CSS SSL 設定の例

ここでは、SSL モジュールによる SSL フローの処理を、プロキシ設定を例にあげながら説明します。それぞれの設定の説明箇所では、実行設定の例と図も示します。

ここでは、次の内容について説明します。

- SSL モジュールによる SSL フローの処理
- SSL 透過プロキシ設定（1 つの SSL モジュール）
- SSL 透過プロキシ設定（2 つの SSL モジュール）
- SSL 透過プロキシ設定（HTTP サーバとバックエンド SSL サーバ）
- SSL フル プロキシ設定（1 つの SSL モジュール）
- SSL 開始の設定

SSL モジュールによる SSL フローの処理

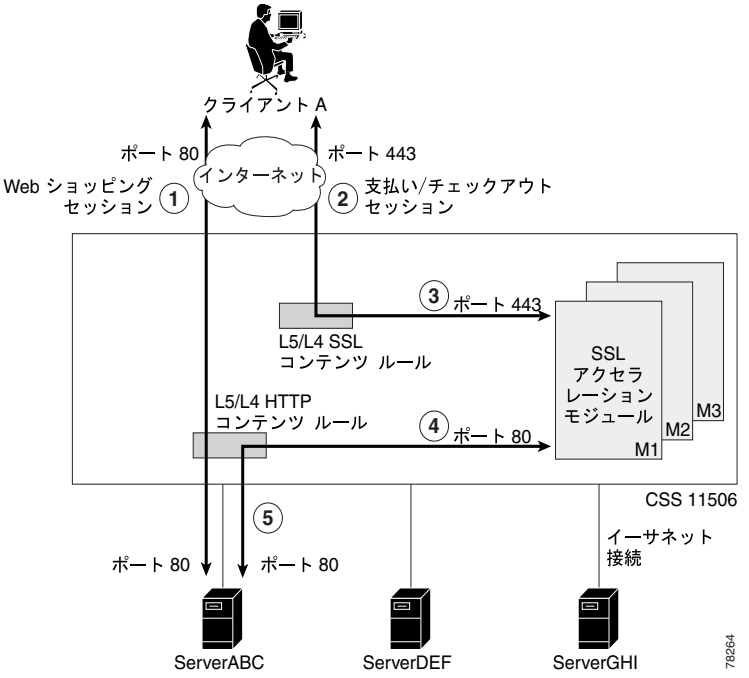
SSL モジュールは、SSL フローを終端させるためにプロキシ サーバとして機能し、着信 SSL トラフィックの TCP エンドポイントとなります。SSL モジュールは、通信のクライアント側とサーバ側で、それぞれ個別に TCP 接続を維持します。プロキシ サーバは TCP ハンドシェイクと SSL ハンドシェイクの両方を実行できます。

次にフロー処理の概要を示します。この例は、CSS および SSL モジュールが着信パケットのフローを HTTPS から HTTP へ、また発信パケットのフローを HTTP から HTTPS へ変換する様子を示しています。

図 8-1 は、バックエンド サーバ（ServerABC、ServerDEF、ServerGHI）からの SSL トラフィックの負荷を分散するように設定された 3 つの SSL モジュール（M1、M2、M3）のある CSS を示しています。図 8-1 ではまた、CSS 内で同じクライアントからの HTTP 接続と SSL 接続とのスティッキ性が維持される様子も示しています。

1. Web のショッピングカート アプリケーションでは通常、トランザクションはショッピングや閲覧で使用される複数の HTTP 接続と、最終的な注文および支払いチェックアウトのシーケンスで使用される少数の SSL 接続で構成されます。クライアントは、このトランザクション全体を通じて、顧客データベース情報を保持する同じサーバに、一貫して接続される必要があります。クライアントからサーバへの HTTP 接続の段階では、クライアントはレイヤ 5 の HTTP クッキーまたは URL コンテンツ ルールを使用して、同じサーバに接続します。チェックアウト時に、クライアントは SSL 接続に移行します。

図 8-1　　複数の SSL モジュールを使用する CSS の設定



2. クライアントは、暗号化された支払いまたは注文情報を SSL 接続（送信先ポート 443 経由で受信される TCP SYN）を通じて送信します。この例では、クライアント接続が CSS に達すると、CSS はレイヤ 5 SSL セッション ID スティッキ コンテンツ ルールを使用して、3 つの SSL モジュール（M1、M2、および M3）に SSL 接続の負荷を分散させます。着信 TCP SYN 接続が SSL モジュール（SSL サーバ）に達すると、SSL モジュールはクライアントからの TCP 接続を終了します。
3. 1 つの SSL モジュール（例では M1）が選択されると、CSS は SSL パケットをそのモジュールに転送します。同じクライアントからの後続する SSL 接続に備え、セッション ID がスティッキ テーブルに保存されます。いったん SSL フローがマップされると、CSS はこの接続で搬入される後続の全パケットを SSL モジュール M1 に転送します。このトランザクションに関連する追加の SSL 接続がある場合（SSL セッション ID で確認される）、CSS はパケットを SSL モジュール M1 に転送し、マップします。
4. SSL モジュールは SSL 接続を終了し、パケット データを復号化します。次に、SSL モジュールは CSS に設定されたコンテンツ ルールへの HTTP 接続を開始します。この HTTP 接続のデータはクリア テキストです。
5. HTTP コンテンツ ルールは、この HTTP 要求にレイヤ 5 HTTP クッキーまたは URL スティッキ コンテンツ ルールを使用します。このクリア テキスト HTTP 要求に含まれるクッキーや URL の文字列は、同じトランザクションの前段階で、SSL を介さない HTTP 接続（オンライン ショッピングなど）によって使用されていたサーバ（ServerABC）の特定に使用されます。CSS はこの要求を ServerABC に転送し、このフローをマップします。いったんフローがマップされると、サーバからの HTTP 応答が、元の要求を送った SSL モジュール（M1）に送信されます。SSL モジュール M1 は応答を SSL パケットとして暗号化（発信パケットのフローを HTTP から HTTPS へ変換）し、適切な SSL 接続を通じてパケットをクライアントに返信します。

