class-map match-all CMAP-HTTPS
  2 match virtual-address 2001:DB:2::100 tcp eq https

policy-map type management first-match MGMTPOLICY
  class class-default
    permit

policy-map type loadbalance first-match SLB
  class class-default
    serverfarm WEB

policy-map multi-match VIPS
  class CMAP-HTTPS
    loadbalance vip inservice
    loadbalance policy SLB
    loadbalance vip icmp-reply

interface vlan 20
  ip address 2001:DB:2::1/64
  access-group input EVERYONE
  service-policy input VIPS
  no shutdown
interface vlan 40
```

```
  ip address 2001:DB:1::1/64
  service-policy input MGMTPOLICY
  no shutdown
```

IPv4 の例

```
crypto authgroup AUTH-GROUP1

access-list EVERYONE line 8 extended permit ip any any

rserver redirect EXPIRED
  webhost-redirection https://%h/support/expiredclientcert.html 302
  inservice
rserver redirect INVALID
  webhost-redirection https://%h/support/invalidclientcert.html 302
  inservice
rserver host SERVER1
  ip address 192.168.1.10
  inservice
rserver host SERVER2
  ip address 192.168.1.20
  inservice

serverfarm redirect EXPIRED-CERT
  rserver EXPIRED
    inservice
serverfarm redirect INVALID-CERT
  rserver INVALID
    inservice
serverfarm host WEB
  rserver SERVER1 80
    inservice
  rserver SERVER2 80
    inservice

parameter-map type ssl SSLPARAM
  authentication-failure redirect cert-not-yet-valid serverfarm
INVALID-CERT
  authentication-failure redirect cert-expired serverfarm EXPIRED-CERT
  authentication-failure redirect unknown-issuer url
https://www.example.com/NewCertRequest.html 302

ssl-proxy service SSLTERM-CLIENTAUTH
  ssl advanced-options SSLPARAM

class-map match-all CMAP-HTTPS
  2 match virtual-address 172.16.1.100 tcp eq https
```

```
policy-map type management first-match MGMTPOLICY
  class class-default
    permit

policy-map type loadbalance first-match SLB
  class class-default
    serverfarm WEB

policy-map multi-match VIPS
  class CMAP-HTTPS
    loadbalance vip inservice
    loadbalance policy SLB
    loadbalance vip icmp-reply

interface vlan 20
  ip address 172.16.1.1 255.255.255.0
  access-group input EVERYONE
  service-policy input VIPS
  no shutdown
interface vlan 40
  ip address 192.168.1.1 255.255.255.0
  service-policy input MGMTPOLICY
  no shutdown
```

CDP エラーによる認証の失敗を無視する ACE 設定

デフォルトでは、クライアント証明書失効用に **crl best-effort** コマンドを設定すると、ACE が提示された証明書内の CRL 分散ポイント（CDP）エラー、または CRL ダウンロード時に発生したエラーを検出した場合、ACE はその SSL 接続を拒否します。

cdp-errors ignore コマンドでは、**crl best-effort** コマンドが設定されている場合に、SSL パラメータ マップが CDP エラーやダウンロード エラーを無視するように設定できます。**cdp-errors ignore** コマンドを設定する場合、ACE では、提示された証明書内に CDP エラーを検出した場合や、有効な証明書失効リスト（CRL）を証明書上の有効な CDP からダウンロードできない場合に、SSL 接続が許可されます。パラメータ マップ SSL コンフィギュレーション モードでのこのコマンドの構文は次のとおりです。

cdp-errors ignore

たとえば、CDP エラーを無視するように ACE を設定する場合は、次のように入力します。

```
host1/Admin(config)# parameter-map type ssl PARAMMAP_SSL
```

```
host1/Admin(config-parammap-ssl)# cdp-errors ignore
```

CDP エラーが発生した場合に ACE が SSL 接続を拒否するデフォルトの動作をリセットするには、**no** 形式の **cdp-errors ignore** コマンドを使用します。例を示します。

```
host1/Admin(config-parammap-ssl)# no cdp-errors ignore
```

提示された SSL 証明書内にある CDP エラーを ACE が無視し、SSL 接続を許可した回数を表示するには、**show crypto cdp-errors** コマンドを使用します。このコマンドは、[Best Effort CDP Errors Ignored] フィールドの出力を表示します。

close-protocol 動作の定義

パラメータ マップ SSL コンフィギュレーション モードで **close-protocol** コマンドを使用して、ACE が終了通知メッセージの送信を処理する方法を設定できます。このコマンドの構文は次のとおりです。

close-protocol {**disabled** | **none**}

キーワードは次のとおりです。

- **disabled** : セッションを閉じるときに ACE が終了通知アラート メッセージをピアに送信しないように指定します。ピアからの応答は想定しません。
- **none** : セッションを閉じるときに ACE が終了通知アラート メッセージをピアに送信するように指定します。ピアからの応答は想定しません。

たとえば、**close-protocol** を disabled に設定するには、次のように入力します。

```
host1/Admin(config)# parameter-map type ssl SSL_PARAMMAP_SSL
host1/Admin(config-parammap-ssl)# close-protocol disabled
```

none のデフォルト設定で **close-protocol** コマンドを設定するには、このコマンドの **no** 形式を使用します。

```
host1/Admin(config-parammap-ssl)# no close-protocol
```

証明書での目的確認の無効化

デフォルトでは、証明書チェーンのクライアント認証中に、ACE は次の場合に basicContraint フィールドの目的確認を実行します。

- クライアント証明書に CA FALSE 設定がある場合。
- 中間証明書に CA TRUE 設定がある場合。

このフィールドにこれらの設定がない場合、証明書の認証は失敗します。

証明書の認証時に ACE が目的確認をする必要がないと判断した場合は、パラメータ マップ SSL コンフィギュレーション モードで **purpose-check disabled** コマンドを使用して無効にすることができます。

このコマンドの構文は次のとおりです。

purpose-check disabled

たとえば、目的確認を無効にするには、次のように入力します。

```
host1/Admin(config)# parameter-map type ssl SSL_PARAMMAP_SSL
host1/Admin(config-parammap-ssl)# purpose-check disabled
```

目的確認を実行するデフォルト設定を再度有効にするには、このコマンドの **no** 形式を使用します。

```
host1/Admin(config-parammap-ssl)# no purpose-check disabled
```

SSL セッションの再ハンドシェイクの有効化

デフォルトでは、SSL セッションの再ハンドシェイクは無効になっています。SSL セッションの再ハンドシェイク機能を有効化するには、パラメータ マップ SSL コンフィギュレーション モードで **rehandshake enabled** コマンドを使用します。

このコマンドの構文は次のとおりです。

rehandshake enabled

（注） コンテキストのすべての VIP の SSL 再ハンドシェイクを有効にするには、「コンテキストのすべての VIP の SSL 再ハンドシェイクの有効化」の項を参照してください。

たとえば、SSL の再ハンドシェイク機能を有効にするには、次のように入力します。

```
host1/Admin(config)# parameter-map type ssl PARAMMAP_SSL
host1/Admin(config-parammap-ssl)# rehandshake enabled
```

再ハンドシェイク機能を無効にするには、次のように入力します。

```
host1/Admin(config-parammap-ssl)# no rehandshake enabled
```

rehandshake enabled コマンドのステータスを表示するには、**show parameter-map** コマンドを使用します。

SSL および TLS のバージョンの定義

ピアとの SSL ハンドシェイク時に ACE がサポートするセキュリティ プロトコルのバージョンを指定するには、パラメータ マップ SSL コンフィギュレーション モードで **version** コマンドを使用します。

このコマンドの構文は次のとおりです。

version {**all** | **ssl3** | **tls1**}

キーワードは次のとおりです。

- **all**：（デフォルト）ACE は、SSL バージョン 3.0 と Transport Layer Security（TLS）バージョン 1.0 の両方をサポートします。
- **ssl3**：ACE は、SSL バージョン 3.0 だけをサポートします。
- **tls1**：ACE は、TLS バージョン 1.0 だけをサポートします。

たとえば、パラメータ マップ用に SSL バージョン 3.0 を指定するには、次のように入力します。

```
host1/Admin(config)# parameter-map type ssl PARAMMAP_SSL
host1/Admin(config-parammap-ssl)# version ssl3
```

SSL プロキシ パラメータ マップからセキュリティ プロトコルのバージョンを削除するには、次のように入力します。

```
host1/Admin(config-parammap-ssl)# no version tlsl
```

SSL キュー遅延の設定

ACE は、サーバからのパケット データをクライアントへの送信のために暗号化する前に、キューに入れます。ACE は、次のイベントの 1 つが発生すると、暗号化のキューからデータを解放します。

- キューが 4096 バイトに到達する。

- サーバが TCP-FIN セグメントを送信する。
- キューが 4096 バイトに到達しなくても、ACE のキューの遅延時間が超過する。

キューの遅延時間は、暗号化のためにキュー内のデータを解放するまで ACE が待機する時間です。デフォルトでは、キュー遅延タイマーは無効になっています。パラメータ マップ SSL コンフィギュレーション モードで **queue-delay timeout** コマンドを使用すると、遅延時間を設定できます。このコマンドの構文は次のとおりです。

queue-delay timeout *milliseconds*

milliseconds 引数は、データがキューから解放されるまでの時間（ミリ秒単位）です。0 ～ 10000 の整数を入力します。値 0 は遅延タイマーを無効にし、ACE はサーバから到着したデータを暗号化し、暗号化されたデータをクライアントに送信します。

（注） このキュー遅延は、ACE がそのクライアントに送信する暗号化されたデータにのみ適用されます。

たとえば、キュー遅延時間を 500 ミリ秒に設定するには、次のように入力します。

```
host1/Admin(config-parammap-ssl)# queue-delay timeout 500
```

キュー遅延タイマーを無効にするには、次のように入力します。

```
host1/Admin(config-parammap-ssl)# no queue-delay timeout
```

SSL セッション キャッシュ タイムアウトの設定

クライアントおよび ACE が完全な SSL キー交換を実行し、新しいマスター秘密キーを確立するたびに、SSL セッション ID が作成されます。クライアントと ACE との SSL ネゴシエーション プロセスを迅速化するため、SSL セッション ID の再利用機能により、ACE はセッション キャッシュ内の秘密キー情報を再利用できます。クライアントのその後の接続では、ACE が、最後のネゴシエートされたセッションでキャッシュに格納されたキーを再利用します。ACE は、セッションのキャッシュに最大 250,000（ACE モジュール）または 100,000（ACE アプライアンス）の SSL セッション ID を格納できます。

デフォルトでは、SSL セッション ID の再利用は、ACE で無効になっています。ACE で新しいセッションを確立するために完全な SSL ハンドシェイクが必要になるまで SSL セッション ID が有効になる時間の合計について、セッション キャッシュ タイムアウト値を設定することによって、セッション ID の再利用を有効にできます。

パラメータ マップ SSL コンフィギュレーション モードで **session-cache timeout** コマンドを使用して、セッション キャッシュ タイムアウトを設定できます。このコマンドの構文は次のとおりです。

session-cache timeout *seconds*

seconds 引数は、ACE がセッション ID を削除する前にキャッシュ内に格納されているキーを再利用する秒単位の時間です。0 ～ 72000（20 時間）の整数を入力します。デフォルトでは、セッション ID の再利用は無効です。値が 0 の場合、キャッシュが満杯になると ACE がキャッシュからセッション ID を削除し、Least Recently Used（LRU）タイムアウト ポリシーを適用します。

たとえば、600 秒にセッション キャッシュ タイムアウトを設定するには、次のように入力します。

```
host1/Admin(config-parammap-ssl)# session-cache timeout 600
```

タイマーを無効にし、ACE との新しい接続ごとに SSL 完全ハンドシェイクが発生するようにするには、次のように入力します。

```
host1/Admin(config-parammap-ssl)# no session-cache timeout
```

コンテキストのセッション キャッシュ情報をクリアするには、**clear crypto session-cache** コマンドを使用します。このコマンドの構文は次のとおりです。

clear crypto session-cache [**all**]

オプションのキーワード **all** では、すべてのコンテキストのすべてのセッション キャッシュ情報がクリアされます。このオプションを使用できるのは、管理コンテキストのみです。

期限切れの CRL クライアント証明書の拒否

「クライアント認証中の CRL の使用」に説明されているように、クライアント認証用に証明書失効リスト（CRL）を ACE 上で設定すると、この CRL には、新しいバージョンが使用可能になる日付を指定するための更新フィールドが含まれ

ます。デフォルトでは、ACE は更新フィールドが期限切れの日付になっている CRL を使用しません。そのため、CRL を使用したクライアント証明書の受け取りを拒否しません。

使用されている CRL が期限切れの場合にクライアント証明書を拒否するように ACE を設定するには、パラメータ マップ SSL コンフィギュレーション モードで **expired-crl reject** コマンドを使用します。このコマンドの構文は次のとおりです。

expired-crl reject

例を示します。

```
host1/Admin(config-parammap-ssl)# expired-crl reject
```

使用中の CRL の期限が切れた後もクライアント認証を受け入れる ACE のデフォルト動作にをリセットするには、次のように入力します。

