Cisco Unified Presence の設定

この章では、Cisco Unified Presence 向けに適応型セキュリティ アプライアンスを設定する方法を説明します。

この章の内容は、次のとおりです。

Cisco Unified Presence に関する情報

この項では、次のトピックについて取り上げます。

SIP フェデレーション配置の Cisco Unified Presence のアーキテクチャ

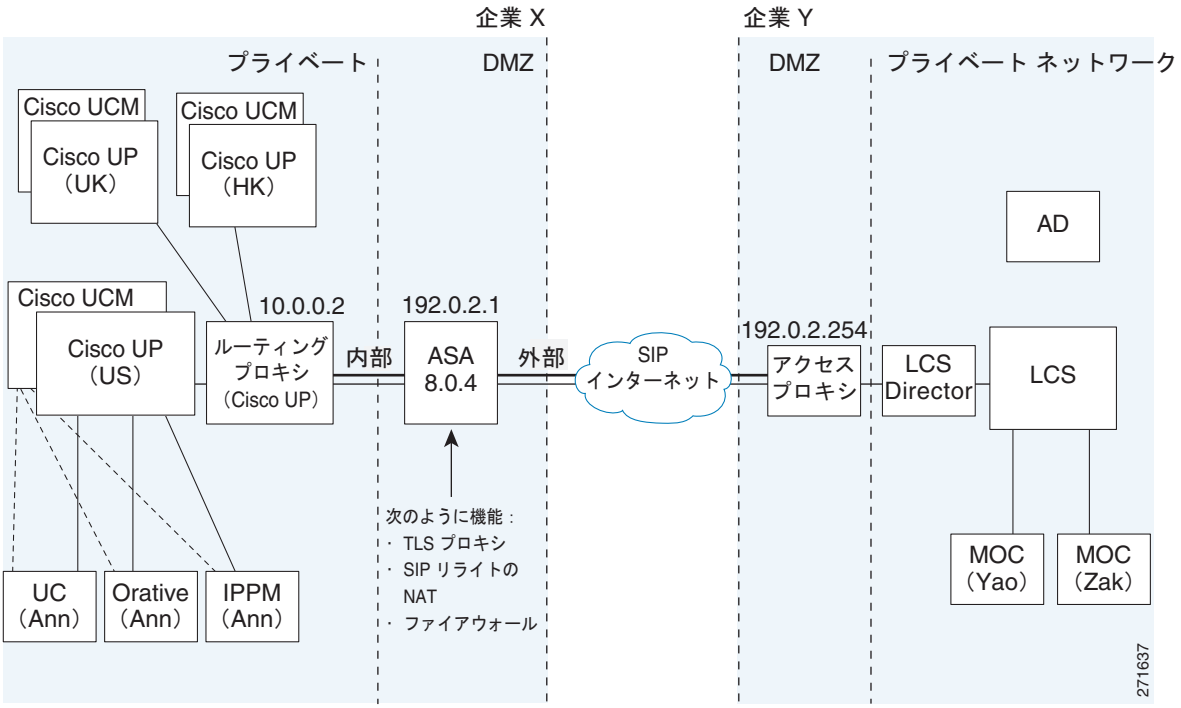
図 17-1 は、ASA を（TLS プロキシとして実装されている）プレゼンス フェデレーション プロキシとして使用する、Cisco Unified Presence/LCS フェデレーションのシナリオを示しています。TLS 接続を使用する 2 つのエンティティは、企業 X の「ルーティング プロキシ」（専用の Cisco UP）と、企業 Y の Microsoft アクセス プロキシです。ただし、構成はこのシナリオに制限されません。ASA の左側には、あらゆる Cisco UP または Cisco UP クラスタを展開できます。リモート エンティティには任意のサーバ（LCS、OCS、または別の Cisco UP）を使用できます。

次のアーキテクチャは、TLS 接続で SIP（または他の ASA 検査対象のプロトコル）を使用する 2 つのサーバの一般的なアーキテクチャです。

エンティティ X：企業 X の Cisco UP/ルーティング プロキシ

エンティティ Y：企業 Y の LCS/OCS 用の Microsoft アクセス プロキシ/エッジ サーバ

図 17-1　標準的な Cisco Unified Presence/LCS フェデレーション シナリオ



上記のアーキテクチャでは、ASA はファイアウォール、NAT、および TLS プロキシとして機能します。これは推奨のアーキテクチャです。ただし、ASA は、NAT および TLS プロキシのみとして機能し、既存のファイアウォールを使用することもできます。

いずれかのサーバが TLS ハンドシェイクを開始できます（クライアントだけが TLS ハンドシェイクを開始できる IP テレフォニーまたは Cisco Unified Mobility とは異なります）。双方向の TLS プロキシ ルールと設定があります。各企業は、ASA を TLS プロキシとして使用できます。