TCP 接続が完了すると、4 本のフロー（クライアントと SSL モジュール間の 2 本と SSL モジュールとサーバ間の 2 本）はすべて切断されます。

1 つの SSL セッションが複数の TCP 接続から構成されることもあります。どの接続についても、クライアント、SSL モジュール、およびサーバの間で実行される処理は、上記と同様です。SSL セッション ID によってクライアントと SSL モジュール間のスティッキ性が維持され、クッキーが SSL モジュールとサーバ間のスティッキ性を維持します。この方法によって、Web トランザクション全体にわたってスティッキ性を維持できます。

（注）　デフォルトでは、SSL モジュールの SSL セッション キャッシュには 10000 セッションまで格納できます。キャッシュ サイズは、1 つの SSL モジュールの専用セッション キャッシュに格納できる SSL セッション ID の最大数で表します。SSL サービスに必要な場合は、**session-cache-size** コマンドを使用して、SSL サービスの SSL セッション ID キャッシュのサイズを再設定します。

バックエンド セッション ID のキャッシュに格納できるエントリ数は 4096 です。この値は変更できません。

CSS にバックエンド SSL サーバを設定する場合（図 8-2）、クライアントから CSS への処理のフローは、前の例のステップ 1 ～ 4 と同じです。ただし、図 8-2 のステップ 4 では、SSL モジュールは、サービスをバックエンド SSL サーバに設定した HTTP-SSL コンテンツ ルールへの HTTP 接続を開始します。

図 8-2　バックエンド SSL サーバを使用する CSS の設定

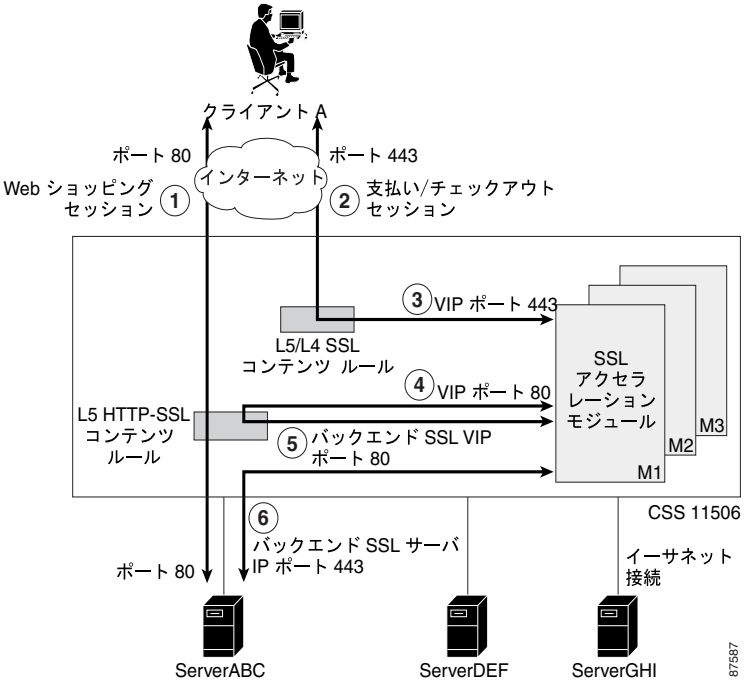


図 8-2 のステップ 5 では、CSS はバックエンド SSL サーバに直接マッピングされる IP アドレスを通じて、クリア テキストのトラフィックを SSL モジュールに返送します。SSL モジュールではこのクリア テキスト接続を終端します。

図 8-2 のステップ 6 で、SSL モジュールはこのトラフィックを再度暗号化し、バックエンド SSL サーバへの SSL 接続を確立します。SSL モジュールは、このトラフィックを CSS を介して、選択したバックエンド SSL サーバに返送します。

SSL 透過プロキシ設定（1 つの SSL モジュール）

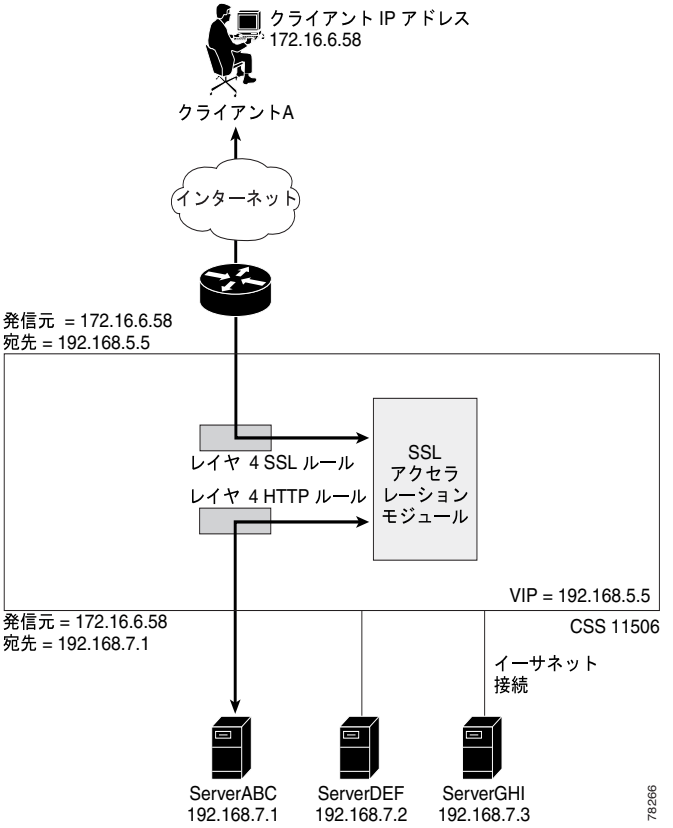
SSL 透過プロキシ サーバは、クライアントの IP アドレスを、サーバへのバックエンド接続の発信元 IP アドレスとして保持するプロキシ サーバです。CSS に SSL 透過プロキシを設定すると、CSS は、発信元 IP アドレスを変更しなくても、ネットワーク上の HTTP サーバへ送信されたクライアント要求を代行受信し、リダイレクトします。