```
host1/Admin(config-parammap-ssl)# no expired-crl reject
```

コンテキストのすべての VIP の SSL 再ハンドシェイクの有効化

デフォルトでは、SSL 再ハンドシェイクは、コンテキストのすべての VIP に対して無効になっています。コンテキストのすべての VIP の SSL 再ハンドシェイクを有効にするには、コンフィギュレーション モードで **crypto rehandshake enabled** コマンドを使用します。このコマンドの構文は次のとおりです。

crypto rehandshake enabled

SSL プロキシ サービスのパラメータ マップでの SSL 再ハンドシェイクの有効化の詳細については、「SSL セッションの再ハンドシェイクの有効化」の項を参照してください。

（注）　**crypto rehandshake enabled** コンフィギュレーション モード コマンドは、SSL プロキシ サービスで個別に設定できる **rehandshake enable** パラメータ マップ コマンドよりも優先されます。

たとえば、コンテキストのすべての VIP の SSL 再ハンドシェイクを有効化するには、次のコマンドを入力します。

```
host1/Admin(config)# crypto rehandshake enabled
```

ACE の動作を、コンテキストのすべての VIP に対する再ハンドシェイクを無効にするデフォルトに戻すには、次のコマンドを入力します。

```
host1/Admin(config)# no crypto rehandshake enabled
```

SSL プロキシ サービスの作成および定義

SSL プロキシ サービスは、ACE が SSL ハンドシェイク時に使用する SSL パラメータ マップ、キー ペア、証明書、およびチェーン グループを定義します。SSL 終了の場合は、SSL プロキシ サーバ サービスで ACE を設定します。これは、ACE が SSL サーバとして機能するためです。

コンフィギュレーション モードで **ssl-proxy service** コマンドを使用して、SSL プロキシ サーバ サービスを作成できます。

このコマンドの構文は次のとおりです。

ssl-proxy service *pservice_name*

pservice_name 引数は SSL プロキシ サーバ サービスの名前です。最大 64 文字の英数字からなる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 サーバ サービス PSERVICE _SERVER を作成するには、次のように入力します。

```
host1/Admin(config)# ssl-proxy service PSERVICE_SERVER
```

SSL プロキシ サーバ サービスを作成したあと、CLI は SSL プロキシ コンフィギュレーション モードになります。

```
host1/Admin(config-ssl-proxy)#
```

既存の SSL プロキシ サーバ サービスを削除するには、次のように入力します。

```
host1/Admin(config)# no ssl-proxy PSERVICE_SERVER
```

ここでは、次の内容について説明します。

- SSL プロキシ サーバ サービスと SSL パラメータ マップの関連付け
- キー ペアの指定

- 証明書の指定
- 証明書チェーン グループの指定
- クライアント認証のイネーブル化
- クライアント認証中の CRL の使用
- CRL のダウンロード場所の設定
- グローバルな CRL パラメータの設定

SSL プロキシ サーバ サービスと SSL パラメータ マップの関連付け

SSL プロキシ サーバ サービスと SSL パラメータ マップを関連付けるには、**ssl advanced-options** コマンドを SSL プロキシ コンフィギュレーション モードで使用します。

このコマンドの構文は次のとおりです。

ssl advanced-options *parammap_name*

parammap_name 引数は、既存の SSL パラメータ マップの名前です（「SSL パラメータ マップの作成および定義」の項を参照）。最大 64 文字の英数字からなる文字列を引用符で囲まずに入力します。

たとえば、パラメータ マップ PARAMMAP_SSL を SSL プロキシ サービスと関連付けるには、次のように入力します。

```
host1/Admin(config)# ssl-proxy service PSERVICE_SERVER
host1/Admin(config-ssl-proxy)# ssl advanced-options PARAMMAP_SSL
```

SSL プロキシ サービスと SSL パラメータ マップの関連付けを削除するには、次のように入力します。

```
host1/Admin(config-ssl-proxy)# no ssl advanced-options PARAMMAP_SSL
```

キー ペアの指定

データ暗号化のために ACE が SSL ハンドシェイク時に使用するキー ペアを、SSL プロキシ コンフィギュレーション モードで **key** コマンドを使用して指定できます。

（注） ユーザが選択するキー ペア ファイル内の公開キーは、選択する証明書に埋め込まれた公開キーに一致する必要があります（「証明書の指定」を参照）。公開キーの一致確認の詳細については、第 2 章「証明書およびキーの管理」の「キー ペアと比較した証明書の確認」の項を参照してください。

このコマンドの構文は次のとおりです。

key {*key_filename* | **cisco-sample-key**}

引数とキーワードは次のとおりです。

- *key_filename* : ACE にロードされた既存のキー ペア ファイルの名前です。最大 40 文字の英数字からなる文字列を引用符で囲まずに入力します。
- **cisco-sample-key** : ACE にプレインストールされている、cisco-sample-key という名前のサンプル RSA 1024 ビット キー ペアです。このファイルは、同じファイル名で、任意のコンテキストで使用できます。このキー ペアの詳細については、「ACE のサンプル証明書とキー ペアの使用」の項を参照してください。

たとえば、キー ペア ファイル MYKEY.PEM の秘密キーを指定するには、次のように入力します。

```
host1/Admin(config)# ssl-proxy service PSERVICE_SERVER
host1/Admin(config-ssl-proxy)# key MYKEY.PEM
```

SSL プロキシ サービスから秘密キーを削除するには、次のように入力します。

```
host1/Admin(config-ssl-proxy)# no key MYKEY.PEM
```

証明書の指定

ACE がアイデンティティを証明するために SSL ハンドシェイク プロセスで使用する証明書を、SSL プロキシ コンフィギュレーション モードで **cert** コマンドを使用して指定できます。

（注） 選択した証明書に埋め込まれた公開キーは、選択したキー ペア ファイル内の公開キーに一致する必要があります（「キー ペアの指定」の項を参照）。公開キーの一致確認の詳細については、第 2 章「証明書およびキーの管理」の「キー ペアと比較した証明書の確認」の項を参照してください。

このコマンドの構文は次のとおりです。

cert {*cert_filename* | **cisco-sample-cert**}

引数とキーワードは次のとおりです。

- *cert_filename*：ACE にロードされた既存の証明書ファイルの名前です。最大 40 文字の英数字からなる文字列を引用符で囲まずに入力します。
- **cisco-sample-cert**：ACE にプレインストールされた cisco-sample-cert という名前の自己署名証明書を指定します。このファイルは、同じファイル名で、任意のコンテキストで使用できます。この証明書の詳細については、「ACE のサンプル証明書とキー ペアの使用」の項を参照してください。

たとえば、証明書ファイル MYCERT.PEM に証明書を指定するには、次のように入力します。

```
host1/Admin(config)# ssl-proxy service PSERVICE_SERVER
host1/Admin(config-ssl-proxy)# cert MYCERT.PEM
```

SSL プロキシ サービスから証明書ファイルを削除するには、次のように入力します。

```
host1/Admin(config-ssl-proxy)# no cert MYCERT.PEM
```

証明書チェーン グループの指定

ACE が SSL ハンドシェイク時にピアに送信する証明書チェーンを、SSL プロキシ コンフィギュレーション モードで **chaingroup** コマンドを使用して指定できます。ACE には、SSL プロキシ サービスに指定した証明書付きの証明書チェー

ンが含まれます（「証明書の指定」の項を参照）。

このコマンドの構文は次のとおりです。

chaingroup *group_name*

group_name 引数は、既存の証明書チェーン グループの名前です（第 2 章「証明書およびキーの管理」の「チェーン グループの作成」の項を参照）。チェーン グループの最大サイズは 11 KB です。最大 64 文字の英数字からなる文字列を引用符で囲まずに入力します。

（注）　チェーン グループ証明書を変更した場合、その変更は、**chaingroup** コマンドを使用して SSL プロキシ サービスに関連付けられたチェーン グループを再指定した後に有効になります。

たとえば、証明書チェーン グループ MYCHAINGROUP を指定するには、次のように入力します。

```
host1/Admin(config)# ssl-proxy service PSERVICE_SERVER
host1/Admin(config-ssl-proxy)# chaingroup MYCHAINGROUP
```

SSL プロキシ サービスから証明書チェーン グループを削除するには、次のように入力します。

```
host1/Admin(config-ssl-proxy)# no chaingroup MYCHAINGROUP
```

クライアント認証のイネーブル化

通常の SSL ハンドシェイクでは、サーバが自身の証明書をクライアントに送信します。クライアントはその証明書からサーバのアイデンティティを確認します。ただし、クライアントは、クライアント自身の識別情報をサーバに送信しません。クライアント認証機能を ACE で有効にすると、ACE はクライアントがサーバに証明書を送信することを要求します。次に、サーバは証明書の下記の情報について検証します。

- 認定されている CA が証明書を発行した。
- 証明書の有効期間が満了していない。
- 証明書の署名が有効である。
- CA が証明書を取り消していない。

SSL プロキシ コンフィギュレーション モードで **authgroup** コマンド使用して、ACE が SSL ハンドシェイク中に使用する証明書認証グループを指定し、この SSL プロキシ サービスでクライアント認証を有効にできます。ACE には、SSL プロキシ サービスに指定した証明書付きのグループに設定された証明書が含まれます（「証明書の指定」の項を参照）。

このコマンドの構文は次のとおりです。

authgroup *group_name*

group_name 引数は、既存の証明書認証グループ名です（第 2 章「証明書およびキーの管理」の「認証のための証明書グループの設定」の項を参照）。最大 64 文字の英数字からなる文字列を引用符で囲まずに入力します。

（注） ACE のクライアント認証を有効にした場合、ACE のパフォーマンスが大幅に低下し、アクセスの失敗が発生する場合があります。さらに、CRL 取得を設定したり（「クライアント認証中の CRL の使用」の項を参照）、証明書の取り消しエラーが原因で VIP トラフィックが 100 ～ 200 TPS を超えると、遅延が発生する可能性があります。

（注） authgroup を変更した場合、その変更は **authgroup** コマンドを使用して SSL プロキシ サービスに関連付けられた authgroup を再指定した後にのみ有効になります。

たとえば、証明書認証グループ AUTH-CERT1 を指定するには、次のように入力します。

```
host1/Admin(config-ssl-proxy)# authgroup AUTH-CERT1
```

SSL プロキシ サービスから証明書認証グループを削除するには、次のように入力します。

```
host1/Admin(config-ssl-proxy)# no authgroup AUTH-CERT1
```

クライアント認証中の CRL の使用

デフォルトでは、ACE は、クライアント認証中に証明書失効リスト（CRL）を使用しません。ACE は、HTTP または LDAP を介した CRL のダウンロードをサポートします。次のいずれかの方法で、CRL を使用するように SSL プロキシ サービスを設定できます。

- ACE は、サービス用の各クライアント証明書をスキャンして、認証拡張内の CRL をポイントする CRL 分散ポイント（CDP）が含まれているかどうかを判断し、その後で、CDP が有効な場合はその場所からその CRL を取得することができます。CDP に http:// または ldap:// ベースの URL がある場合は、その URL を使用して、CRL を ACE にダウンロードします。
- ACE が CRL を取得する CRL のダウンロード場所を手動で設定できます（「CRL のダウンロード場所の設定」の項を参照）。

（注）　デフォルトでは、使用されている CRL がその更新日を過ぎた場合、ACE はクライアント証明書を拒否しません。CRL が期限切れになった場合に証明書を拒否するように ACE を設定するには、**expired-crl reject** コマンドを使用します。詳細については、「期限切れの CRL クライアント証明書の拒否」のセクションを参照してください。

ベスト エフォート CRL が設定されている場合に CRL のダウンロードを試みると、以下のようになります。

- ACE は、証明書内、または ACE で設定されている最初の 4 つの CDP のみを考慮します。証明書から取得された CDP の場合、ACE は、CRL のダウンロード用に、有効で完全な CDP だけを考慮します。1 つの CDP で CRL が正常にダウンロードされた場合、ACE は、CRL のダウンロード用に後続の CDP は考慮しません。
- 最初の 4 つの CDP のいずれも有効ではなく CRL のダウンロードを続行できない場合、パラメータ マップ SSL コンフィギュレーション モードで **authentication-failure ignore** コマンドを設定していない限り、ACE はその証明書を失効と判断します。
- ACE が 4 つの有効な CDP を試したあとで CRL のダウンロードに失敗した場合、パラメータ マップ SSL コンフィギュレーション モードで **authentication-failure ignore** コマンドを設定していない限り、ACE は開始した SSL 接続を中止します。

- 提示された証明書内に CDP エラーを検出した場合や、CRL のダウンロード中に発生したエラーを検出した場合、ACE は、パラメータ マップ SSL コンフィギュレーション モードで **cdp-errors ignore** コマンドが設定されていない限り、SSL 接続を拒否します。
- ACE は、形式が正しくない CDP をスキップし、後続の CDP を処理します。形式が正しくない CDP を含む CDP エラー統計情報を表示するには、**show crypto cdp-errors** コマンドを使用します。

詳細な CRL ダウンロード統計情報については、第 6 章「SSL 情報および統計情報の表示」の「CRL 情報の表示」の項を参照してください。

SSL プロキシ コンフィギュレーション モードで **crl** コマンドを使用して、クライアント認証にどの CRL 情報を使用するかを判断できます。このコマンドの構文は次のとおりです。

crl {*crl_name* | **best-effort**}

引数とキーワードは次のとおりです。

- *crl_name*：コンフィギュレーション モードで **crypto crl** コマンドを使用して、CRL をダウンロードするときに CRL に割り当てた名前です。「CRL のダウンロード場所の設定」の項を参照してください。
- **best-effort**：ACE が各クライアント証明書をスキャンして、認証拡張内にある CRL をポイントする CDP が含まれているかどうかを判断し、その CDP が有効な場合は、その場所から CRL を取得するように指定します。

たとえば、SSL プロキシ サービスのクライアント認証用の CRL1 CRL を有効にするには、次のように入力します。