図 17-1 では、NAT または PAT を使用して、エンティティ X のプライベート アドレスを非表示にできます。この状況では、スタティック NAT または PAT を、接続または TLS ハンドシェイク（着信）を開始した外部サーバ（エンティティ Y）に設定する必要があります。通常、パブリック ポートは 5061 にする必要があります。次のスタティック PAT コマンドは、着信接続を受け入れる Cisco UP で必要です。

```
hostname(config)# object network obj-10.0.0.2-01
hostname(config-network-object)# host 10.0.0.2
hostname(config-network-object)# nat (inside,outside) static 192.0.2.1 service tcp 5061
5061
```

次のスタティック PAT は、（SIP SUBSCRIBE を送信して）外部サーバへの接続を開始できる各 Cisco UP に対して設定する必要があります。

アドレスが 10.0.0.2 の Cisco UP の場合は、次のコマンドを入力します。

```
hostname(config)# object network obj-10.0.0.2-02
hostname(config-network-object)# host 10.0.0.2
hostname(config-network-object)# nat (inside,outside) static 192.0.2.1 service tcp 5062
5062
hostname(config)# object network obj-10.0.0.2-03
hostname(config-network-object)# host 10.0.0.2
hostname(config-network-object)# nat (inside,outside) static 192.0.2.1 service udp 5070
5070
hostname(config)# object network obj-10.0.0.2-04
hostname(config-network-object)# host 10.0.0.2
```

```
hostname(config-network-object)# nat (inside,outside) static 192.0.2.1 service tcp 5060
5060
```

アドレスが 10.0.0.3 の別の Cisco UP の場合は、45062 または 45070 などの PAT ポートの異なるセットを使用する必要があります。

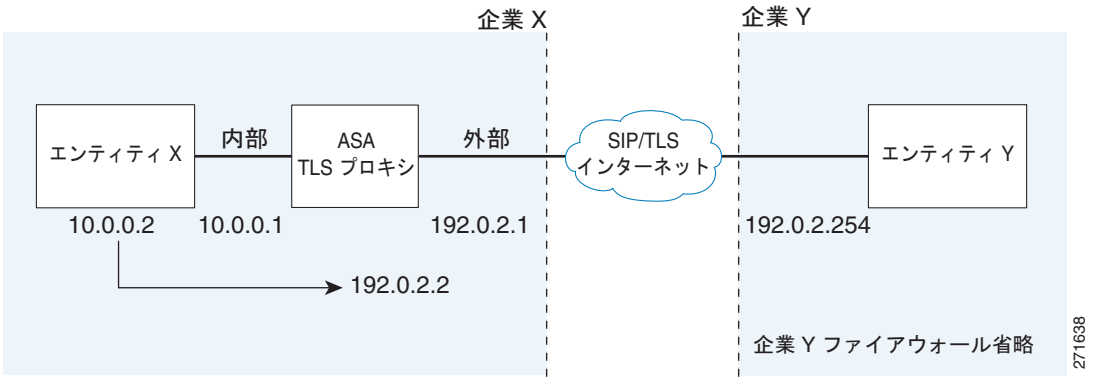
```
hostname(config)# object network obj-10.0.0.3-01
hostname(config-network-object)# host 10.0.0.3
hostname(config-network-object)# nat (inside,outside) static 192.0.2.1 service tcp 5061
45061
hostname(config)# object network obj-10.0.0.3-02
hostname(config-network-object)# host 10.0.0.3
hostname(config-network-object)# nat (inside,outside) static 192.0.2.1 service tcp 5062
45062
hostname(config)# object network obj-10.0.0.3-03
hostname(config-network-object)# host 10.0.0.3
hostname(config-network-object)# nat (inside,outside) static 192.0.2.1 service udp 5070
5070
hostname(config)# object network obj-10.0.0.2-03
hostname(config-network-object)# host 10.0.0.2
hostname(config-network-object)# nat (inside,outside) static 192.0.2.1 service tcp 5070
45070
hostname(config)# object network obj-10.0.0.3-04
hostname(config-network-object)# host 10.0.0.3
hostname(config-network-object)# nat (inside,outside) static 192.0.2.1 service tcp 5060
45060
```

ダイナミック NAT または PAT を、発信接続または TLS ハンドシェイクの残りに対して使用できます。ASA SIP インスペクション エンジンは、必要な変換（フィックスアップ）を処理します。

```
hostname(config)# object network obj-0.0.0.0-01
hostname(config-network-object)# subnet 0.0.0.0 0.0.0.0
hostname(config-network-object)# nat (inside,outside) dynamic 192.0.2.1
```

図 17-2 は、ASA 上のプレゼンス フェデレーション プロキシを通じてエンティティ Y に接続されているエンティティ X を抽象化したシナリオを示しています。プロキシは、エンティティ X と同じ管理ドメイン内に存在します。エンティティ Y は別の ASA をプロキシとして使用できますが、ここでは簡略化のために省略されています。