ここでは、クライアント、SSL モジュールを 1 つ備えた CSS、および 3 台の HTTP サーバ（ServerABC、ServerDEF、および ServerGHI）間の簡単な SSL 透過プロキシ設定を示します。この設定では、2 つのコンテンツ ルール（SSL コンテンツ ルールと HTTP コンテンツ ルール）が使用されます。この設定に含まれる SSL モジュールは 1 つだけであり、さらにクライアントとサーバ（SSL）間でスティッキ性を維持する必要がないため、SSL コンテンツ ルールはレイヤ 4 用です。この設定でレイヤ 4 コンテンツ ルールを使用すると、CSS のパフォーマンスが向上します。

図 8-3 は、この透過プロキシ設定を表しています。

図 8-3 の設定例では単純化のために、SSL コンテンツ ルール（ssl-rule）の VIP アドレスと、HTTP コンテンツ ルール（http-rule）の VIP アドレスが同じになっていますが、これら 2 つの VIP アドレスは必ずしも同じでなくてもかまいません。HTTP サーバ上のセキュア コンテンツへのアクセス許可の方法によっては、クリアテキスト コンテンツ ルールをルーティングできないアドレス空間に配置できるようにするために、異なる VIP を指定する必要がある場合もあります。この例では、http-rule コンテンツ ルールに VIP アドレス 192.168.5.5 を指定する代わりに、VIP アドレス 10.1.1.5 を指定することもできます。その結果、クリアテキスト http-rule がインターネットから隔離されるため、CSS をシームレスに統合してトランザクションのセキュリティを維持したまま、柔軟できめ細かな設定を行うことが可能になります。

図 8-3　1 つの SSL モジュールを使用した透過プロキシ設定




```
!*************************** GLOBAL ****************************
  logging commands enable

  ssl associate dsakey dsakey dsakey.pem
  ssl associate dhparam dhparams dhparams.pem
  ssl associate rsakey rsakey rsakey.pem
  ssl associate cert rsacert rsacert.pem

  ftp-record ssl_record 161.44.174.127 anonymous des-password
deye2gtcld1b6feeeebabfcfagyezc5f /
```

```
!************************* CIRCUIT **************************
circuit VLAN1

  ip address 192.168.8.254 255.255.255.0

circuit VLAN2

  ip address 192.168.7.254 255.255.255.0

!********************** SSL PROXY LIST **********************
ssl-proxy-list test
  ssl-server 111
  ssl-server 111 vip address 192.168.5.5
  ssl-server 111 port 443
  ssl-server 111 rsacert rsacert
  ssl-server 111 rsakey rsakey
  ssl-server 111 cipher rsa-with-rc4-128-md5 192.168.5.5 80
  active

!************************* SERVICE **************************
service ssl_module1
  type ssl-accel
  keepalive type none
  slot 5
  add ssl-proxy-list test
  active

service serverABC
  ip address 192.168.7.1
  protocol tcp
  port 80
  keepalive type http
  active

service serverDEF
  ip address 192.168.7.2
  protocol tcp
  port 80
  keepalive type http
  active

service serverGHI
  ip address 192.168.7.3
  protocol tcp
  port 80
  keepalive type http
  active
```

```
!************************** OWNER ****************************
owner ap.com
content ssl-rule
    vip address 192.168.5.5
    protocol tcp
    port 443
    add service ssl_module1
    active

content http-rule
    vip address 192.168.5.5
    protocol tcp
    port 80
    add service serverABC
    add service serverDEF
    add service serverGHI
    advanced-balance cookies
    active
```

SSL 透過プロキシ設定（2 つの SSL モジュール）

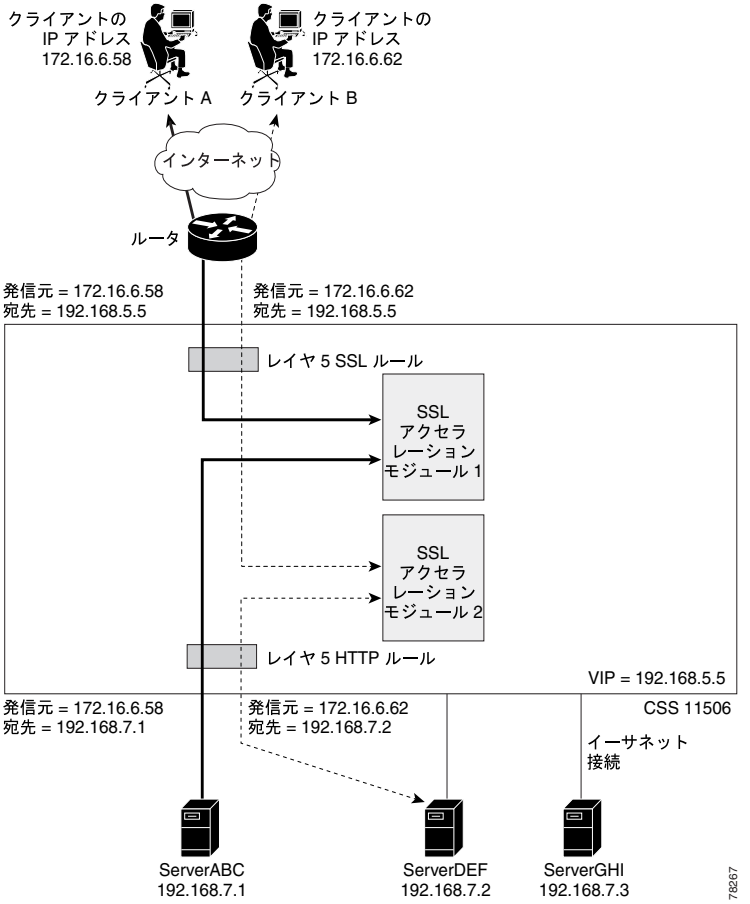
ここでは、クライアント、2 つの SSL モジュールを備えた CSS、および 3 台の HTTP サーバ（ServerABC、ServerDEF、および ServerGHI）間の簡単な SSL 透過プロキシ設定の一例を示します。この設定では、特定の SSL モジュールへのクライアントのスティッキ性を維持するため、レイヤ 5 SSL スティッキ コンテンツ ルールが使用されています。レイヤ 5 SSL スティッキ コンテンツ ルールでは、SSL セッション ID を再使用して、再ハンドシェイク プロセスを省き（SSL ネゴシエーション プロセスが高速化されます）、全体的なパフォーマンスを向上させることができます。