```
host1/Admin(config-ssl-proxy)# crl CRL1
```

CRL 情報のクライアント証明書をスキャンするには、次のように入力します。

```
host1/Admin(config-ssl-proxy)# crl best-effort
```

クライアント認証中に、ダウンロードされた CRL の使用を無効にするには、次のように入力します。

```
host1/Admin(config-ssl-proxy)# no crl CRL1
```

クライアント認証中に、CRL 情報のためにクライアント証明書の使用を無効にするには、次のように入力します。

```
host1/Admin(config-ssl-proxy)# no crl best-effort
```

CRL のダウンロード場所の設定

ACE がクライアント認証用に SSL プロキシ サービスに CRL をダウンロードするときに使用する場所を設定できます。サービスがポリシー マップで設定されていない場合や、ポリシー マップがアクティブでない場合、ACE は CRL をダウンロードしません。ACE は、次の条件下で CRL をダウンロードします。

- 最初に CRL を設定し、アクティブ レイヤ 4 ポリシー マップにアクションとして適用する場合（「ポリシー マップと SSL プロキシ サーバ サービスの関連付け」の項を参照）。
- ACE を再ロードする場合。
- CRL 自体で指定される NextUpdate CRL に到達すると、ACE はこの情報を読み取り、それに基づいて CRL を更新します。ACE は、次のクライアント認証要求時に、更新された CRL をダウンロードします。

コンテキストごとに最大 8 つの CRL を設定できます。CRL を設定したら、クライアント認証用に SSL プロキシ サービスに割り当てます（「クライアント認証中の CRL の使用」の項を参照）。

ACE は、ユーザが設定したドメイン ネーム システム（DNS）クライアントを使用して、CRL 内のホスト名を IP アドレスに変換します。DNS クライアントの設定の詳細については、「DNS クライアントの設定」の項を参照してください。

ダウンロードされた CRL を設定するには、コンフィギュレーション モードで **crypto crl** コマンドを使用します。このコマンドの構文は次のとおりです。

crypto crl *crl_name url*

引数は次のとおりです。

- *crl_name* : CRL に割り当てる名前です。最大 64 文字の英数字からなる文字列を引用符で囲まずに入力します。
- *url* : ACE が CRL を取得する URL です。CRL ファイル名が含まれる URL のフル パスを、最大 255 文字の英数字からなる文字列として引用符で囲まずに入力します。HTTP および LDAP URL の両方がサポートされます。http:// プレフィックスまたは ldap:// プレフィックスで始まる URL を指定します。

  ldap:/// プレフィックスは、サーバ証明書の CDP 部分で有効な LDAP CRL リンクとは見なされません。LDAP URL の有効な形式は次のとおりです。

  - ldap://10.10.10.1:389/dc=cisco,dc=com?o=bu?certificateRevocationList
  - ldap://10.10.10.1/dc=cisco,dc=com?o=bu?certificateRevocationList

- ldap://ldapsrv.cisco.com/dc=cisco,dc=com?o=bu?certificateRevocationList
- ldap://ldapsrv.cisco.com:389/dc=cisco,dc=com?o=bu?certificateRevocationList

URL の一部として疑問符（?）文字を使用するには、入力する前に Ctrl+V キーを押します。押さないと、ACE は help コマンドとして疑問符を解釈します。

（注）　ldap:// リンク内のホスト名は、DNS 設定を使用して解決されます。LDAP では、TCP ポート 389 が使用されます。CRL を発行する LDAP サーバが標準ではない LDAP ポートでリッスンする場合は、標準ではない LDAP ポートを CDP で設定する必要があります。

たとえば、CRL1 という名前を付ける CRL を http://crl.verisign.com/class1.crl から設定するには、次のように入力します。

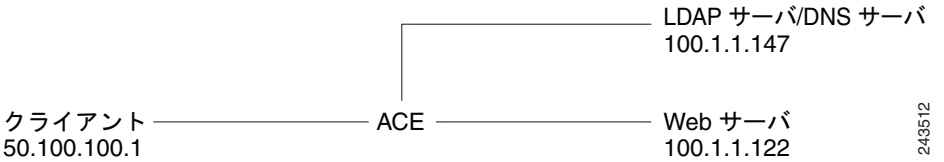
```
host1/Admin(config)# crypto crl CRL1
http://crl.verisign.com/class1.crl
```

CRL を削除するには、次のように入力します。

```
host1/Admin(config)# no crypto crl CRL1
```

たとえば、図 3-4 に、クライアント認証の LDAP による CRL のダウンロードの設定例を示します。

図 3-4　LDAP プロトコルによる CRL のダウンロード



次に、クライアント証明書に署名したルート証明書を持つ認証グループ設定例を示します。

```
crypto authgroup root_ca_pool
    cert root-cert-2.cer
```

次に、ldap:// ベースの CDP URL の設定例を示します。

```
crypto crl win2003crl1
ldap://windows2003-srv.win2003.cisco.com/CN=root-ca(2),CN=windows2003-
srv,CN=CDP,CN=Public%20Key%20Services,CN=Services,CN=Configuration,DC=
win2003,DC=cisco,DC=com?certificateRevocationList?base?objectClass=cRL
DistributionPoint

access-list capture-acl line 8 extended permit tcp any any
access-list permit-http line 8 extended permit tcp any any eq https
```

次に、CRL のダウンロード中に ACE が ldap:// URL のホスト名を正常に解決するための DNS の設定例を示します。

```
ip domain-lookup
ip domain-name win2003.cisco.com
ip name-server 10.1.1.147

rserver host SERVER1
    ip address 10.1.1.122
    inservice

ssl-proxy service SSL_PSERVICE_SERVER
    key MYKEY.PEM
    cert MYSCERT.PEM
    authgroup root_ca_pool
    crl win2003crl1

serverfarm host SFARM1
    rserver SERVER1 80
        inservice

class-map match-any L4_SSL-TERM_CLASS
    3 match virtual-address 192.168.1.100 tcp eq https
class-map type http loadbalance match-all URLCLASS1
    2 match http url .*

policy-map type loadbalance first-match L7_SSL-TERM_POLICY
    class URLCLASS1
        serverfarm SFARM1

policy-map multi-match L4_SSL-VIP_POLICY
    class L4_SSL-TERM_CLASS
        loadbalance vip inservice
        loadbalance policy L7_SSL-TERM_POLICY
        loadbalance vip icmp-reply
        ssl-proxy server SSL_PSERVICE_SERVER

interface vlan 50
```

```
    ip address 10.1.1.138 255.255.0.0
    no shutdown

interface vlan 200
    ip address 192.168.1.254 255.255.0.0
    access-group input permit-http
    service-policy input L4_SSL-VIP_POLICY
    no shutdown
```

グローバルな CRL パラメータの設定

コンフィギュレーション コマンド モードで **crypto crlparams** コマンドを使用して、証明書失効リスト（CRL）の署名検証を設定して、信頼できる認証局からのものであるかを判定したり、CRL ダウンロードのタイムアウトを設定して、ACE がサーバから CRL データを取得するのを待機する最大時間を指定できます。このコマンドの構文は次のとおりです。

crypto crlparams *crl_name* **cacert** *ca_cert_filename* | **timeout** *number*

次のキーワードと引数があります。

- *crl_name*：既存の CRL の名前です。
- **cacert** *ca_cert_filename*：署名確認に使用する CA 証明書ファイルの名前です。
- **timeout** *number*：サーバとの接続を閉じる前に ACE が CRL データを待機する秒数を指定します。スタティック CRL の場合、2 ～ 300 の整数を入力します。ベスト エフォート CRL の場合、タイムアウトは 60 秒で、ユーザによる設定はできません。ACE が、タイムアウト制限内に CRL データ全体を受信しなかった場合、ACE はサーバとのソケット接続を閉じます。スタティック CRL の場合、設定からスタティック CRL を削除することで、CRL データのダウンロードを中止できます。

たとえば、CRL で署名確認を設定するには、次のように入力します。

```
host1/Admin(config)# crypto crlparams CRL1 cacert MYCERT.PEM
```

CRL から署名確認を削除するには、次のように入力します。

```
host1/Admin(config)# no crypto crlparams CRL1
```

たとえば、CRL1 に 200 秒の CRL ダウンロードのタイムアウトを設定するには、次のコマンドを入力します。

host1/Admin(config)# **crypto crl-params CRL1 timeout 200**

CRL データのダウンロードのタイムアウトが期限切れになり、ダウンロードが中止されると、ACE は、次のようにイベントを記録する syslog を生成します。

```
%ACE-6-253008: CRL crl_name could not be retrieved, reason: crl data
dnld timeout error
```

crl_name 変数は、CRL のダウンロードのタイムアウトが期限切れになったためにダウンロードが中止された既存の CRL の名前を示します。

OCSP の設定

Online Certificate Status Protocol（OCSP）は、証明書の失効ステータスを識別するためにクライアントとサーバ間で使用される方法です。OCSP サーバ（OCSP レスポンダ）は、異なる認証局（CA）からの証明書情報を維持または取得し、この情報を要求に応じてクライアントに提供します。OCSP の利点は、証明書に関する最新情報を提供し、大量のメモリが必要になる場合がある証明書失効リスト（CRL）をダウンロードして保存する必要性を排除することです。

ACE は OCSP クライアントとして機能し、HTTP を介して OCSP サーバに要求を送信します。必要に応じて、要求は、指定された CA 証明書の公開キーで署名される場合があります。要求には、次の主要な情報が含まれます。

- 証明書の ID
- 証明書発行元の識別名のハッシュ
- 証明書発行元の公開キーのハッシュ

サーバは、証明書が失効している、失効していない、またはステータス不明のいずれかを示す証明書失効情報で応答します。

ACE は、CRL を使用する代わりとして OCSP を使用します。この OCSP の実装は、RFC 2560 に準拠していますが、RFC のすべてのオプション機能は含まれません。ACE は次の RFC 2560 のオプション機能をサポートします。

- HTTP は唯一の OCSP トランスポート メカニズムです。

- 署名者証明書には、id-kp-OCSPsigning extendedKeyUsage 拡張（RFC のセクション 4.2.2.2）が含まれる場合と、含まれない場合があります。これは、証明書のその拡張を要求すると、要求への署名に適した証明書の選択が非常に限定的になる可能性があるためです。
- オプションの要求の署名者証明書は、ACE が OCSP サーバに送る要求には含まれません。
- ACE は、承認された OCSP レスポンダ証明書の認証または失効チェックを実行しません。
- オプションでサポートされるナンスを除き、RFC のセクション 4.4 で定義されているその他の拡張はサポートされません。

証明書の複数の OCSP サーバ アクセスの場所の処理は、証明書内の複数の証明書の CRL 分散ポイント（CDP）に対する手順と同じです。最大 4 つの Authority Infomation Access（AIA）フィールドが証明書から取得されます。

同じ SSL プロキシ内に CRL サーバおよび OCSP サーバの両方を設定できます。この場合、ACE が失効チェックを実行する順序を設定できます。デフォルトでは、ACE は最初に OCSP サーバから、次に CRL から失効ステータスを取得します。

（注）　CRL および OCSP サーバ情報の共存と、これらのトラバーサルによって、ハンドシェイクの完了に必要な時間が増加し、ACE 全体のパフォーマンスが低下することがあります。

ここでは、次の内容について説明します。

- 注意事項および制約事項
- OCSP サーバの設定
- SSL プロキシ サービスへの OCSP サーバの適用
- 失効チェックのプライオリティの設定

注意事項および制約事項

OCSP の設定には次の注意事項と制限があります。

- ACE に設定できる OCSP サーバの最大数は 64 です。
- SSL プロキシ サービスに設定できる OCSP サーバの最大数は 10 です。

- ACE は、スタティック OCSP サーバとベスト エフォート OCSP サーバの両方を組み合わせて、最大で 64 の OCSP サーバ接続を処理できます。
- 同じプロキシ リスト内にベスト エフォート OCSP サーバとベスト エフォート CRL を設定した場合、ACE は、リソースを節約するために、最大で 4 つの AIA と 4 つの CDP を取得します。
- 同じ SSL プロキシ サービスに OCSP サーバと CRL を設定すると、クライアント認証に遅延が生じる可能性があります。
- ACE は、署名者証明書に応答して、認証および失効チェックを実行しません。

OCSP サーバの設定

ACE が失効チェックに使用する OCSP サーバを設定できます。ACE に設定できる OCSP サーバの最大数は 64 です。64 の OCSP サーバすべてを 1 つのコンテキストに設定したり、複数のコンテキストに分散させることができます。OCSP サーバ接続の最大数も 64 です。

SSL プロキシ サービスに適用する予定の OCSP サーバを設定するには、コンフィギュレーション モードで **crypto ocspserver** コマンドを使用します。このコマンドの構文は次のとおりです。

crypto ocspserver *ocsp_server_name url* [**conninactivitytout** *timeout*] [**nonce enable** | **disable**] [**reqsigncert** *signer_cert_filename* {**reqsignkey** *signer_key_filename*}] [**respsigncert** *response_signer_cert*]

キーワード、オプション、および引数は次のとおりです。

- *ocsp_server_name* : OCSP サーバの ID です。SSL プロキシ サービスに OCSP サーバを適用するには、この名前を使用します。最大 64 文字の英数字からなる文字列を引用符で囲まずに入力します。スペースは使用しません。
- *url* : http://ocsphost.com:port_id/ の形式の HTTP URL です。ポート ID はオプションです。ポートを指定しない場合、デフォルト値の 2560 が使用されます。IPv4 または IPv6 ベースの URL を指定できます。スペースを含まず引用符なしの英数字を入力します（最大 255 文字）。
- **conninactivitytout** *timeout* :（任意）TCP 接続無活動タイムアウト。秒単位。2 ～ 3600 の整数を入力します。デフォルトは 300 秒です。

- **nonce enable | disable**：（任意）ナンスの使用を有効または無効にします。デフォルトでは、**nonce** は無効化されています。ナンスは、OCSP 要求と応答をバインドするために使用される一意の文字列です。ナンスが有効な場合、ACE は OCSP サーバへ送信する要求に固有の文字列を含めます。サーバは、応答を確認するために ACE への応答に文字列を含める必要があります。
- **reqsigncert** *signer_cert_filename*：（任意）OCSP サーバへの発信要求に署名するための署名者の証明書ファイル名です。デフォルトでは、要求は署名されていません。
- **reqsignkey** *signer_key_filename*：（任意）OCSP サーバへの発信要求に署名するための署名者の秘密キーのファイル名です。デフォルトでは、要求は署名されていません。**reqsigncert** オプションを入力した場合、**reqsignkey** オプションを入力する必要があります。
- **respsigncert** *response_signer_cert*：（任意）OCSP サーバの応答の署名を確認するための証明書です。デフォルトでは、OCSP サーバからの応答の署名は確認されません。

たとえば、SSL 証明書の失効ステータスをチェックするために ACE によって使用される OCSP サーバを設定するには、次のコマンドを入力します。