図 17-2　2 つのサーバ エンティティ間の抽象化されたプレゼンス フェデレーション プロキシ シナリオ



ASA がそのクレデンシャルを保持している場合にエンティティ X のドメイン名を正しく解決するには、ASA を、エンティティ X に対して NAT を実行するように設定します。またドメイン名は、ASA がプロキシ サービスを提供するエンティティ X のパブリック アドレスとして解決されます。

SIP フェデレーション用の Cisco Unified Presence Federation の設定方法の詳細については、『Integration Guide for Configuring Cisco Unified Presence for Interdomain Federation』を参照してください。

http://www.cisco.com/en/US/products/ps6837/products_installation_and_configuration_guides_list.html

プレゼンス フェデレーションの信頼関係

企業内では、自己署名した証明書を使用して信頼関係を設定するか、内部 CA で信頼関係を設定できます。

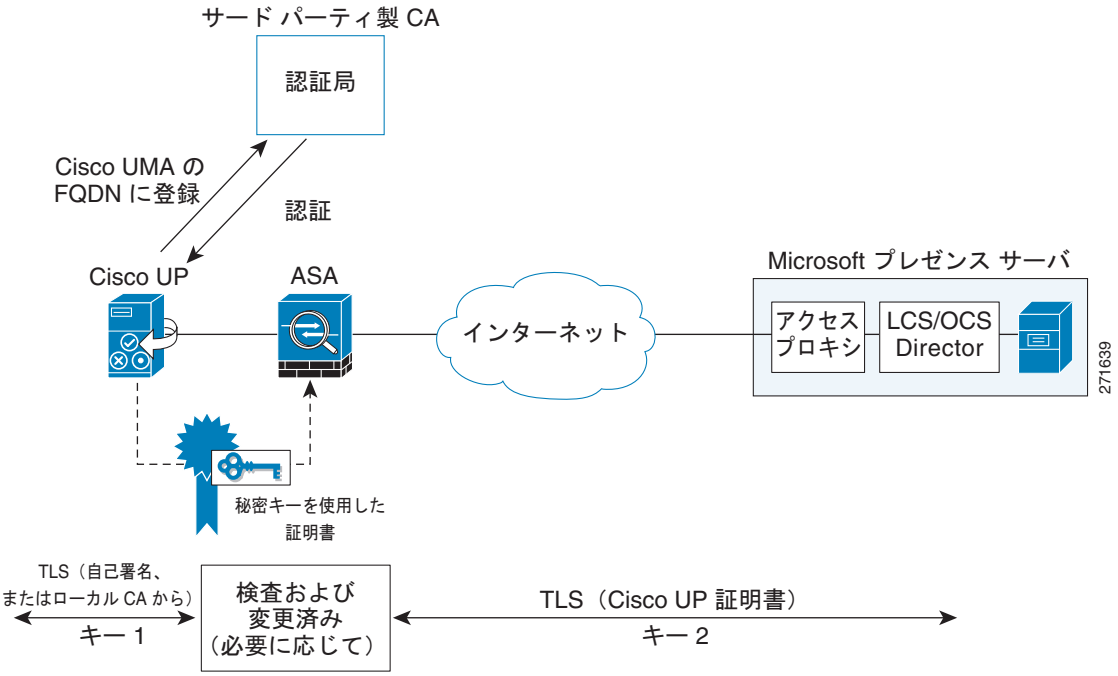
企業間または管理ドメイン間における信頼関係の確立は、フェデレーションにとって重要です。企業間では、信頼できるサードパーティ CA（VeriSign など）を使用する必要があります。ASA は、Cisco UP の Fully Qualified Domain Name（FQDN; 完全修飾ドメイン名）を使用して証明書を取得します（証明書偽装）。

TLS ハンドシェイクの場合、2 つのエンティティが、信頼できるサードパーティ認証局への証明書チェーンを通じてピア証明書を検証できます。両方のエンティティが CA に登録されます。TLS プロキシとしての ASA は、両方のエンティティによって信頼されている必要があります。ASA は、企業のいずれかに常に関連付けられています。企業（図 17-1 の企業 X）内では、エンティティと ASA は、ローカル CA を通じて、または自己署名した証明書を使用して相互に認証を行うことができます。

ASA とリモート エンティティ（エンティティ Y）間で信頼関係を確立するために、ASA はエンティティ X（Cisco UP）の代わりに CA に登録できます。登録要求で、エンティティ X の ID（ドメイン名）が使用されます。

図 17-3 に、信頼関係を確立する方法を示します。ASA は、ASA が Cisco UP であるかのように、Cisco UP FQDN を使用してサードパーティ CA に登録します。