図 8-4 は、この透過プロキシ設定を表しています。

図 8-4 の設定例では単純化のために、SSL コンテンツ ルール（ssl-rule）の VIP アドレスと、HTTP コンテンツ ルール（http-rule）の VIP アドレスが同じになっていますが、これら 2 つの VIP アドレスは必ずしも同じでなくてもかまいません。HTTP サーバ上のセキュア コンテンツへのアクセス許可の方法によっては、クリアテキスト コンテンツ ルールをルーティングできないアドレス空間に配置できるようにするために、このコンテンツ ルールに異なる VIP アドレスを指定する必要がある場合もあります。この例では、http-rule コンテンツ ルールに VIP アドレス 192.168.5.5 を指定する代わりに、VIP アドレス 10.1.1.5 を指定することもで

きます。その結果、クリアテキスト http-rule がインターネットから隔離されるため、CSS をシームレスに統合してトランザクションのセキュリティを維持したまま、柔軟できめ細かな設定を行うことが可能になります。

図 8-4　　2 つの SSL モジュールを使用する透過プロキシ設定



```
!************************** GLOBAL ***************************
  logging commands enable

  ssl associate dsakey dsakey dsakey.pem
  ssl associate rsakey rsakey rsakey.pem
  ssl associate cert rsacert rsacert.pem
  ssl associate dhparam dhparams dhparams.pem

  ftp-record ssl_record 161.44.174.127 anonymous des-password
deye2gtcld1b6feeeebabfcfagyezc5f /
!************************** CIRCUIT **************************
circuit VLAN1

  ip address 192.168.8.254 255.255.255.0

circuit VLAN2

  ip address 192.168.7.254 255.255.255.0

!********************** SSL PROXY LIST **********************
ssl-proxy-list test
  ssl-server 111
  ssl-server 111 vip address 192.168.5.5
  ssl-server 111 rsacert rsacert
  ssl-server 111 rsakey rsakey
  ssl-server 111 cipher rsa-with-rc4-128-md5 192.168.5.5 80
  ssl-server 111 port 443
  active

!************************** SERVICE **************************
service ssl_module1
  type ssl-accel
  keepalive type none
  slot 5
  add ssl-proxy-list test
  active

service ssl_module2
  type ssl-accel
  keepalive type none
  slot 6
  add ssl-proxy-list test
  active
```

```
service serverABC
  ip address 192.168.7.1
  protocol tcp
  port 80
  keepalive type http
  active

service serverDEF
  ip address 192.168.7.2
  protocol tcp
  port 80
  keepalive type http
  active

service serverGHI
  ip address 192.14.7.3
  protocol tcp
  port 80
  keepalive type http
  active

!************************** OWNER ***************************
owner ap.com

content ssl-rule
    vip address 192.168.5.5
    protocol tcp
    port 443
    add service ssl_module1
    add service ssl_module2
    application ssl
    advanced-balance ssl
    active

content http-rule
    vip address 192.168.5.5
    protocol tcp
    port 80
    url "/*"
    add service serverABC
    add service serverDEF
    add service serverGHI
    advanced-balance cookies
    active
```

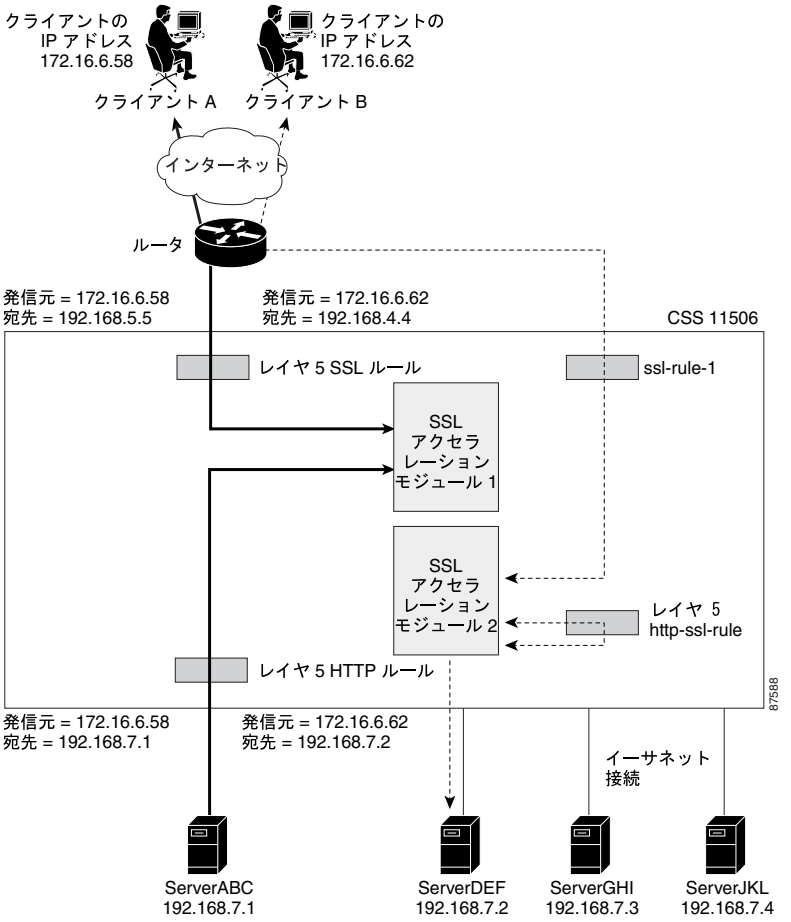
SSL 透過プロキシ設定（HTTP サーバとバックエンド SSL サーバ）

ここでは、2 つのクライアント、2 つの SSL モジュールを備えた CSS、2 つの HTTP サーバ（ServerABC、ServerGHI）および 2 つのバックエンド SSL サーバ（ServerDEF、ServerJKL）を含む SSL 透過プロキシ設定の一例を示します。この設定は、前述の設定とほぼ同じです（「SSL 透過プロキシ設定（2 つの SSL モジュール）」参照）。ただし、バックエンド SSL サーバが構成に含まれる点が異なります。