```
host1/Admin(config)# crypto ocspserver OCSP_SSERVER1
http://10.10.10.10/ nonce enable conninactivitytout 60
```

設定から OCSP サーバを削除するには、次のコマンドを入力します。

```
host1/Admin(config)# no crypto ocspserver OCSP_SSERVER1
```

SSL プロキシ サービスへの OCSP サーバの適用

SSL プロキシ サービスに最大 10 の OCSP サーバを適用できます。SSL プロキシ サービスに OCSP サーバを適用するには、SSL プロキシ コンフィギュレーション モードで **ocspserver** コマンドを使用します。このコマンドの構文は次のとおりです。

ocspserver *ocsp_server_name* | **best-effort**

引数およびオプションは、次のとおりです。

- *ocsp_server_name*：この SSL プロキシ サービスに適用する OCSP サーバの ID です。既存の OCSP サーバの名前を、最大 64 文字の英数字としてスペースを含めずに入力します。
- **best-effort**：ACE が OCSP サーバから証明書の失効情報をベスト エフォート方式で取得するように指定します。このキーワードを設定するときに、ACE はクライアント証明書から OCSP サーバ情報（最大で 4 つの OCSP サーバ情報要素）を取得します。このキーワードは、受け取るクライアントまたはサーバの証明書の AuthorityInfoAccess（AIA）の拡張を検索するように ACE に強制します。この拡張の形式は次のとおりです。

  ```
  authorityInfoAccess = OCSP;URI:
  http://test1.ocsp.ve/,OCSP;URI:http://test2.ocsp.ve/
  ```

  best-effort が設定されている場合にこの拡張が証明書に含まれていない場合、証明書は失効していると見なされます。

たとえば、PSERVICE_SERVER SSL プロキシ サービスに OCSP_SERVER1 OCSP サーバを適用するには、次のコマンドを入力します。

```
host1/Admin(config)# ssl-proxy service PSERVICE_SERVER
host1/Admin(config-ssl-proxy)# ocspserver OCSP_SERVER1
```

SSL プロキシ サービスにベスト エフォート OCSP サーバを適用するには、次のコマンドを入力します。

```
host1/Admin(config)# ssl-proxy service PSERVICE_SERVER
host1/Admin(config-ssl-proxy)# ocspserver best-effort
```

SSL プロキシ サービスから OCSP サーバを削除するには、次のコマンドを入力します。

```
host1/Admin(config-ssl-proxy)# no ocspserver OCSP_SERVER1
```

失効チェックのプライオリティの設定

同じ SSL プロキシ サービスに OCSP と CRL の両方を設定する場合、ACE が SSL 証明書の失効ステータスをチェックするためにこれら 2 つのリソースを使用する順序を制御できます。失効チェックの順序を設定するには、SSL プロキシ コンフィギュレーション モードで **revcheckprio** コマンドを使用します。OCSP または CRL の（両方の方法ではなく）いずれかが SSL プロキシ サービスに適用されている場合、このコマンドは設定可能ではありません。このコマンドの構文は次のとおりです。

revcheckprio crl-ocsp | **ocsp-crl**

キーワードは次のとおりです。

- **crl-ocsp**：クライアントの SSL 証明書の失効ステータスを判定するために、最初に CRL を使用し、次に OCSP を使用するように ACE に指示します。
- **ocsp-crl**：（デフォルト）クライアントの SSL 証明書の失効ステータスを判定するために、最初に OCSP を使用し、次に CRL を使用するように ACE に指示します。

このコマンドが設定されていない場合、ACE は、最初に OSCP を使用し、次に CRL を使用して、SSL 証明書の失効ステータスを判定します。両方の方法で証明書のステータスが判定できない場合、証明書は無効と見なされます。

（注）　デフォルトの失効チェックの優先順位（**revcheckprio ocsp-crl**）は、該当する優先順位が設定されていても、**show running-config** コマンドの出力には表示されません。

たとえば、最初に CRL を使用し、次に OCSP を使用して失効ステータスをチェックするように ACE を設定するには、次のコマンドを入力します。

```
host1/Admin(config)# ssl-proxy service PSERVICE_SERVER
host1/Admin(config-ssl-proxy)# revcheckprio crl-ocsp
```

OCSP サーバの証明書の失効を最初に確認し次に CRL を確認する、ACE のデフォルトの動作をリセットするには、次のコマンドを入力します。

```
host1/Admin(config-ssl-proxy)# no revcheckprio crl-ocsp
```

DNS クライアントの設定

クライアント認証機能により、CRL のホスト名にホスト名から IP アドレスへの変換を提供する DNS サーバと通信するために、ドメイン ネーム システム（DNS）クライアントを ACE に設定できます。クライアント認証の詳細については、「クライアント認証中の CRL の使用」の項を参照してください。

ACE の DNS クライアントを設定する前に、1 つ以上の DNS ネーム サーバが正しく設定され、到達可能であることを確認します。設定されていないと、DNS クライアントからの変換要求（ドメイン ルックアップ）は破棄されます。ネー

ム サーバは、最大 3 台まで設定できます。ACE は、設定されたネーム サーバに対して順番に、変換が成功するまでホスト名の解決を試行します。変換が失敗した場合、ACE はエラーを報告します。

非修飾ホスト名（ドメイン名が含まれないホスト名）の場合は、デフォルト ドメイン名を設定するか、ACE が次のタスクを行うために使用できるドメイン名のリストを設定できます。

- ホスト名を入力する
- DNS サーバと、ホスト名から IP アドレスへの解決を試みる

DNS クライアントの設定を表示するには、**show running-config** コマンドを使用します。

ここでは、次の内容について説明します。

ドメイン ルックアップの有効化

ACE による DNS サーバとのドメイン ルックアップ（ホストからアドレスへの変換）の実行を有効にするには、コンフィギュレーション モードで **ip domain-lookup** コマンドを使用します。デフォルトでは、このコマンドはディセーブルです。このコマンドの構文は次のとおりです。

ip domain-lookup

たとえば、ドメイン ルックアップを有効にするには、次のように入力します。

```
host1/Admin(config)# ip domain-lookup
```

ドメイン ルックアップの状態をデフォルト値（無効）に戻すには、次のように入力します。

```
host1/Admin(config)# no ip domain-lookup
```

デフォルトのドメイン名の設定

DNS クライアント機能により、ACE が非修飾ホスト名を完成するために使用するデフォルトのドメイン名を設定することができます。未修飾のホスト名は、ドメイン名が含まれない名前です（ドットのない名前）。ドメイン ルックアップが有効で、デフォルトのドメイン名が設定されている場合、ACE は、非修飾ホスト名にドット（.）と設定されたデフォルト ドメイン名を付加し、ドメイン ルックアップを試みます。

デフォルトのドメイン名を設定するには、コンフィギュレーション モードで **ip domain-name** コマンドを使用します。このコマンドの構文は次のとおりです。

ip domain-name *name*

name 引数は、引用符とスペースが含まれないテキスト文字列で、最大 85 文字の英数字です。

たとえば、cisco.com のデフォルト ドメイン名を指定するには、次のように入力します。

```
host1/Admin(config)# ip domain-name cisco.com
```

上記の例で、DNS ネーム サーバを使用して ACE がホスト名の IP アドレスへの解決を試行する前に、ACE は CRL の任意の非修飾ホスト名に .cisco.com を追加します。

設定からデフォルト ドメインを削除するには、次のように入力します。

```
host1/Admin(config)# no ip domain-name cisco.com
```

ドメイン名の検索リストの設定

単一のデフォルト ドメイン名を設定する代わりに、非修飾ホスト名を完成するために ACE が使用するドメイン名の検索リストを設定できます。ドメイン名のリストには、最大で 3 つのドメイン名を含めることができます。ドメイン名のリストおよびデフォルト ドメイン名の両方を設定する場合、ACE は単一のデフォルト名ではなく、ドメイン名のリストだけを使用します。ドメイン名の検索を有効にして、ドメイン名のリストを設定すると、ACE は IP アドレスに 1 つのドメイン名が解決されるまで各ドメイン名を順番に使用します。

ドメイン名の検索のリストを設定するには、**ip domain-list** コマンドを使用します。このコマンドの構文は次のとおりです。

ip domain-list *name*

name 引数は、引用符とスペースが含まれないテキスト文字列で、最大 85 文字の英数字です。

たとえば、ドメイン名のリストを設定するには、次のように入力します。

```
host1/Admin(config)# ip domain-list cisco.com
host1/Admin(config)# ip domain-list foo.com
host1/Admin(config)# ip domain-list xyz.com
```

リストからドメイン名を削除するには、次のように入力します。

```
host1/Admin(config)# no ip domain-list xyz.com
```

ドメイン ネーム サーバの設定

ホスト名を IP アドレスに解決するには、ACE に 1 つ以上（最大 3 つ）の既存の DNS ネーム サーバを設定します。サーバが到達不能であることを確認するために、設定する前に各ネーム サーバの IP アドレスに ping を送信します。

ネーム サーバを設定するには、コンフィギュレーション モードで **ip name-server** コマンドを使用します。このコマンドの構文は次のとおりです。

ip name-server *ip_address*

ip_address 引数は、ネーム サーバの IP アドレスのドット付き 10 進表記です（たとえば、192.168.12.15）。1 つのコマンドラインに最大 3 個のネーム サーバの IP アドレスを入力できます。

たとえば、DNS クライアント機能に対して 3 つのネーム サーバを設定するには、次のように入力します。

```
host1/Admin(config)# ip name-server 192.168.12.15 192.168.12.16
192.168.12.17
```

リストからネーム サーバを削除するには、次のように入力します。

```
host1/Admin(config)# no ip name-server 192.168.12.15
```

SSL URL 書き換えと HTTP ヘッダー挿入の設定

クライアントが SSL 終了設定で暗号化トラフィックを ACE に送信する場合、ACE は SSL トラフィックを終了してサーバにクリア テキストを送信します。サーバは、クライアントと ACE の間で暗号化されたトラフィックがフローしていることを認識していません。レイヤ 7 HTTP ロードバランシング ポリシー マップに関連付けられているアクション リストを使用して、次のタスクを実行するように ACE に指示できます。

- SSL URL 書き換え：ACE は、クライアントに応答を送信する前に、サーバからの Location 応答ヘッダーのリダイレクト URL を http:// から https:// に変更します。
- SSL HTTP ヘッダー挿入：ACE は、接続を介して受け取る HTTP 要求を HTTP ヘッダーを挿入することによって、次の SSL セッション情報をサーバに提供します。
  - セッション パラメータ：ACE およびクライアントが SSL ハンドシェイク時にネゴシエートする SSL セッション パラメータ。
  - サーバ認証のフィールド：ACE に存在する SSL サーバ証明書に関する情報。
  - クライアント認証のフィールド：クライアント認証を実行するように ACE を設定した場合に、ACE がクライアントから取得する SSL クライアント認証に関する情報。

次の各項では、必要な指示を ACE に与えるアクション リストを使用して、SSL URL 書き換えと HTTP ヘッダー挿入のために ACE を設定する方法について説明します。

ここでは、次の内容について説明します。

アクション リストの設定

SSL URL 書き換えまたは HTTP ヘッダー挿入を設定するには、まず、新しいアクション リストを作成するか、タイプ modify の既存のアクション リストを使用する必要があります。

注意　SSL HTTP ヘッダー挿入に設定するアクション リストは class-default クラス マップだけに関連付ける必要があります。したがって、アクション リストが class-default クラス マップではないクラス マップに現在関連付けられている場合、SSL HTTP ヘッダー挿入の既存のアクション リストを設定できません。

アクション リストは、ACE で実行する関連アクションの名前付きグループです。たとえば、アクション リストを作成するには、コンフィギュレーション モードで次のコマンドを入力します。