図 17-3　セキュリティ アプライアンスで Cisco Unified Presence を表す方法：証明書偽装



Cisco UP とセキュリティ アプライアンス間でのセキュリティ証明書の交換

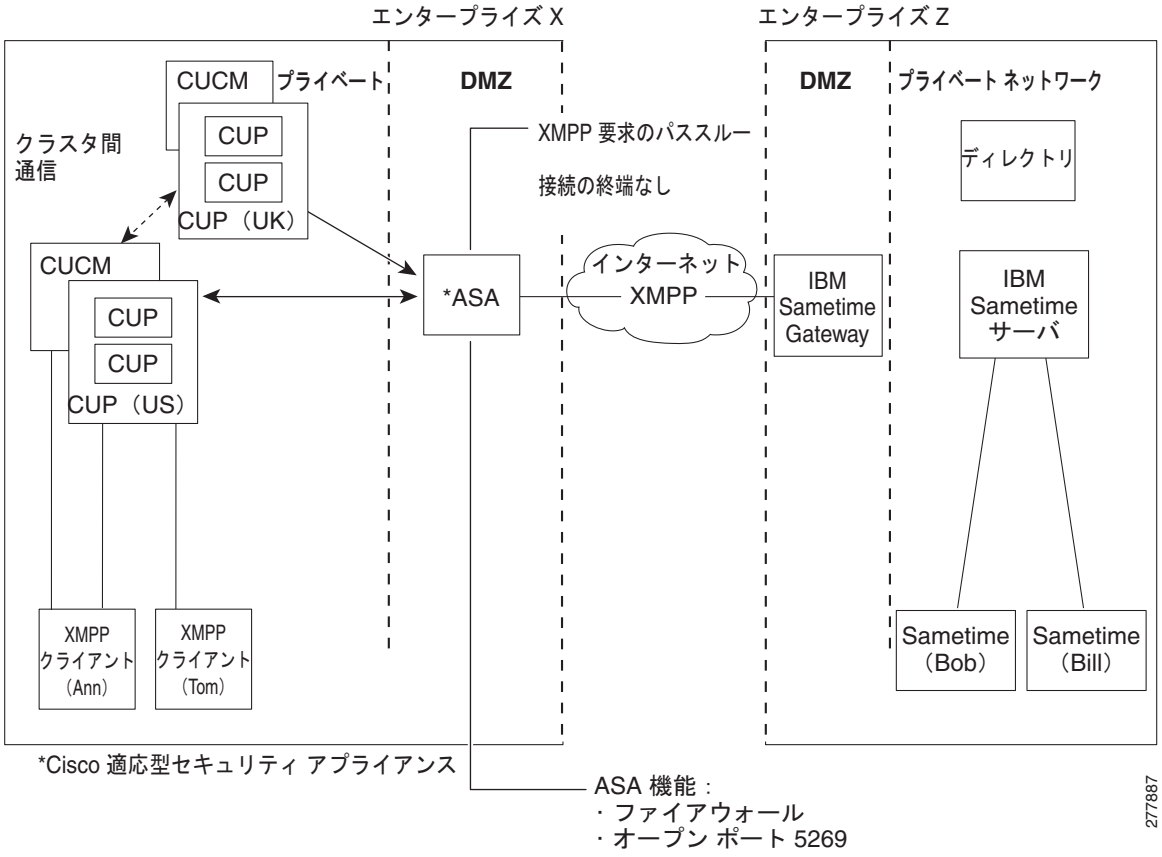
ASA で使用される証明書のキー ペア（cup_proxy_key など）を生成し、TLS ハンドシェイクで ASA から Cisco UP に送信された自己署名証明書を識別するためのトラストポイント（cup_proxy など）を設定します。

ASA で Cisco UP の証明書を信頼するためには、Cisco UP からの証明書を識別するトラストポイント（cert_from_cup など）を作成し、登録タイプを端末として指定して、Cisco UP から受信した証明書を端末に貼り付けることを示します。

XMPP フェデレーション配置

図 17-4 に、Cisco Unified Presence 企業配置と IBM Sametime 企業配置の間の XMPP フェデレーション ネットワークの例を示します。XMPP フェデレーションでは、TLS はオプションです。XMPP フェデレーションでは、ASA はファイアウォールとしてだけ機能します。TLS 機能や PAT は XMPP フェデレーションに対して提供されません。

図 17-4　Cisco Unified Presence と IBM Sametime の間の基本的な XMPP フェデレーション ネットワーク



内部の Cisco Unified Presence 企業配置内には 2 台の DNS サーバが存在します。一方の DNS サーバは、Cisco Unified Presence プライベート アドレスをホストします。もう一方の DNS サーバは、SIP フェデレーション用の Cisco Unified Presence パブリック アドレスと DNS SRV レコード

（_sipfederationtle）、および Cisco Unified Presence による XMPP フェデレーション（_xmpp-server）をホストします。Cisco Unified Presence パブリック アドレスをホストする DNS サーバは、ローカルの DMZ に配置します。