図 8-5 で、クライアント A の SSL 接続の送信先アドレス 192.168.5.5 はコンテンツ ルール ssl-rule と一致します。CSS は SSL 接続を、SSL モジュール 1 に負荷分散します。このモジュールで SSL 接続が終端し、データが復号化されクリア テキストになり、コンテンツ ルール http-rule への HTTP 接続が開始されます。CSS は、要求を HTTP サーバ ServerABC に転送します。

クライアント B の SSL 接続の送信先アドレス 192.28.4.4 はコンテンツ ルール ssl-rule-1 と一致します。CSS は SSL 接続を SSL モジュール 2 に負荷分散します。このモジュールで SSL 接続が終端し、データが復号化されクリア テキストになり、コンテンツ ルール http-ssl-rule への HTTP 接続が開始されます。CSS は クリア テキスト データを SSL モジュール 2 に返送します。このモジュールで接続が終端し、トラフィックは再度暗号化され、SSL サーバ ServerDEF への SSL 接続が確立されます。

図 8-5　SSL 透過プロキシ設定（HTTP サーバとバックエンド SSL サーバ）



次の設定には、実行設定には表示されないデフォルト値を含むコマンドが含まれています。これらのコマンドはイタリック体で表されています。

```
!************************** GLOBAL ***************************
  logging commands enable

  ssl associate dsakey dsakey dsakey.pem
  ssl associate rsakey rsakey rsakey.pem
  ssl associate cert rsacert rsacert.pem
  ssl associate dhparam dhparams dhparams.pem

  ftp-record ssl_record 161.44.174.127 anonymous des-password
deye2gtcld1b6feeeebabfcfagyezc5f /
!************************** CIRCUIT **************************
circuit VLAN1

  ip address 192.168.8.254 255.255.255.0

circuit VLAN2

  ip address 192.168.7.254 255.255.255.0

!********************** SSL PROXY LIST ***********************
ssl-proxy-list test
  ssl-server 111
  ssl-server 111 vip address 192.168.5.5
  ssl-server 111 port 443
  ssl-server 111 rsacert rsacert
  ssl-server 111 rsakey rsakey
  ssl-server 111 cipher rsa-with-rc4-128-md5 192.168.5.5 80
  active

  ssl-server 2
  ssl-server 2 vip address 192.28.4.4
  ssl-server 2 port 443
  ssl-server 2 rsacert rsacert
  ssl-server 2 rsakey rsakey
  ssl-server 2 cipher rsa-with-rc4-128-md5 192.28.4.4 8080
  active

  backend-server 3
  backend-server 3 ip address 192.168.7.2
  backend-server 3 port 8080
  backend-server 3 server-ip 192.168.7.2
  backend-server 3 rsacert rsacert
  active

  backend-server 4
```

```
  backend-server 4 ip address 192.168.7.4
  backend-server 4 port 8080
  backend-server 4 server-ip 192.168.7.4
  backend-server 4 rsacert rsacert
  active

!************************* SERVICE **************************
service ssl_module1
  type ssl-accel
  keepalive type none
  slot 5
  add ssl-proxy-list test
  active

service ssl_module2
  type ssl-accel
  keepalive type none
  slot 6
  add ssl-proxy-list test
  active

service serverABC
  ip address 192.168.7.1
  protocol tcp
  port 80
  keepalive type http
  active

service serverDEF
  type ssl-accel-backend
  ip address 192.168.7.2
  protocol tcp
  keepalive type ssl
  keepalive port 443
  add ssl-proxy-list test
  active

service serverGHI
  ip address 192.14.7.3
  protocol tcp
  port 80
  keepalive type http
  active

service serverJKL
  type ssl-accel-backend
  ip address 192.168.7.4
  protocol tcp
```

```
  keepalive type ssl
  keepalive port 443
  add ssl-proxy-list test
  active

!************************** OWNER ***************************
owner ap.com

content ssl-rule
    vip address 192.168.5.5
    protocol tcp
    port 443
    add service ssl_module1
    add service ssl_module2
    application ssl
    advanced-balance ssl
    active

content http-rule
    vip address 192.168.5.5
    protocol tcp
    port 80
    url "/*"
    add service serverABC
    add service serverGHI
    advanced-balance cookies
    active

content ssl-rule-1
    vip address 192.28.4.4
    protocol tcp
    port 443
    add service ssl_module1
    add service ssl_module2
    application ssl
    advanced-balance ssl
    active

content http-ssl-rule
    vip address 192.28.4.4
    protocol tcp
    port 8080
    url "/*"
    add service serverDEF
    add service serverJKL
    advanced-balance arrowpoint-cookie
    active
```

SSL フル プロキシ設定（1 つの SSL モジュール）

SSL フル プロキシ サーバは、クライアントの SSL 接続を終端させ、クライアントとは異なる発信元 IP アドレスを使用して HTTP サーバへのバックエンド接続を開始するプロキシ サーバです。この設定では、HTTP サーバへのバックエンド接続については、クライアントの IP アドレスは保持されません。