```
host1/Admin(config)# action-list type modify http SSL_ACTLIST
host1/Admin(config-actlist-modify)#
```

action-list type modify http コマンドにより、アクション リスト変更コンフィギュレーション モードが開始され、次の機能のパラメータを定義できます。

- SSL URL 書き換え（「SSL URL 書き換えの設定」の項を参照）
- SSL セッション パラメータの挿入（「SSL セッション パラメータの HTTP ヘッダー挿入の設定」の項を参照）
- SSL サーバ証明書フィールドの挿入（「SSL サーバ証明書情報の HTTP ヘッダー挿入の設定」の項を参照）
- SSL クライアント証明書フィールドの挿入（「SSL クライアント証明書情報の HTTP ヘッダー挿入の設定」の項を参照）

アクション リストの詳細については、『*Server Load-Balancing Guide, Cisco ACE Application Control Engine*』を参照してください。

SSL URL 書き換えの設定

サーバは、クライアントと ACE 間の暗号化トラフィックには気づかないため、HTTP リダイレクト応答の Location ヘッダー（301：Moved Permanently または 302：Found）にある URL を https://www.cisco.com ではなく

http://www.cisco.com 形式でクライアントに返すことがあります。その場合、クライアントは、元の要求がセキュリティのある URL だったにもかかわらず、暗号化されていない無防備な URL へ要求を送信します。クライアント接続が HTTP に変更されているため、空白のテキスト接続を使用して要求データをサーバから取得できない場合があります。

ACE は SSL URL を書き換えることで、この問題を解決しています。これにより、クライアントに応答を送信する前に、サーバからの Location 応答ヘッダーにあるリダイレクト URL を http:// から https:// に変更できます。URL を書き換えることで、無防備な HTTP へのリダイレクトを回避できます。Web サーバへのすべてのクライアント接続が SSL になるため、セキュリティが確保された HTTPS コンテンツをクライアントへ返すことができます。ACE は、URL に書き換えが必要かどうかの判断に正規表現を一致条件として使用します。Location 応答ヘッダーが指定の正規表現と一致した場合、ACE は URL を書き換えます。さらに、ACE はコマンドを提供して SSL と空白のポート番号の追加または変更を実行します。

アクション リスト変更コンフィギュレーション モードで **ssl url rewrite** コマンドを使用して、SSL URL、SSL ポート、および書き換え用の空白のポートを定義できます。このコマンドの構文は次のとおりです。

ssl url rewrite location *expression* [**sslport** *number1*] [**clearport** *number2*]

引数、キーワード、およびオプションは次のとおりです。

- **location** *expression*：URL 正規表現の一致条件に基づき Location 応答ヘッダー内の URL を書き換えます。Location ヘッダー内の URL が、指定した URL 正規表現文字列に一致した場合、ACE は URL を http:// から https:// に書き換えます。また、同時にポート番号も書き換えます。スペースを含まず引用符なしの英数字を入力します（最大 255 文字）。または、ストリング全体を引用符（"）で囲むことによって、スペースが含まれるテキスト ストリングを入力することもできます。

  入力した regex の場所は、ポートやパスが指定されていない元の URL（例：www\.cisco\.com）と同じにする必要があります。ポートに対応付けるために、この項で後述する **sslport** および **clearport** キーワードを使用します。パスを一致させる必要がある場合、HTTP ヘッダーの書き換え機能を使用して文字列を書き換えます。HTTP ヘッダーの書き換え機能の詳細については、『*Server Load-Balancing Guide, Cisco ACE Application Control Engine*』を参照してください。

  ACE は、データ ストリングの照合に正規表現を使用することをサポートします。正規表現に使用できる、サポート対象文字の一覧については、表 3-4 を参照してください。

（注） データ ストリングの照合時の正規表現では、ピリオド（.）および疑問符（?）は文字通りの意味を持ちません。これらの記号を照合する場合は、角カッコ（[ ]）を使用します（たとえば、www.xyz.com の代わりに www[.]xyz[.]com を入力します）。ドット（.）または疑問符（?）のエスケープとしてバックスラッシュ（\）を使用することもできます。

- **sslport** *number1*：（任意）サーバ リダイレクト応答をクライアントに送信する前に、ACE が空白のポート番号を変換する SSL ポート番号を指定します。1 ～ 65535 の整数を入力します。デフォルトは 443 です。
- **clearport** *number2*：（任意）サーバ リダイレクト応答をクライアントに送信する前に、ACE が SSL ポート番号を変換する空白のポート番号を指定します。1 ～ 65535 の整数を入力します。デフォルトは 80 です。

たとえば、デフォルトの SSL ポート 443 とクリア ポート 8080 を使用して、www.cisco.com または www.cisco.net の URL に対する SSL URL 書き換えを指定するには、次のように入力します。

```
host1/Admin(config-actlist-modify)# ssl url rewrite location
www\.cisco\..* sslport 443 clearport 8080
```

上記の例で、ACE は次のタスクの実行を試みます。

- すべての HTTP リダイレクトを http://www.cisco.com:8080 または http://www.cisco.net:8080 に一致させる
- HTTP リダイレクトを https://www.cisco.com:443 または https://www.cisco.net:443 として書き換える
- HTTP リダイレクトをクライアントに転送する

ssl url rewrite コマンドを入力したら、レイヤ 3 およびレイヤ 4 ポリシー マップにアクション リストを関連付けます。「レイヤ 7 HTTP ロードバランシング ポリシー マップとアクション リストの関連付け」の項を参照してください。

表 3-4　文字列表現の一致に使用する特殊文字

表記法	説明
.	任意の 1 文字
.*	0 個以上の任意の文字
\.	ピリオド（エスケープ）

表 3-4　文字列表現の一致に使用する特殊文字 （続き）

表記法	説明
[*charset*]	範囲内の任意の 1 文字に一致します。
[^charset]	範囲内の文字はどれも一致しません。その他の文字はすべて、その文字のままです。
()	表現のグループ化
(expr1 | expr2)	表現の論理和
(expr)*	0 個以上の表現
(expr)+	1 個以上の表現
expr{m,n}	表現を *m* ～ *n* 回繰り返します。*m* および *n* の範囲は 1 ～ 255 です。
expr{m}	表現が正確に *m* 回だけ繰り返す場合に一致します。*m* の範囲は 1 ～ 255 です。
expr{m,}	表現が *m* 回以上繰り返す場合に一致します。*m* の範囲は 1 ～ 255 です。
\a	アラート（ASCII 7）
\b	バックスペース（ASCII 8）
\f	用紙送り（ASCII 12）
\n	改行（ASCII 10）
\r	復帰（ASCII 13）
\t	タブ（ASCII 9）
\v	垂直タブ（ASCII 11）
\0	ヌル（ASCII 0）
\\	バックスラッシュ
\x##	2 桁の 16 進表記で指定された任意の ASCII 文字

SSL セッション パラメータの HTTP ヘッダー挿入の設定

ACE とクライアントが SSL ハンドシェイク時にネゴシエートする SSL セッション パラメータの情報（情報または SSL セッション ID の暗号化に使用される暗号スイートなど）をサーバに提供するように、ACE に指示できます。この SSL セッション情報をサーバに転送するために、ACE はネゴシエートされたセッ

ション パラメータ フィールドを含む HTTP ヘッダーを挿入します。これらのフィールドは、ACE がクライアント接続を介して受信する HTTP 要求に指定されます。この後、ACE はサーバに HTTP 要求を転送します。

（注）　HTTP ヘッダーのスプーフィングを防ぐため、ACE は、HTTP 要求に挿入するヘッダーのいずれかに一致する受信 HTTP ヘッダーをすべて削除します。

SSL セッション情報を挿入するように ACE に指示した場合、持続性再バランスがデフォルトで有効になっているため、ACE はクライアント接続を介して受信するすべての HTTP 要求に対して、HTTP ヘッダー情報を挿入します。ACE とクライアントが接続を再ネゴシエートする必要がある場合、新しいセッション パラメータを反映するために、ACE はサーバに送信する HTTP ヘッダー情報を更新します。ACE がすべての HTTP 要求に SSL ヘッダー情報を挿入しないようにする場合は、HTTP パラメータ マップで持続性再バランスを無効にします。また、**header modify per-request** コマンドを有効にして HTTP パラメータ マップを作成して、ACE が接続を介して受け取るすべての HTTP 要求にセッション情報を挿入するように指示することもできます。その後、ACE がトラフィックに適用するポリシー マップでパラメータ マップを参照します。HTTP パラメータ マップの作成の詳細については、『*Server Load-Balancing Guide, Cisco ACE Application Control Engine*』を参照してください。

（注）　ACE が挿入できるデータの最大量は 2048 バイトです。この制限を超えると、ACE はデータを切り捨てます。

アクション リスト変更コンフィギュレーション モードで **ssl header-insert session** コマンドを使用して、特定の SSL セッション情報が含まれる HTTP ヘッダーを挿入できます。SSL セッション情報のフィールドを含む HTTP ヘッダーを削除するには、このコマンドの **no** 形式を使用します。

このコマンドの構文は次のとおりです。

ssl header-insert session *specific_field* [**prefix** *prefix_string* | **rename** *new_field_name*]

次のキーワードと引数があります。

- *specific_field*：HTTP ヘッダーに挿入するセッション フィールドの名前です。有効なセッション フィールド名のリストについては、表 3-5 を参照してください。
- **prefix** *prefix_string*：（任意）プレフィックス文字列を指定の SSL セッションのフィールドの前に挿入します。たとえば、SSL セッションのフィールド名 Cipher-Name にプレフィックス Acme-SSL を指定する場合、そのフィールド名は Acme-SSL-Session-Cipher-Name になります。テキスト文字列を入力します。ACE が許容するプレフィックス文字列とフィールドの名前の最大合計数は、32 です。
- **rename** *new_field_name*：（任意）指定された SSL セッション フィールドに新しい名前を割り当てます。スペースを含まないテキスト ストリングを、引用符で囲まずに入力します。ACE が許容するフィールド名文字の最大数は、32 です。

（注）　**prefix** と **rename** オプションは相互に排他的なため、両方を設定することはできません。名前を変更する SSL セッション フィールド名にプレフィックスも割り当てる場合は、**rename** オプションを使用します。

表 3-5 に、サポートされる SSL セッション フィールドを示します。

表 3-5　SSL セッション情報：SSL セッション フィールド

セッション フィールド	説明
Cipher-Key-Size	対称暗号キーのサイズ。 形式：共有キーの長さをバイト単位で指定するすべての整数。 例：Session-Cipher-Key-Size: 32
Cipher-Name	対称暗号スイートの名前。 形式：セッション中にネゴシエートされる暗号スイートの OpenSSL のバージョン名。 例：Session-Cipher-Name: EXP1024-RC4-SHA

表 3-5　SSL セッション情報 : SSL セッション フィールド （続き）

セッション フィールド	説明
Cipher-Use-Size	対称暗号で使用されるサイズ。 形式 : Cipher-Key-Size が使用されるバイト数を指定するすべての整数。使用されるアルゴリズムによっては、すべてのバイト数が使用されない場合があります。 例 : Session-Cipher-Use-Size: 7
Id	SSL セッション ID。デフォルト値は 0 です。 形式 : セッション ID がネゴシエートされていてビッグエンディアン形式で出力された場合に、このセッション中にネゴシエートされた 32 バイトのセッション ID。先頭に 0x が付かず、コロン（:）で区切られた小文字の英数字からなる 16 進数。 例 : Session-Id: 75:45:62:cf:ee:71:de:ad:be:ef:00:33:ee:23:89:25:75:45:62:cf:ee:71:de:ad:be:ef:00:33:ee:23:89:25
Protocol-Version	SSL または TLS のバージョン。 形式 : SSL または TLS のどちらのプロトコルが使用されているかを示す文字列に続いてバージョン番号。 例 : Session-Protocol-Version: TLSv1

表 3-5　SSL セッション情報 : SSL セッション フィールド （続き）

セッション フィールド	説明
Step-Up	SGC または StepUp 暗号化を使用。 形式 : ACE が、128 ビット暗号化を使用してセキュリティを強化するために、Server Gated Cryptography（SGC）または Step-Up 暗号化のどちらを使用しているかを示す文字列。 例 : Session-Step-Up: YES

表 3-5　SSL セッション情報 : SSL セッション フィールド （続き）

セッション フィールド	説明
Verify-Result	SSL セッションの検証結果。 形式 : SSL セッションの検証結果を示す文字列値。表示される可能性のある値は次のとおりです。 • ok : SSL セッションが確立されました。 • certificate is not yet valid : クライアント証明書がまだ有効ではありません。 • certificate is expired : クライアント証明書の期限が切れています。 • bad key size : クライアント証明書のキー サイズが不正です。 • invalid not before field : クライアント証明書の notBefore フィールドが認識されない形式です。 • invalid not after field : クライアント証明書の notAfter フィールドが認識されない形式です。 • certificate has unknown issuer : クライアント証明書の発行元が不明です。 • certificate has bad signature : クライアント証明書に不正な署名が含まれます。 • certificate has bad leaf signature : クライアント証明書に不正なリーフ シグニチャが含まれます。 • unable to decode issuer public key : ACE が発行元の公開キーを解読できません。 • unsupported certificate : クライアント証明書がサポートされていません。 • certificate revoked : クライアント証明書が失効しました。 • internal error : 内部エラーです。 例 : Session-Verify-Result: ok

たとえば、HTTP ヘッダーに SSL セッションで使用される暗号スイート名を挿入するには、次のように入力します。