XMPP フェデレーション用の Cisco Unified Presence Federation の設定方法の詳細については、『*Integration Guide for Configuring Cisco Unified Presence Release 8.0 for Interdomain Federation*』を参照してください。

http://www.cisco.com/en/US/products/ps6837/products_installation_and_configuration_guides_list.html

XMPP フェデレーションの設定要件

XMPP フェデレーションの場合、ASA はファイアウォールとしてだけ機能します。ASA 上では、着信と発信の両方の XMPP フェデレーション トラフィックに対してポート 5269 を開く必要があります。

次に、ASA 上でポート 5269 を開く ACL の例をいくつか示します。

ポート 5269 上で任意のアドレスから任意のアドレスへのトラフィックを許可する場合

```
access-list ALLOW-ALL extended permit tcp any any eq 5269
```

ポート 5269 上で任意のアドレスから任意のシングル ノードへのトラフィックを許可する場合

```
access-list ALLOW-ALL extended permit tcp any host <private cup IP address> eq 5269
```

上述の ACL を設定せずに、DNS で追加の XMPP フェデレーション ノードを公開する場合は、次の例のように、追加する各ノードへのアクセスを設定する必要があります。

```
object network obj_host_<private cup ip address>
#host <private cup ip address>
object network obj_host_<private cup2 ip address>
#host <private cup2 ip address>
object network obj_host_<public cup ip address>
#host <public cup ip address>
....
```

次の NAT コマンドを設定します。

```
nat (inside,outside) source static obj_host_<private cup1 IP> obj_host_<public cup IP>
service
obj_udp_source_eq_5269 obj_udp_source_eq_5269
nat (inside,outside) source static obj_host_<private cup1 IP> obj_host_<public cup IP>
service
obj_tcp_source_eq_5269 obj_tcp_source_eq_5269
```

単一のパブリック IP アドレスを DNS で公開し、任意のポートを使用する場合は、次を設定します。

（この例では、追加の XMPP フェデレーション ノードが 2 つあります）

```
nat (inside,outside) source static obj_host_<private cup2 ip> obj_host_<public cup IP>
service
obj_udp_source_eq_5269 obj_udp_source_eq_25269
nat (inside,outside) source static obj_host_<private cup2 ip> obj_host_<public cup IP>
service
obj_tcp_source_eq_5269 obj_tcp_source_eq_25269

nat (inside,outside) source static obj_host_<private cup3 ip> obj_host_<public cup IP>
service
obj_udp_source_eq_5269 obj_udp_source_eq_35269
nat (inside,outside) source static obj_host_<private cup3 ip> obj_host_<public cup IP>
service
```

```
obj_tcp_source_eq_5269 obj_tcp_source_eq_35269
```

すべてがポート 5269 を使用する複数のパブリック IP アドレスを DNS で公開する場合は、次を設定します。

（この例では、追加の XMPP フェデレーション ノードが 2 つあります）

```
nat (inside,outside) source static obj_host_<private cup2 ip> obj_host_<public cup2 IP>
service
obj_udp_source_eq_5269 obj_udp_source_eq_5269
nat (inside,outside) source static obj_host_<private cup2 ip> obj_host_<public cup2 IP>
service
obj_tcp_source_eq_5269 obj_tcp_source_eq_5269

nat (inside,outside) source static obj_host_<private cup3 ip> obj_host_<public cup3 IP>
service
obj_udp_source_eq_5269 obj_udp_source_eq_5269
nat (inside,outside) source static obj_host_<private cup3 ip> obj_host_<public cup IP>
service
obj_tcp_source_eq_5269 obj_tcp_source_eq_5269
```

Cisco Unified Presence のライセンス

ASA でサポートされる Cisco Unified Presence 機能には、Unified Communications Proxy ライセンスが必要です。

次の表に、Unified Communications Proxy ライセンスの詳細をプラットフォーム別に示します。

（注） この機能は、ペイロード暗号化機能のないモデルでは使用できません。

モデル	ライセンス要件[1]
ASA 5505	基本ライセンスと Security Plus ライセンス：2 セッション。 *オプション ライセンス：24 セッション。*
ASA 5510	基本ライセンスと Security Plus ライセンス：2 セッション。 *オプション ライセンス：24、50、または 100 セッション。*
ASA 5520	基本ライセンス：2 セッション。 *オプション ライセンス：24、50、100、250、500、750、または 1000 セッション。*
ASA 5540	基本ライセンス：2 セッション。 *オプション ライセンス：24、50、100、250、500、750、1000、または 2000 セッション。*
ASA 5550	基本ライセンス：2 セッション。 *オプション ライセンス：24、50、100、250、500、750、1000、2000、または 3000 セッション。*
ASA 5580	基本ライセンス：2 セッション。 *オプション ライセンス：24、50、100、250、500、750、1000、2000、3000、5000、または 10,000 セッション。*[2]
ASA 5512-X	基本ライセンス：2 セッション。 *オプション ライセンス：24、50、100、250、または 500 セッション。*