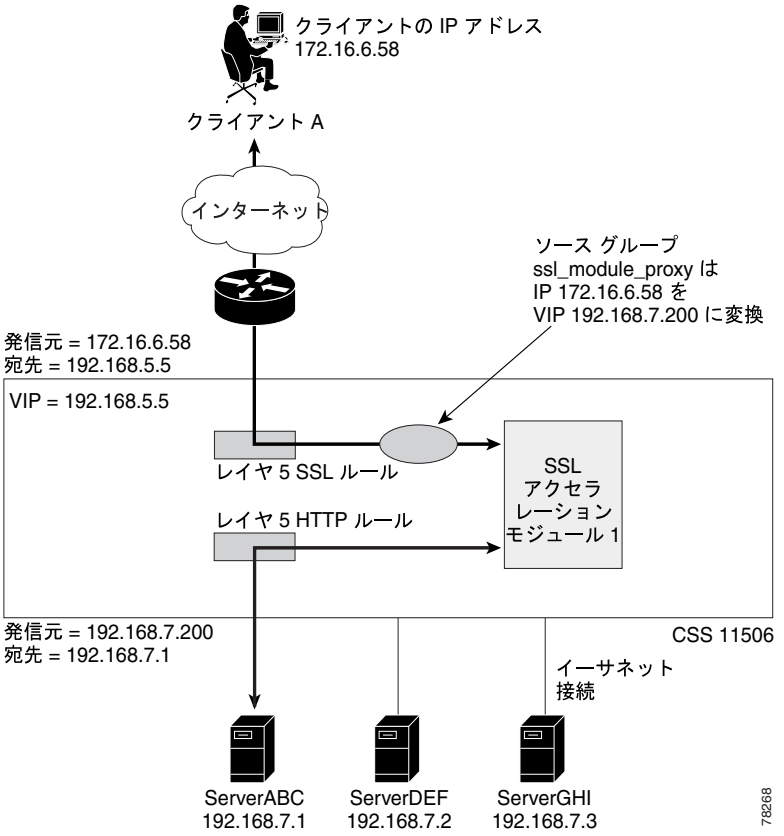
ここでは、クライアント、1 つの SSL モジュールを備えた CSS、および 3 台の HTTP サーバ（ServerABC、ServerDEF、および ServerGHI）間の SSL フル プロキシの設定例を示します。この設定では、レイヤ 5 スティッキ コンテンツ ルールが使用されます。1 つの SSL モジュールでフル プロキシを実装する CSS の設定には、SSL モジュール トラフィックを隔離し、その発信元アドレスを NAT 変換するためのソース グループが含まれます。

図 8-6 は、このフル プロキシ設定を表しています。

図 8-6 の設定例では単純化のために、SSL コンテンツ ルール（ssl-rule）の VIP アドレスと、HTTP コンテンツ ルール（http-rule）の VIP アドレスが同じになっていますが、これら 2 つの VIP アドレスは必ずしも同じでなくてもかまいません。HTTP サーバ上のセキュア コンテンツへのアクセス許可の方法によっては、クリアテキスト コンテンツ ルールをルーティングできないアドレス空間に配置できるようにするために、この コンテンツ ルールに異なる VIP アドレスを指定する必要がある場合もあります。

この例では、http-rule コンテンツ ルールに VIP アドレス 192.168.5.5 を指定する代わりに、VIP アドレス 10.1.1.5 を指定することもできます。その結果、クリアテキスト http-rule がインターネットから隔離されるため、CSS をシームレスに統合してトランザクションのセキュリティを維持したまま、柔軟できめ細かな設定を行うことが可能になります。

図 8-6　　1 つの SSL モジュールを使用したフル プロキシ設定




```
!*************************** GLOBAL ****************************
  logging commands enable

  ssl associate dsakey dsakey dsakey.pem
  ssl associate dhparams dhparams dhparams.pem
  ssl associate rsakey rsakey rsakey.pem
  ssl associate cert rsacert rsacert.pem

  ftp-record ssl_record 161.44.174.127 anonymous des-password
deye2gtcld1b6feeeebabfcfagyezc5f /
```

```
!************************* CIRCUIT **************************
circuit VLAN1

  ip address 192.168.8.254 255.255.255.0

circuit VLAN2

  ip address 192.168.7.254 255.255.255.0

!********************** SSL PROXY LIST **********************
ssl-proxy-list test
  ssl-server 111
  ssl-server 111 vip address 192.168.5.5
  ssl-server 111 port 443
  ssl-server 111 rsacert rsacert
  ssl-server 111 rsakey rsakey
  ssl-server 111 cipher rsa-with-rc4-128-md5 192.168.5.5 80
  active

!************************* SERVICE **************************
service ssl_module1
  type ssl-accel
  keepalive type none
  slot 5
  add ssl-proxy-list test
  active

service ssl_module2
  type ssl-accel
  keepalive type none
  slot 6
  add ssl-proxy-list test
  active

service serverABC
  ip address 192.168.7.1
  protocol tcp
  port 80
  keepalive type http
  active

service serverDEF
  ip address 192.168.7.2
  protocol tcp
  port 80
  keepalive type http
  active
```

```
service serverGHI
  ip address 192.168.7.3
  protocol tcp
  port 80
  keepalive type http
  active

!************************** OWNER ***************************
owner ap.com

content ssl-rule
    vip address 192.168.5.5
    protocol tcp
    port 443
    add service ssl_module1
    add service ssl_module2
    application ssl
    advanced-balance ssl
    active

content http-rule
    vip address 192.168.5.5
    protocol tcp
    port 80
    url "/*"
    add service serverABC
    add service serverDEF
    add service serverGHI
    advanced-balance cookies
    active

!************************** GROUP ***************************
group ssl_module_proxy
  add destination service ssl_module1
  add destination service ssl_module2
  vip address 192.168.8.1
  active
```

SSL 開始の設定

SSL 開始とは、1 つの SSL モジュールを内蔵する、適切に設定された CSS が、クライアントから着信したクリア テキストのフローを、その SSL モジュールに設定されているバックエンド サーバから発信される SSL フローに接続する処理のことです。この設定は、サイト間でセキュリティを確保してデータを転送する手段として使用します。