```
host1/Admin(config-actlist-modify)# ssl header-insert session
Cipher-Name
```

サーバによって受信される各セッション パラメータ フィールドに **ssl header-insert session** コマンドを繰り返します。

SSL の HTTP ヘッダー情報挿入の成功率を追跡するカウンタの詳細については、第 6 章「SSL 情報および統計情報の表示」を参照してください。

SSL サーバ証明書情報の HTTP ヘッダー挿入の設定

ACE 上にある、公開キーまたは証明書のシリアル番号に使用するアルゴリズムなどのサーバ証明書に関する情報をサーバに提供するように、ACE に指示できます。この SSL セッション情報をサーバに転送するために、ACE はサーバ証明書フィールドを含む HTTP ヘッダーを挿入します。これらのフィールドは、ACE がクライアント接続を介して受信する HTTP 要求に指定されます。この後、ACE はサーバに HTTP 要求を転送します。

（注） HTTP ヘッダーのスプーフィングを防ぐため、ACE は、HTTP 要求に挿入するヘッダーのいずれかに一致する受信 HTTP ヘッダーをすべて削除します。

SSL サーバ証明書情報を挿入するように ACE に指示した場合、持続性再バランスがデフォルトで有効になっているため、ACE はクライアント接続を介して受信するすべての HTTP 要求に対して、HTTP ヘッダー情報を挿入します。ACE が接続を介して受信するすべての HTTP 要求に情報を挿入しないようにする場合は、HTTP パラメータ マップで持続性再バランスを無効にします。また、**header modify per-request** コマンドを有効にして HTTP パラメータ マップを作成して、ACE が接続を介して受け取るすべての HTTP 要求に情報を挿入するように指示することもできます。その後、ACE がトラフィックに適用するポリシー マップでパラメータ マップを参照します。HTTP パラメータ マップの作成の詳細については、『*Server Load-Balancing Guide, Cisco ACE Application Control Engine*』を参照してください。

（注） ACE が挿入できるデータの最大量は 512 バイトです。この制限を超えると、ACE はデータを切り捨てます。

アクション リスト変更コンフィギュレーション モードで **ssl header-insert server-cert** コマンドを使用して、特定の SSL サーバ証明書情報フィールドが含まれる HTTP ヘッダーを挿入できます。サーバ証明書情報のフィールドを含む SSL HTTP ヘッダーを削除するには、このコマンドの **no** 形式を使用します。

このコマンドの構文は次のとおりです。

ssl header-insert server-cert *specific_field* [**prefix** *prefix_string* | **rename** *new_field_name*]

次のキーワードと引数があります。

- *specific_field*：HTTP ヘッダーに挿入するサーバ証明書（ServerCert）フィールド名です。有効なサーバ証明書フィールド名のリストについては、表 3-6 を参照してください。
- **prefix** *prefix_string*：（任意）プレフィックス文字列を指定のサーバ証明書フィールド名の前に挿入します。たとえば、サーバ証明書のフィールド名 Authority-Key-Id にプレフィックス Acme-SSL を指定する場合、フィールド名は Acme-SSL-ServerCert-Authority-Key-Id になります。テキスト文字列を入力します。ACE が許容するプレフィックス文字列とフィールドの名前の最大合計数は、32 です。
- **rename** *new_field_name*：（任意）指定されたサーバ証明書フィールドに新しい名前を割り当てます。スペースを含まないテキスト ストリングを、引用符で囲まずに入力します。ACE が許容するフィールドの名前とプレフィックス文字列の最大合計数は、32 です。

（注） **prefix** と **rename** オプションは相互に排他的なため、両方を設定することはできません。名前を変更するサーバ証明書フィールド名にプレフィックスも割り当てる場合は、**rename** オプションを使用します。

表 3-6 に、サポートされる SSL サーバ証明書フィールドを示します。証明書が生成された方法や、使用された主要アルゴリズムによっては、証明書でこれらのフィールドの一部が表示されない可能性があります。

表 3-6　SSL セッション情報：サーバ証明書フィールド

ServerCert フィールド	説明
Authority-Key-Id	X.509 認証のキー ID。 形式：X.509 バージョン 3 認証キー ID を示す、コロンで区切られた、16 進バイトの ASCII 文字列。 例：ServerCert-Authority-Key-Identifier:16:13:15:97:FD:8E:16:B9:D2:99
Basic-Constraints	X.509 の基本制約。 形式：証明書のサブジェクトが認証局として機能できるかどうかを示す文字列。取り得る値は CA=TRUE または CA=FALSE です。 例：ServerCert-Basic-Constraints: CA=TRUE
Certificate-Version	X.509 証明書のバージョン。 形式：X.509 バージョンの数値（3、2、または 1）に続いて、X.509 バージョンの ASN.1 定義値（2、1、または 0）がカッコ内に示されます。 例：ServerCert-Certificate-Version: 3 (0x2)
Data-Signature-Alg	X.509 ハッシュと暗号化方式。 形式：証明書とアルゴリズム パラメータの署名に使用される md5WithRSAEncryption、sha1WithRSAEncryption、または dsaWithSHA1 アルゴリズム。 例：ServerCert-Signature-Algorithm: md5WithRSAEncryption
Fingerprint-SHA1	証明書の SHA1 ハッシュ出力。 形式：コロンで区切られた 16 進バイトの ASCII 文字列。 例：ServerCert-Fingerprint-SHA1: 64:75:CE:AD:9B:71:AC:25:ED:FE:DB:C7:4B:D4:1A:BA
Issuer	X.509 証明書発行元の識別名。 形式：この証明書を発行した認証局を表す文字列。 例：ServerCert-Issuer: CN=Example CA, ST=Virginia, C=US/Email=ca@exampleca.com, 0=Root

表 3-6　SSL セッション情報：サーバ証明書フィールド （続き）

ServerCert フィールド	説明
Issuer-CN	X.509 証明書発行元の通常名。 形式：証明書発行元の通常名を表す文字列。 例：ServerCert-Issuer-CN: www.exampleca.com
Not-After	証明書が無効になる日付。 形式：[Validity] フィールドの [Not After] の日付に指定される、ユニバーサル時間の文字列または汎用時間文字列。 例：ServerCert-Not-After: Dec 12 22:45:13 2014 GMT
Not-Before	証明書が有効になる日付。 形式：[Validity] フィールドの [Not Before] の日付に指定される、ユニバーサル時間の文字列または汎用時間文字列。 例：ServerCert-Not-Before: Dec 12 22:45:13 2011 GMT
Public-Key-Algorithm	公開キーに使用されるアルゴリズム。 形式：証明書の公開キーの作成に使用される rsaEncryption、rsa、または dsaEncryption の公開キー アルゴリズム。 例：ServerCert-Public-Key-Algorithm: rsaEncryption
RSA-Exponent	公開 RSA 指数。 形式：RSA アルゴリズム指数（e）を表すすべての整数。 例：ServerCert-RSA-Exponent: 65537
RSA-Modulus	RSA アルゴリズムの係数。 形式：先頭に 0x が付かず、コロン（:）で区切られた小文字の英数字からなるビッグエンディアン形式で出力された 16 進数の RSA アルゴリズムの係数（n）。指数（e）とともに、この係数は RSA 証明書の公開キー部分を形成します。 例：ServerCert-RSA-Modulus: + 00:d8:1b:94:de:52:a1:20:51:b1:77

表 3-6　SSL セッション情報：サーバ証明書フィールド （続き）

ServerCert フィールド	説明
RSA-Modulus-Size	RSA 公開キーのサイズ。 形式：RSA 係数のすべての整数としてのビット数（通常 512、1024、または 2048）に続いて、ワード ビットが示されます。 例：ServerCert-RSA-Modulus-Size: 1024 bit
Serial-Number	証明書のシリアル番号。 形式：認証局が割り当てるすべて整数値。これには、任意の整数値を指定できます。 例：ServerCert-Serial-Number: 2
Signature	証明書の署名。 形式：先頭に 0x が付かず、コロン（:）で区切られた小文字の英数字からなる、ビッグエンディアン形式で出力された 16 進数の証明書の他のフィールドのセキュア ハッシュとハッシュのデジタル署名。 例：ServerCert-Signature: 33:75:8e:a4:05:92:65
Signature-Algorithm	証明書の署名アルゴリズム。 形式：セキュア ハッシュ アルゴリズムの md5WithRSAEncryption、sha1WithRSAEncryption、または dsaWithSHA1。 例：ServerCert-Signature-Algorithm: nmd5WithRSAEncryption
Subject	X.509 サブジェクトの識別名。 形式：認証される秘密キーを所有するサブジェクトを表す文字列。 例：ServerCert-Subject: CN=Example, ST=Virginia, C=US/Email=ca@example.com, 0=Root

表 3-6　SSL セッション情報：サーバ証明書フィールド （続き）

ServerCert フィールド	説明
Subject-CN	X.509 サブジェクトの共通名。 形式：証明書発行元の通常名を表す文字列。 例：ServerCert-Subject-CN: CN=Example, ST=Virginia, C=US/Email=ca@example.com, 0=Root
Subject-Key-Id	X.509 サブジェクトのキー識別子。 形式：X.509 バージョン 3 サブジェクト キー ID を示す、コロンで区切られた、16 進バイトの ASCII 文字列。 例：ServerCert-Subject-Key-Identifier: 16:13:15:97: FD:8E:16:B9:D2:99

たとえば、HTTP ヘッダーにサーバ証明書の識別名を挿入するには、次のように入力します。