モデル	ライセンス要件[1]
ASA 5515-X	基本ライセンス：2 セッション。 *オプション ライセンス：24、50、100、250、または 500 セッション。*
ASA 5525-X	基本ライセンス：2 セッション。 *オプション ライセンス：24、50、100、250、500、750、または 1000 セッション。*
ASA 5545-X	基本ライセンス：2 セッション。 *オプション ライセンス：24、50、100、250、500、750、1000、または 2000 セッション。*
ASA 5555-X	基本ライセンス：2 セッション。 *オプション ライセンス：24、50、100、250、500、750、1000、2000、または 3000 セッション。*
ASA 5585-X（SSP-10）	基本ライセンス：2 セッション。 *オプション ライセンス：24、50、100、250、500、750、1000、2000、または 3000 セッション。*
ASA 5585-X（SSP-20、-40、または -60）	基本ライセンス：2 セッション。 *オプション ライセンス：24、50、100、250、500、750、1000、2000、3000、5000、または 10,000 セッション。*[2]
ASASM	基本ライセンス：2 セッション。 *オプション ライセンス：24、50、100、250、500、750、1000、2000、3000、5000、または 10,000 セッション。*[2]

1. 次のアプリケーションでは、接続時に TLS プロキシ セッションを使用します。これらのアプリケーションで使用される各 TLS プロキシ セッション（およびこれらのアプリケーションのみ）は UC ライセンスの制限に対してカウントされます。
   - 電話プロキシ
   - プレゼンス フェデレーション プロキシ
   - 暗号化音声インスペクション

   TLS プロキシ セッションを使用するその他のアプリケーション（ライセンスが不要な Mobility Advantage Proxy、個別の IME ライセンスが必要な IME など）では、UC 制限に対してカウントしません。

   UC アプリケーションによっては、1 つの接続に複数のセッションを使用する場合があります。たとえば、プライマリとバックアップの Cisco Unified Communications Manager を電話に設定した場合は、TLS プロキシ接続が 2 つあるため、UC Proxy セッションも 2 つ使用されます。

   tls-proxy maximum-sessions コマンドを使用して TLS プロキシの制限を独立して設定できます。モデルの制限を表示するには、**tls-proxy maximum-sessions ?** コマンドを入力します。デフォルトの TLS プロキシ制限よりも高い UC ライセンスを適用する場合、ASA では、その UC 制限に一致するように TLS プロキシの制限が自動的に設定されます。UC ライセンスの制限よりも TLS プロキシ制限が優先されます。TLS プロキシ制限を UC ライセンスよりも少なく設定すると、UC ライセンスですべてのセッションを使用できません。

   注：「K8」で終わるライセンス製品番号（たとえばユーザ数が 250 未満のライセンス）では、TLS プロキシ セッション数は 1000 までに制限されます。「K9」で終わるライセンス製品番号（たとえばユーザ数が 250 以上のライセンス）では、TLS プロキシの制限はコンフィギュレーションに依存し、モデルの制限が最大数になります。K8 と K9 は、エクスポートについてそのライセンスが制限されるかどうかを示します。K8 は制限されず、K9 は制限されます。

   注：（たとえば **clear configure all** コマンドを使用して）コンフィギュレーションをクリアすると、TLS プロキシ制限がモデルのデフォルトに設定されます。このデフォルトが UC ライセンスの制限よりも小さいと、tls-proxy maximum-sessions コマンドを使用したときに、再び制限を高めるようにエラー メッセージが表示されます。プライマリ装置でフェールオーバーを使用して、**write standby** コマンドを入力して強制的にコンフィギュレーションの同期を行うと、セカンダリ装置で **clear configure all** コマンドが自動的に生成され、セカンダリ装置に警告メッセージが表示されることがあります。コンフィギュレーションの同期によりプライマリ装置の TLS プロキシ制限の設定が復元されるため、この警告は無視できます。

   接続には、SRTP 暗号化セッションを使用する場合もあります。
   - K8 ライセンスでは、SRTP セッション数は 250 までに制限されます。
   - K9 ライセンスに制限はありません。

   （注）メディアの暗号化/復号化を必要とするコールだけが、SRTP 制限に対してカウントされます。コールに対してパススルーが設定されている場合は、両方のレッグが SRTP であっても、SRTP 制限に対してカウントされません。
2. 10,000 セッション UC ライセンスの場合、組み合わせたセッション数は合計 10,000 までですが、電話プロキシ セッションの最大数は 5000 です。

ライセンスの詳細については、一般的な操作のコンフィギュレーション ガイドの Chapter 5, “Managing Feature Licenses,” を参照してください。