ここでは、SSL 開始の 2 つの例を示します。

- 4 つのデータ センターへの SSL トンネル
- サーバ認証を行う 1 つのデータ センターへの SSL トンネル

4 つのデータ センターへの SSL トンネル

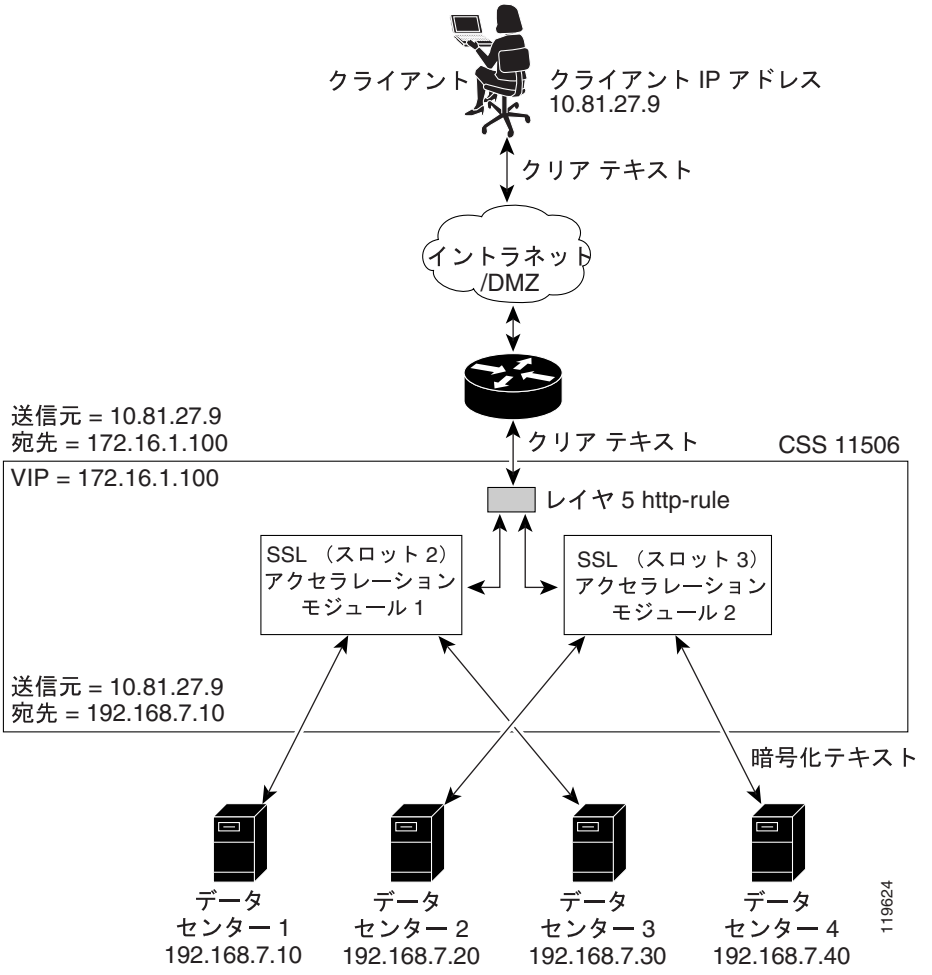
図 8-7 に示すオフィス内では、2 つの SSL モジュールを備えた CSS 11506 が使用されています。クライアントは、クリア テキストを使用して CSS VIP に接続します。CSS は、この接続にロード バランシング（**advanced-balance** スティッキ コマンドの 1 つを適用することにより）と NAT 変換を施し、SSL 開始サービスに送信します。

ssl-init タイプのサービスは、**slot** コマンドにより定義された SSL モジュールに接続を送信するよう CSS に通知します。このサービスは、さらに宛先（リモート サイト）の IP アドレスも定義します。

このサービスから発信された接続が適切な SSL モジュールに着信するには、SSL プロキシ リストに宛先 IP アドレス（**ssl-init** サービスの IP アドレス）が含まれている必要があります。SSL モジュールは、トラフィックを暗号化して、設定された宛先に送信します。

- フローの最適なロード バランシングを行うには、（この例のように）複数の SSL モジュールが存在する場合、SSL 開始の各 VIP 間および各 SSL モジュール間で負荷が均等に分散するようにする必要がある。
- SSL 開始機能では、タイプ **ssl-init** のサービスを使用してプロキシ リストを SSL モジュールに適用する必要がある。

図 8-7　CSS および 4 つのデータ センター間での SSL 開始




```
!!************************** GLOBAL **************************
  ssl associate rsakey rsakey_association rsakey.pem
  ssl associate cert rsacert_association rsacert.pem

  ftp-record acct-ftp 192.168.7.241 root des-password
ig5haaufqbnfuarb/tmp
```

```
!************************ INTERFACE ************************
interface 1/1
  bridge vlan 10
interface 1/2
  bridge vlan 20

!************************** CIRCUIT **************************
circuit VLAN10

  ip address 172.16.1.1 255.255.255.0

circuit VLAN20

  ip address 192.168.7.1 255.255.255.0

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

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

service DC1
  type ssl-init
  ip address 192.168.7.10
  protocol tcp
  port 80
  slot 2
  keepalive type ssl
  keepalive port 443
  add ssl-proxy-list SSLInit_list
active
```

```
service DC2
  type ssl-init
  ip address 192.168.7.20
  protocol tcp
  port 80
  slot 3
  keepalive type ssl
  keepalive port 443
  add ssl-proxy-list SSLInit_list
  active

service DC3
  type ssl-init
  ip address 192.168.7.30
  protocol tcp
  port 80
  slot 2
  keepalive type ssl
  keepalive port 443
  add ssl-proxy-list SSLInit_list
  active

service DC4
  type ssl-init
  ip address 192.168.7.40
  protocol tcp
  port 80
  slot 3
  keepalive type ssl
  keepalive port 443
  add ssl-proxy-list SSLInit_list
  active

!************************** OWNER ***************************
owner Example

  content ssl-init
    protocol tcp
    vip address 172.16.1.100
    port 80
    add service DC1
    add service DC2
    add service DC3
    add service DC4
    advanced-balance arrowpoint-cookie
    active
```