```
host1/Admin(config-actlist-modify)# ssl header-insert server-cert
Subject
```

サーバで受信する各サーバ証明書のフィールドごとに **ssl header-insert server-cert** コマンドを繰り返します。

SSL の HTTP ヘッダー情報挿入の成功率を追跡するカウンタの詳細については、第 6 章「SSL 情報および統計情報の表示」を参照してください。

SSL クライアント証明書情報の HTTP ヘッダー挿入の設定

クライアント認証のために ACE を設定する場合、ACE がクライアントから受信するクライアント証明書に関する情報をサーバに提供するように ACE に指示できます。この SSL セッション情報により、サーバはクライアント要求を適切に管理し、証明書のシリアル番号または公開キー アルゴリズムなどの、証明書内の公開キーの作成に使用される証明書の情報を含めることができます。SSL セッション情報をサーバに転送するために、ACE はクライアント証明書フィールドを含む HTTP ヘッダーを挿入します。これらのフィールドは、ACE がクライアント接続を介して受信する HTTP 要求に指定されます。この後、ACE はサーバに HTTP 要求を転送します。

（注）　HTTP ヘッダーのスプーフィングを防ぐため、ACE は、HTTP 要求に挿入するヘッダーのいずれかに一致する受信 HTTP ヘッダーをすべて削除します。

SSL クライアント証明書情報を挿入するように ACE に指示した場合、持続性再バランスがデフォルトで有効になっているため、ACE はクライアント接続を介して受信するすべての HTTP 要求に対して、HTTP ヘッダー情報を挿入します。ACE が接続を介して受信するすべての HTTP 要求に情報を挿入しないようにする場合は、HTTP パラメータ マップで持続性再バランスを無効にします。また、**header modify per-request** コマンドを有効にして HTTP パラメータ マップを作成して、ACE が接続を介して受け取るすべての HTTP 要求に情報を挿入するように指示することもできます。その後、ACE がトラフィックに適用するポリシー マップでパラメータ マップを参照します。HTTP パラメータ マップの作成の詳細については、『*Server Load-Balancing Guide, Cisco ACE Application Control Engine*』を参照してください。

（注）　SSL クライアント証明書のフィールドの情報が含まれる HTTP ヘッダーを挿入するためには、クライアント認証のために設定された ACE が必要です（「クライアント認証のイネーブル化」を参照）。ヘッダー挿入を設定しても、クライアント認証用に ACE が設定されていない場合、ヘッダー情報は挿入されず、ヘッダー挿入操作を追跡するカウンタはインクリメントしません（第 6 章「SSL 情報および統計情報の表示」を参照）。

（注）　ACE が挿入できるデータの最大量は 512 バイトです。この制限を超えると、ACE はデータを切り捨てます。

アクション リスト変更コンフィギュレーション モードで **ssl header-insert client-cert** コマンドを使用して、特定の SSL クライアント証明書フィールドが含まれる HTTP ヘッダーを挿入できます。HTTP ヘッダーからクライアント証明書情報を削除するには、このコマンドの **no** 形式を使用します。

このコマンドの構文は次のとおりです。

ssl header-insert client-cert *specific_field* [**prefix** *prefix_string* | **rename** *new_field_name*]

次のキーワードと引数があります。

- *specific_field*：HTTP ヘッダーに挿入するクライアント証明書（ClientCert）フィールド名です。有効なサーバ証明書フィールド名のリストについては、表 3-7 を参照してください。
- **prefix** *prefix_string*：（任意）プレフィックス文字列を指定のクライアント証明書フィールド名の前に挿入します。たとえば、クライアント証明書のフィールド名 Authority-Key-Id にプレフィックス Acme-SSL を指定する場合、フィールド名は Acme-SSL-ClientCert-Authority-Key-Id になります。テキスト文字列を入力します。ACE が許容するプレフィックス文字列とフィールドの名前の最大合計数は、32 です。
- **rename** *new_field_name*：（任意）指定されたクライアント証明書フィールドに新しい名前を割り当てます。スペースを含まないテキスト ストリングを、引用符で囲まずに入力します。ACE が許容するフィールドの名前とプレフィックス文字列の最大合計数は、32 です。

（注）　**prefix** と **rename** オプションは相互に排他的なため、両方を設定することはできません。名前を変更するクライアント証明書フィールド名にプレフィックスも割り当てる場合は、**rename** オプションを使用します。

表 3-7 に、サポートされる SSL クライアント証明書フィールドを示します。証明書が生成された方法や、使用された主要アルゴリズムによっては、証明書でこれらのフィールドの一部が表示されない可能性があります。

表 3-7　SSL セッション情報：SSL クライアント証明書フィールド

ClientCert フィールド	説明
Authority-Key-Id	X.509 認証のキー ID。 形式：X.509 バージョン 3 認証キー ID を示す、コロンで区切られた、16 進バイトの ASCII 文字列。 例：ClientCert-Authority-Key-Identifier: 16:13:15:97:FD:8E:16:B9:D2:99
Basic-Constraints	X.509 の基本制約。 形式：証明書のサブジェクトが認証局として機能できるかどうかを示す文字列。取り得る値は CA=TRUE または CA=FALSE の基本制約です。 例：ClientCert-Basic-Constraints: CA=TRUE

表 3-7　SSL セッション情報：SSL クライアント証明書フィールド （続き）

ClientCert フィールド	説明
Certificate-Version	X.509 証明書のバージョン。 形式：X.509 バージョンの数値（3、2、または 1）に続いて、X.509 バージョンの ASN.1 定義値（2、1、または 0）がカッコ内に示されます。 例：ClientCert-Certificate-Version: 3 (0x2)
Data-Signature-Alg	X.509 ハッシュと暗号化方式。 形式：証明書とアルゴリズム パラメータの署名に使用される md5WithRSAEncryption、sha1WithRSAEncryption、または dsaWithSHA1 アルゴリズム。 例：ClientCert-Signature-Algorithm: md5WithRSAEncryption
Fingerprint-SHA1	証明書の SHA1 ハッシュ。 形式：コロンで区切られた 16 進バイトの ASCII 文字列。 例：ClientCert-Fingerprint-SHA1: 64:75:CE:AD:9B:71:AC:25:ED:FE:DB:C7:4B:D4:1:BA
Issuer	X.509 証明書発行元の識別名。 形式：証明書を発行した認証局を表す文字列。 例：ClientCert-Issuer: CN=Example CA, ST=Virginia, C=US/Email=ca@exampleca.com, 0=Root
Issuer-CN	X.509 証明書発行元の通常名。 形式：証明書発行元の通常名を表す文字列。 例：ClientCert-Issuer-CN: www.exampleca.com
Not-After	証明書が無効になる日付。 形式：[Validity] フィールドの [Not After] の日付に指定される、ユニバーサル時間の文字列または汎用時間文字列。 例：ClientCert-Not-After: Dec 12 22:45:13 2014 GMT

表 3-7　SSL セッション情報：SSL クライアント証明書フィールド （続き）

ClientCert フィールド	説明
Not-Before	証明書が有効になる日付。 形式：[Validity] フィールドの [Not Before] の日付に指定される、ユニバーサル時間の文字列または汎用時間文字列。 例：ClientCert-Not-Before: Dec 12 22:45:13 2011 GMT
Public-Key-Algorithm	公開キーに使用されるアルゴリズム。 形式：証明書の公開キーの作成に使用される rsaEncryption、rsa、または dsaEncryption の公開キー アルゴリズム。 例：ClientCert-Public-Key-Algorithm: rsaEncryption
RSA-Exponent	公開 RSA 指数。 形式：RSA アルゴリズムの指数（e）のためにすべての整数として出力される。 例：ClientCert-RSA-Exponent: 65537
RSA-Modulus	RSA アルゴリズムの係数。 形式：先頭に 0x が付かず、コロン（:）で区切られた小文字の英数字からなるビッグエンディアン形式で出力された 16 進数の RSA アルゴリズムの係数（n）。指数（e）とともに、この係数は RSA 証明書の公開キー部分を形成します。 例：ClientCert-RSA-Modulus: +00:d8:1b:94:de:52:a1:20:51:b1:77
RSA-Modulus-Size	RSA 公開キーのサイズ。 形式：RSA 係数のすべての整数としてのビット数（通常 512、1024、または 2048）に続いて、ワード ビットが示されます。 例：ClientCert-RSA-Modulus-Size: 1024 bit

表 3-7　SSL セッション情報：SSL クライアント証明書フィールド （続き）

ClientCert フィールド	説明
Serial-Number	証明書のシリアル番号。 形式：認証局が割り当てるすべて整数値。これには、任意の整数値を指定できます。 例： ClientCert-Serial-Number: 2
Signature	証明書の署名。 形式：先頭に 0x が付かず、コロン（:）で区切られた小文字の英数字からなる、ビッグエンディアン形式で出力された 16 進数の証明書の他のフィールドのセキュア ハッシュとハッシュのデジタル署名。 例：ClientCert-Signature: 33:75:8e:a4:05:92:65
Signature-Algorithm	証明書の署名アルゴリズム。 形式：セキュア ハッシュ アルゴリズムの md5WithRSAEncryption、sha1WithRSAEncryption、または dsaWithSHA1。 例：ClientCert-Signature-Algorithm: md5WithRSAEncryption
Subject	X.509 サブジェクトの識別名。 形式：認証される秘密キーを所有するサブジェクトを表す文字列。 例：ClientCert-Subject: CN=Example, ST=Virginia, C=US/Email=ca@example.com, 0=Root

表 3-7　SSL セッション情報：SSL クライアント証明書フィールド （続き）

ClientCert フィールド	説明
Subject-CN	X.509 サブジェクトの共通名。 形式：証明書が発行されたサブジェクトの共通名を表す文字列。 例：ClientCert-Subject-CN: www.cisco.com
Subject-Key-Id	X.509 サブジェクトのキー識別子。 形式：X.509 バージョン 3 サブジェクト キー ID を示す、コロンで区切られた、16 進バイトの ASCII 文字列。 例：ClientCert-Subject-Key-Identifier: 16:13:15:97:FD:8E:16:B9:D2:99

たとえば、HTTP ヘッダーにクライアント証明書の識別名を挿入するには、次のように入力します。

```
host1/Admin(config-actlist-modify)# ssl header-insert client-cert
Subject
```

サーバで受信する各クライアント証明書のフィールドごとに **ssl header-insert client-cert** コマンドを繰り返します。

SSL の HTTP ヘッダー情報挿入の成功率を追跡するカウンタの詳細については、第 6 章「SSL 情報および統計情報の表示」を参照してください。

レイヤ 7 HTTP ロードバランシング ポリシー マップとアクション リストの関連付け

アクション リストとレイヤ 7 HTTP ロードバランシング ポリシー マップを関連付けるには、ポリシー マップ ロードバランシング クラス コンフィギュレーション モードで **action** コマンドを使用します。クラス マップおよびポリシー マップの作成の詳細については、「SSL 終了用のレイヤ 3 およびレイヤ 4 クラス マップの作成」および「SSL 終了用のレイヤ 3 およびレイヤ 4 ポリシー マップの作成」の項を参照してください。

注意　SSL HTTP ヘッダー挿入が設定されたアクション リストを class-default クラス マップのみと関連付ける必要があります。

このコマンドの構文は次のとおりです。

action *name*

name 引数は、既存アクション リストの識別子です。最大 64 文字の英数字からなる文字列を引用符で囲まずに入力します。

たとえば、レイヤ 7 HTTP ロードバランシング ポリシー マップと SSL URL 書き換えのアクション リストを関連付けるには、次のように入力します。

```
host1/Admin(config)# policy-map type loadbalance http first-match
L7_POLICY
host1/Admin(config-pmap-lb)# class CLASS-DEFAULT
host1/Admin(config-pmap-lb-c)# action SSL_ACTLIST
```

アクションとポリシー マップの関連付けを解除するには、次のように入力します。

```
host1/Admin(config-pmap-lb-c)# no action SSL_ACTLIST
```

HTTP ヘッダー挿入を含む設定の例

ここでは、次の設定例について説明します。

- 最初の HTTP 要求のみへの SSL セッション情報の挿入
- すべての HTTP 要求への SSL セッション情報の挿入

すべての HTTP 要求への SSL セッション情報の挿入

ここでは、SSL セッション情報を挿入するためのアクション リスト（ACTION-SSL-INS）が含まれる設定例を示します。この設定では、接続を介して受信する各 HTTP 要求にセッション情報を挿入するデフォルトの方式を使用します。

設定例は次のとおりです。

```
serverfarm host SFARM-1
  rserver SERVER1
```

```
    inservice
 rserver SERVER2
    inservice

crypto authgroup A1
  cert CACERT3.PEM

ssl-proxy service SSL_PSERVICE_TERMINATION
  key RSAKEY.PEM
  cert RSACERT.PEM
  authgroup A1

class-map type http loadbalance match-all CM-1
  2 match http url /index.html

action-list type modify http ACTION-SSL-INS
  ssl header-insert session Id prefix SSL-
  ssl header-insert server-cert Issuer
  ssl header-insert client-cert Serial-Number rename
Client-Serial-Number

policy-map type loadbalance http first-match PM-HTTP-LB
  class CM-1
    serverfarm SFARM-1
   class class-default
    action ACTION-SSL-INS

policy-map multi-match SP-HTTP-LB-POLICY
  class VIP-MERCURY
    loadbalance vip inservice
    loadbalance policy PM-HTTP-LB
    loadbalance vip icmp-reply
    inspect http
    appl-parameter http advanced-options HTTP-PMAP
    ssl-proxy server SSL_PSERVICE_TERMINATION

interface vlan 2524
  ip address 2001:DB8:1::1/64 <---------- IPv6 address
  or
  ip address 192.168.1.1 255.255.255.0 <--IPv4 address
  access-group input ALL
  service-policy input SP-HTTP-LB-POLICY
  service-policy input MGMT-POLICY
  no shutdown
```

最初の HTTP 要求のみへの SSL セッション情報の挿入

ここでは、SSL セッション情報を挿入するためのアクション リスト（ACTION-SSL-INS）が含まれる設定例を示します。この設定には、ACE が接続を介して受信する最初の HTTP 要求のみにセッション情報を挿入するように ACE に指示する HTTP パラメータ マップ（HTTP-PMAP）が含まれます。この例では、パラメータ マップは **no persistence-rebalance** コマンドを使用して、すべての HTTP 要求への HTTP ヘッダー挿入を無効にします。HTTP パラメータ マップの作成の詳細については、『*Server Load-Balancing Guide, Cisco ACE Application Control Engine*』を参照してください。

設定例は次のとおりです。

```
serverfarm host SFARM-1
  rserver SERVER1
    inservice
 rserver SERVER2
    inservice

crypto authgroup A1
  cert CACERT3.PEM

ssl-proxy service SSL_PSERVICE_TERMINATION
  key RSAKEY.PEM
  cert RSACERT.PEM
  authgroup A1

class-map type http loadbalance match-all CM-1
  2 match http url /index.html

parameter-map type http HTTP-PMAP
  no persistence-rebalance

action-list type modify http ACTION-SSL-INS
  ssl header-insert session Id prefix SSL-
  ssl header-insert server-cert Issuer
  ssl header-insert client-cert Serial-Number rename
Client-Serial-Number

policy-map type loadbalance http first-match PM-HTTP-LB
  class CM-1
    serverfarm SFARM-1
   class class-default
    action ACTION-SSL-INS

policy-map multi-match SP-HTTP-LB-POLICY
```

```
  class VIP-MERCURY
    loadbalance vip inservice
    loadbalance policy PM-HTTP-LB
    loadbalance vip icmp-reply
    inspect http
    appl-parameter http advanced-options HTTP-PMAP
    ssl-proxy server SSL_PSERVICE_TERMINATION

interface vlan 2524
  ip address 2001:DB8:1::1/64 <---------- IPv6 address
  or
  ip address 192.168.1.1 255.255.255.0 <--IPv4 address
  access-group input ALL
  service-policy input SP-HTTP-LB-POLICY
  service-policy input MGMT-POLICY
  no shutdown
```

SSL 終了用のレイヤ 3 およびレイヤ 4 クラス マップの作成

ポリシー マップと関連付けるクラス マップは、指定する基準を満たすトラフィックのフィルタとして機能します。SSL 終了の場合は、次のトラフィック特性の 1 つ以上に基づいて、一致基準を定義できます。

- アクセス リスト
- 仮想 IP アドレス
- 送信元 IP アドレスおよびサブネット マスク
- 宛先 IP アドレスおよびサブネット マスク
- TCP/UDP ポート番号またはポート範囲

コンフィギュレーション モードで **class-map** コマンドを使用して、レイヤ 3 およびレイヤ 4 クラス マップを作成できます。レイヤ 3 およびレイヤ 4 クラス マップの作成および設定の詳細については、『*Server Load-Balancing Guide, Cisco ACE Application Control Engine*』を参照してください。

SSL 終了用のレイヤ 3 およびレイヤ 4 ポリシー マップの作成

SSL 終了の場合、クライアントによって SSL サーバとして認識されるように ACE を設定します。これを実現するには、ACE が着信トラフィックに適用するレイヤ 3 およびレイヤ 4 ポリシー マップを設定します。ポリシー マップは、指定した基準に着信トラフィックが一致するかどうかを判定するために、関連付けられたレイヤ 3 およびレイヤ 4 クラス マップを使用します。一致が見つかると、ACE はクライアントと SSL ハンドシェイクを実行し、関連する SSL プロキシ サーバ サービスで指定したパラメータを使用して、SSL セッションを確立します。

ここでは、次の内容について説明します。

- レイヤ 3 およびレイヤ 4 ポリシー マップの作成
- レイヤ 3 およびレイヤ 4 クラス マップとポリシー マップの関連付け
- ポリシー マップと SSL プロキシ サーバ サービスの関連付け

レイヤ 3 およびレイヤ 4 ポリシー マップの作成

コンフィギュレーション モードで **policy-map** コマンドを使用して、SSL 終了ポリシー マップを作成できます。

このコマンドの構文は次のとおりです。

policy-map multi-match *policy_name*

policy_name 引数は、ポリシー マップに割り当てる名前です。最大 64 文字の英数字からなる、引用符で囲まれていない、スペースを含まないテキスト文字列を入力します。

たとえば、ポリシー マップ L4POLICY を作成するには、次のように入力します。