SIP フェデレーション用の Cisco Unified Presence Proxy の設定

ここでは、次の項目について説明します。

- 「SIP インスペクションでの TLS プロキシのイネーブル化」（P.17-14）

SIP フェデレーション用の Cisco Unified Presence Federation Proxy の設定のタスク フロー

ローカル ドメイン内にある単一の Cisco UP を含み Cisco UP と ASA の間で自己署名した証明書を使用する（図 17-1 のシナリオを参照）ASA を TLS プロキシとして使用する Cisco Unified Presence/LCS フェデレーション シナリオを設定するには、次のタスクを実行します。

ステップ 1　Cisco UP を含むローカル ドメインで次のスタティック NAT を作成します。

Cisco UP を含むローカル ドメインへの着信接続の場合は、次のコマンドを入力してスタティック PAT を作成します。

```
hostname(config)# object network name
hostname(config-network-object)# host real_ip
hostname(config-network-object)# nat (real_ifc,mapped_ifc) static mapped_ip service {tcp | udp} real_port mapped_port
```

（注）　（SIP SUBSCRIBE を送信して）外部サーバとの接続を開始できる各 Cisco UP の場合は、PAT ポートの異なるセットを使用してスタティック PAT を設定する必要もあります。

発信接続または TLS ハンドシェイクの場合は、ダイナミック NAT または PAT を使用します。ASA SIP インスペクション エンジンは、必要な変換（フィックスアップ）を処理します。

```
hostname(config)# object network name
hostname(config-network-object)# subnet real_ip netmask
hostname(config-network-object)# nat (real_ifc,mapped_ifc) dynamic mapped_ip
```

Cisco Presence Federation プロキシの NAT と PAT を設定する方法の詳細については、第 4 章「ネットワーク オブジェクト NAT の設定」と第 5 章「Twice NAT の設定」を参照してください。

ステップ 2　リモート エンティティに必要な RSA キー ペアとプロキシ証明書（自己署名した証明書）を作成します。「トラストポイントの作成と証明書の生成」（P.17-10）を参照してください。

ステップ 3　証明書をインストールします。「証明書のインストール」（P.17-11）を参照してください。

ステップ 4　Cisco UP サーバに接続している Cisco UP クライアントの TLS プロキシ インスタンスを作成します。「TLS プロキシ インスタンスの作成」（P.17-13）を参照してください。

ステップ 5　SIP インスペクションで TLS プロキシをイネーブルにします。「SIP インスペクションでの TLS プロキシのイネーブル化」（P.17-14）を参照してください。

トラストポイントの作成と証明書の生成

ASA で使用される証明書のキー ペア（cup_proxy_key など）を生成し、TLS ハンドシェイクで ASA から Cisco UP に送信された自己署名証明書を識別するためのトラストポイント（cup_proxy など）を設定します。

	コマンド	目的
ステップ 1	hostname(config)# **crypto key generate rsa label** *key-pair-label* **modulus** *size* **Example:** crypto key generate rsa label ent_y_proxy_key modulus 1024 INFO: The name for the keys will be: ent_y_proxy_key Keypair generation process begin. Please wait... hostname(config)#	トラストポイントに使用できる RSA キー ペアを作成します。 キー ペアは、Cisco UP（リモート エンティティ用のプロキシ）を含むローカル ドメインに提示される自己署名した証明書で使用されます。
ステップ 2	hostname(config)# **crypto ca trustpoint** *trustpoint_name* **Example:** hostname(config)# crypto ca trustpoint ent_y_proxy	リモート エンティティのトラストポイントを作成するには、指定したトラストポイントのトラストポイント コンフィギュレーション モードに入ります。 トラストポイントは、CA が発行する証明書に基づいた CA のアイデンティティとデバイスのアイデンティティを表します。
ステップ 3	hostname(config-ca-trustpoint)# **enrollment self**	自己署名した証明書を生成します。
ステップ 4	hostname(config-ca-trustpoint)# **fqdn none**	登録時に、Subject Alternative Name 拡張子に Fully Qualified Domain Name（FQDN; 完全修飾ドメイン名）を含めるかどうかを指定します。
ステップ 5	hostname(config-ca-trustpoint)# **subject-name** *X.500_name* **Example:** hostname(config-ca-trustpoint)# subject-name cn=Ent-Y-Proxy	登録時に、指定したサブジェクト DN を証明書に含めます。
ステップ 6	hostname(config-ca-trustpoint)# **keypair** *keyname* **Example:** hostname(config-ca-trustpoint)# keypair ent_y_proxy_key	公開キーが認証の対象となるキー ペアを指定します。
ステップ 7	hostname(config-ca-trustpoint)# **exit**	CA トラストポイント コンフィギュレーション モードを終了します。
ステップ 8	hostname(config)# **crypto ca enroll** *trustpoint* **Example:** hostname(config)# crypto ca enroll ent_y_proxy	CA への登録プロセスを開始し、登録するトラストポイントの名前を指定します。

次の作業

ローカル エンティティ トラストストアに証明書をインストールします。ローカル エンティティが信頼するローカル CA に証明書を登録することもできます。「証明書のインストール」（P.17-11）を参照してください。