サーバ認証を行う 1 つのデータ センターへの SSL トンネル

図 8-8 に示すオフィスでは、2 つの SSL モジュールを備えた CSS 11506 が使用されています。クライアントは、クリア テキストを使用して CSS VIP 192.168.7.101 に接続します。CSS は、この接続にロード バランシング（**advanced-balance arrowpoint-cookie** スティッキ コマンドを適用する）と NAT 変換を施し、SSL 開始サービスに送信します。

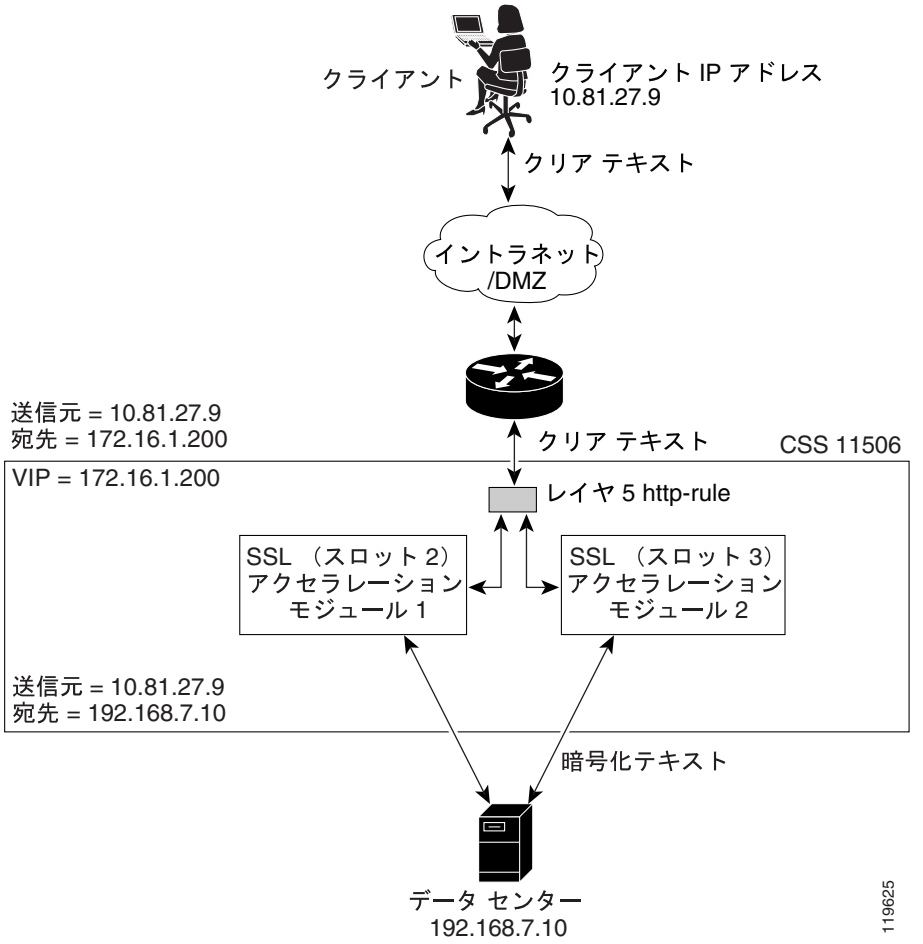
ssl-init タイプのサービスは、**slot** コマンドにより定義された SSL モジュールに接続を送信するよう CSS に通知します。このサービスは、さらに宛先（データ センター）の IP アドレスも定義します。

このサービスからトラフィックが発信され、適切な SSL モジュール（この場合はスロット 2）に着信するには、SSL プロキシ リストに宛先 IP アドレス（**ssl-init** サービスの IP アドレス）が含まれている必要があります。SSL モジュールは、トラフィックを暗号化して、設定された宛先に送信します。SSL サーバ証明書に署名した CA の証明書を追加することにより、CSS は SSL ハンドシェイク中にサーバを認証できます。

次の設定の要件に留意してください。

- 複数の SSL モジュールを最大限に活用するには、設定に含まれる SSL 開始の各 VIP 間、および各 SSL モジュール間で負荷を均等に分散する必要がある。
- タイプ **ssl-init** のサービスを使用して、SSL 開始プロキシ リストを SSL モジュールに適用する必要がある。
- SSL サーバ証明書を発行した CA の証明書を入手する必要がある。CA 証明書をインポートして関連付けた後、SSL プロキシ リスト内で **cacert** として定義します。

図 8-8　CSS と 1 つのデータ センター間での SSL 開始




```
!************************** GLOBAL **************************
  ssl associate rsakey rsakey_association rsakey.pem
  ssl associate cert rsacert_association rsacert.pem

  ftp-record acct-ftp 192.168.7.241 root des-password
ig5haaufqbnfuarb/tmp
  ftp-record config 192.168.1.241 root des-password 4f1bxangrgehjgka
    /users/rclement/ssl-init
```

```
!************************ INTERFACE ************************
interface 1/1
  bridge vlan 10

interface 1/2
  bridge vlan 20

!************************* CIRCUIT *************************
circuit VLAN10

  ip address 172.16.1.1 255.255.255.0

circuit VLAN20

  ip address 192.168.7.1 255.255.255.0

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

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

service DC-SSL1
  type ssl-init
  ip address 192.168.7.10
  protocol tcp
  port 80
  slot 2
  keepalive type ssl
  keepalive port 443
  add ssl-proxy-list SSLInit_list
  active

service DC-SSL2
  type ssl-init
  ip address 192.168.7.10
  protocol tcp
  port 80
  slot 3
  keepalive type ssl
  keepalive port 443
  add ssl-proxy-list SSLInit_list
  active
```

```
!************************** OWNER ***************************
owner Example

  content ssl-init
    protocol tcp
    vip address 192.168.7.200
    port 80
    add service DC-SSL1
    add service DC-SSL2
    advanced-balance arrowpoint-cookie
    active
```