```
host1/Admin(config)# policy-map multi-match L4POLICY
```

ポリシー マップを作成すると、CLI はポリシー マップ コンフィギュレーション モードになります。

```
host1/Admin(config-pmap)#
```

既存のポリシー マップを削除するには、次のように入力します。

```
host1/Admin(config)# no policy-map L4POLICY
```

SSL クラス マップとポリシー マップの関連付けの詳細については、「レイヤ 3 およびレイヤ 4 クラス マップとポリシー マップの関連付け」の項を参照してください。

レイヤ 3 およびレイヤ 4 クラス マップとポリシー マップの関連付け

ポリシー マップ コンフィギュレーション モードで **class** コマンドを使用して、レイヤ 3 およびレイヤ 4 クラス マップをポリシー マップに関連付けることができます。

このコマンドの構文は次のとおりです。

class *class-map*

class-map 引数は、既存のクラス マップの名前です。最大 64 文字の英数字からなる文字列を引用符で囲まずに入力します。スペースは使用しません。

たとえば、クラス マップ L4VIPCLASS をポリシー マップに関連付けるには、次のように入力します。

```
host1/Admin(config)# policy-map multi-match L4POLICY
host1/Admin(config-pmap)# class L4VIPCLASS
```

クラス マップをポリシー マップに関連付けた後、CLI はポリシーマップ クラス マップ コンフィギュレーション モードになります。

```
host1/Admin(config-pmap-c)#
```

ポリシー マップとクラス マップの関連付けを削除するには、次のように入力します。

```
host1/Admin(config-pmap)# no class L4VIPCLASS
```

クラス マップと SSL プロキシ サービスの関連付けの詳細については、「ポリシー マップと SSL プロキシ サーバ サービスの関連付け」の項を参照してください。

ポリシー マップと SSL プロキシ サーバ サービスの関連付け

ポリシー マップ クラス コンフィギュレーション モードで **ssl-proxy server** コマンドを使用して、SSL プロキシ サーバ サービスをポリシー マップと関連付けることができます。

このコマンドの構文は次のとおりです。

ssl-proxy server *pservice*

pservice 引数は既存の SSL プロキシ サーバ サービスの名前です。最大 64 文字の英数字からなる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 サーバ サービス PSERVICE_SERVER をポリシー マップと関連付けるには、次のように入力します。

```
host1/Admin(config)# policy-map multi-match L4POLICY
host1/Admin(config-pmap)# class L4VIPCLASS
host1/Admin(config-pmap-c)# ssl-proxy server PSERVICE_SERVER
```

クラス マップの関連付けを削除するには、次のように入力します。

```
host1/Admin(config-pmap-c)# no ssl-proxy server PSERVICE_SERVER
```

VLAN へのポリシー マップの適用

ここでは、VLAN トラフィックにレイヤ 3 およびレイヤ 4 ポリシー マップを適用する方法を説明します。ACE では、ポリシーを現在のコンテキスト内のすべての VLAN にグローバルに適用することも、コンテキスト内の特定の VLAN に適用することもできます。

ここでは、次の内容について説明します。

- ポリシー マップのグローバルな適用
- 特定の VLAN へのポリシー マップの適用

ポリシー マップのグローバルな適用

コンフィギュレーション モードで **service-policy** コマンドを使用して、ポリシー マップをコンテキスト内のすべての VLAN にグローバルに適用できます。

このコマンドの構文は次のとおりです。

service-policy input *policy_name*

policy_name 引数は、既存のポリシー マップの名前です。最大 64 文字の英数字からなる文字列を引用符で囲まずに入力します。スペースは使用しません。

たとえば、ポリシー マップ L4POLICY をコンテキストのすべての VLAN にグローバルに適用するには、次のように入力します。

```
host1/Admin(config)# service-policy input L4POLICY
```

ポリシーをすべての VLAN からグローバルに削除するには、次のように入力します。

```
host1/Admin(config)# no service-policy input L4POLICY
```

特定の VLAN へのポリシー マップの適用

特定の VLAN インターフェイスにポリシー マップを適用するには、コンフィギュレーション モードで **interface** コマンドを使用して、インターフェイス コンフィギュレーション モードにする必要があります。

このコマンドの構文は次のとおりです。

interface vlan *vlan*

vlan 引数は、コンテキスト VLAN 番号です。2 ～ 4094 の整数を入力します。

たとえば、VLAN 10 のインターフェイス コンフィギュレーション モードにするには、次のように入力します。

```
host1/Admin(config)# interface vlan 10
host1/Admin(config-if)#
```

インターフェイス コンフィギュレーション モードで **service-policy** コマンドを使用して、インターフェイスにポリシー マップを適用できます。

このコマンドの構文は次のとおりです。

service-policy input *policy-name*

policy-name 引数は、既存のポリシー マップの名前です。最大 64 文字の英数字からなる文字列を引用符で囲まずに入力します。スペースは使用しません。

たとえば、VLAN 10 にポリシー マップ L4POLICY を適用するには、次のように入力します。

```
host1/Admin(config)# interface vlan 10
```

```
host1/Admin(config-if)# service-policy input L4POLICY
```

インターフェイスからポリシーを削除するには、次のように入力します。

```
host1/Admin(config-if)# no service-policy input L4POLICY
```

SSL 終了の設定例

次の例では、クライアントからの SSL または TLS 接続を終了し、次に HTTP サーバへの TCP 接続を確立する、SSL プロキシ サーバとして動作する ACE の実行コンフィギュレーションを示します。ACE は、SSL または TLS 接続を終了すると、クライアントからの暗号文を復号化し、データをクリア テキストとして HTTP サーバに送信します。この例では、SSL 終了設定は太字で示されています。

IPv6 の例

```
access-list ACL1 line 10 extended permit ip any any

probe https GEN-HTTPS
  port 80
  interval 50
  faildetect 5
  expect status 200 200

serverfarm host SFARM1
  description SERVER FARM 1 FOR SSL TERMINATION
  probe GEN-HTTPS
  rserver SERVER1 80
    inservice
  rserver SERVER2 80
    inservice
  rserver SERVER3 80
    inservice
  rserver SERVER4 80
    inservice

serverfarm host SFARM2
  description SERVER FARM 2 FOR SSL TERMINATION
  probe GEN-HTTPS
  rserver SERVER5 80
    inservice
  rserver SERVER6 80
    inservice
```

```
  rserver SERVER7 80
    inservice
  rserver SERVER8 80
    inservice

parameter-map type ssl PARAMMAP_SSL_TERMINATION
  cipher RSA_WITH_3DES_EDE_CBC_SHA
  cipher RSA_WITH_AES_128_CBC_SHA priority 2
  cipher RSA_WITH_AES_256_CBC_SHA priority 3
  version all
parameter-map type connection TCP_PARAM
  syn-data drop
  exceed-mss allow

ssl-proxy service SSL_PSERVICE_SERVER
  ssl advanced-options PARAMMAP_SSL_TERMINATION
  key MYKEY.PEM
  cert MYCERT.PEM

class-map type http loadbalance match-all L7_SERVER_CLASS
  description Sticky for SSL Testing
  2 match http url .*.jpg
  3 match source-address 2001:DB8:1::1/64
class-map type http loadbalance match-all L7_SLB-HTTP_CLASS
  2 match http url .*
  3 match source-address 2001:DB8:1::1/64
class-map match-all L4_SSL-TERM_CLASS
  description SSL Termination VIP
  2 match virtual-address 2001:DB8:1::130/64 tcp eq https

policy-map type loadbalance first-match L7_SSL-TERM_POLICY
  class L7_SERVER_CLASS
    serverfarm SFARM1
    insert-http I_AM header-value "SSL_TERM"
    insert-http SRC_Port header-value "%ps"
    insert-http DEST_IP header-value "%id"
    insert-http DEST_Port header-value "%pd"
    insert-http SRC_IP header-value "is"
  class L7_SLB-HTTP_CLASS
    serverfarm SFARM1
    insert-http I_AM header-value "SSL_TERM"
    insert-http SRC_Port header-value "%ps"
    insert-http DEST_IP header-value "%id"
    insert-http DEST_Port header-value "%pd"
    insert-http SRC_IP header-value "is"
policy-map multi-match L4_SSL-VIP_POLICY
  class L4_SSL-TERM_CLASS
    loadbalance vip inservice
```

```
    loadbalance policy L7_SSL-TERM_POLICY
    loadbalance vip icmp-reply
    ssl-proxy server SSL_PSERVICE_SERVER
    connection advanced-options TCP_PARAM

interface vlan 120
  description Upstream VLAN_120 - Clients and VIPs
  ip address 2001:DB8:120::1/64
  fragment chain 20
  fragment min-mtu 68
  access-group input ACL1
  nat-pool 1 2001:DB8:120::70 2001:DB8:120::7F/64 pat
  service-policy input L4_SSL-VIP_POLICY
  no shutdown
ip route 2001:DB8:120::100/64 2001:DB8:120::B
```

IPv4 の例

```
access-list ACL1 line 10 extended permit ip anyv6 anyv6

probe https GEN-HTTPS
  port 80
  interval 50
  faildetect 5
  expect status 200 200

serverfarm host SFARM1
  description SERVER FARM 1 FOR SSL TERMINATION
  probe GEN-HTTPS
  rserver SERVER1 80
    inservice
  rserver SERVER2 80
    inservice
  rserver SERVER3 80
    inservice
  rserver SERVER4 80
    inservice

serverfarm host SFARM2
  description SERVER FARM 2 FOR SSL TERMINATION
  probe GEN-HTTPS
  rserver SERVER5 80
    inservice
  rserver SERVER6 80
    inservice
  rserver SERVER7 80
    inservice
  rserver SERVER8 80
```

```
    inservice

parameter-map type ssl PARAMMAP_SSL_TERMINATION
  cipher RSA_WITH_3DES_EDE_CBC_SHA
  cipher RSA_WITH_AES_128_CBC_SHA priority 2
  cipher RSA_WITH_AES_256_CBC_SHA priority 3
  version all
parameter-map type connection TCP_PARAM
  syn-data drop
  exceed-mss allow

ssl-proxy service SSL_PSERVICE_SERVER
  ssl advanced-options PARAMMAP_SSL_TERMINATION
  key MYKEY.PEM
  cert MYCERT.PEM

class-map type http loadbalance match-all L7_SERVER_CLASS
  description Sticky for SSL Testing
  2 match http url .*.jpg
  3 match source-address 192.168.130.0 255.255.255.0
class-map type http loadbalance match-all L7_SLB-HTTP_CLASS
  2 match http url .*
  3 match source-address 192.168.130.0 255.255.255.0
class-map match-all L4_SSL-TERM_CLASS
  description SSL Termination VIP
  2 match virtual-address 192.168.130.11 tcp eq https

policy-map type loadbalance first-match L7_SSL-TERM_POLICY
  class L7_SERVER_CLASS
    serverfarm SFARM1
    insert-http I_AM header-value "SSL_TERM"
    insert-http SRC_Port header-value "%ps"
    insert-http DEST_IP header-value "%id"
    insert-http DEST_Port header-value "%pd"
    insert-http SRC_IP header-value "is"
  class L7_SLB-HTTP_CLASS
    serverfarm SFARM1
    insert-http I_AM header-value "SSL_TERM"
    insert-http SRC_Port header-value "%ps"
    insert-http DEST_IP header-value "%id"
    insert-http DEST_Port header-value "%pd"
    insert-http SRC_IP header-value "is"
policy-map multi-match L4_SSL-VIP_POLICY
  class L4_SSL-TERM_CLASS
    loadbalance vip inservice
    loadbalance policy L7_SSL-TERM_POLICY
    loadbalance vip icmp-reply
    ssl-proxy server SSL_PSERVICE_SERVER
```

```
    connection advanced-options TCP_PARAM

interface vlan 120
  description Upstream VLAN_120 - Clients and VIPs
  ip address 192.168.120.1 255.255.255.0
  fragment chain 20
  fragment min-mtu 68
  access-group input ACL1
  nat-pool 1 192.168.120.70 192.168.120.80 netmask 255.255.255.0 pat
  service-policy input L4_SSL-VIP_POLICY
  no shutdown
ip route 10.1.0.0 255.255.255.0 192.168.120.254
```