証明書のインストール

「トラストポイントの作成と証明書の生成」（P.17-10）で作成した ASA の自己署名証明書をエクスポートし、信頼できる証明書としてローカル エンティティにインストールします。このタスクは、ローカル エンティティで ASA を認証するために必要です。

前提条件

リモート エンティティが信頼するプロキシ証明書を ASA で作成するには、信頼できる CA から証明書を取得します。信頼できる CA から証明書を取得する方法については、一般的な操作のコンフィギュレーション ガイドの“Configuring Digital Certificates” section on page 36-11 を参照してください。

	コマンド	目的
ステップ 1	hostname(config)# **crypto ca export** *trustpoint* **identity-certificate** **Example:** hostname(config)# crypto ca export ent_y_proxy identity-certificate	ASA 自己署名した（ID）証明書をエクスポートします。
ステップ 2	hostname(config)# **crypto ca trustpoint** *trustpoint_name* **Example:** hostname(config)# crypto ca trustpoint ent_x_cert ! for Entity X's self-signed certificate	ローカル エンティティのトラストポイントを作成するには、指定したトラストポイントのトラストポイント コンフィギュレーション モードに入ります。 トラストポイントは、CA が発行する証明書に基づいた CA のアイデンティティとデバイスのアイデンティティを表します。
ステップ 3	hostname(config-ca-trustpoint)# **enrollment terminal**	このトラストポイントで使用するカット アンド ペースト登録（手動登録）を指定します。 ローカル エンティティで自己署名した証明書を使用する場合は、自己署名した証明書をインストールする必要があります。ローカル エンティティで CA によって発行された証明書を使用する場合は、CA 証明書をインストールする必要があります。この設定は、自己署名した証明書を使用するためのコマンドを示しています。
ステップ 4	hostname(config-ca-trustpoint)# **exit**	CA トラストポイント コンフィギュレーション モードを終了します。
ステップ 5	hostname(config)# **crypto ca authenticate** *trustpoint* **Example:** hostname(config)# crypto ca authenticate ent_x_cert Enter the base 64 encoded CA certificate. End with a blank line or the word "quit" on a line by itself [certificate data omitted] Certificate has the following attributes: Fingerprint: 21B598D5 4A81F3E5 0B24D12E 3F89C2E4 % Do you accept this certificate? [yes/no]: yes Trustpoint CA certificate accepted. % Certificate successfully imported	ローカル エンティティに作成したトラストポイントと関連付けられている CA 証明書をインストールし、認証します。 *trustpoint* には、CA 証明書を取得するトラストポイントを指定します。名前の最大長は 128 文字です。 ASA は、base-64 形式で CA 証明書を端末に貼り付けることを求めるプロンプトを表示します。
ステップ 6	hostname(config)# **crypto ca trustpoint** *trustpoint_name* **Example:** hostname(config)# crypto ca trustpoint ent_y_ca ! for Entity Y's CA certificate	次のコマンドを入力して、ASA でリモート エンティティ証明書に署名する CA 証明書をインストールします。これは、ASA でリモート エンティティを認証するために必要な手順です。
ステップ 7	hostname(config-ca-trustpoint)# **enrollment terminal**	このトラストポイントで使用するカット アンド ペースト登録（手動登録）を指定します。

	コマンド	目的
ステップ 8	hostname(config-ca-trustpoint)# **exit**	CA トラストポイント コンフィギュレーション モードを終了します。
ステップ 9	hostname(config)# **crypto ca authenticate** *trustpoint* **Example:** hostname(config)# crypto ca authenticate ent_y_ca Enter the base 64 encoded CA certificate. End with a blank line or the word "quit" on a line by itself MIIDRTCCAu+gAwIBAgIQKVcqP/KW74VP0NZzL+JbRTANBgkqhkiG9w0BAQUFADCB [certificate data omitted] /7QEM8izy0EOTSErKu7Nd76jwf5e4qttkQ==	ローカル エンティティに作成したトラストポイントと関連付けられている CA 証明書をインストールし、認証します。 ASA は、base-64 形式で CA 証明書を端末に貼り付けることを求めるプロンプトを表示します。

次の作業

ASA でローカル エンティティとリモート エンティティのトラストポイントを作成し、証明書を作成したら、TLS プロキシ インスタンスを作成します。「TLS プロキシ インスタンスの作成」（P.17-13）を参照してください。

TLS プロキシ インスタンスの作成

（TLS ハンドシェイクをクライアントのみが開始できる IP テレフォニーや Cisco Unified Mobility とは異なり）いずれかのサーバが TLS ハンドシェイクを開始できるので、双方向の TLS プロキシ ルールを設定する必要があります。各企業は、ASA を TLS プロキシとして使用できます。

接続を開始したローカル エンティティとリモート エンティティの TLS プロキシ インスタンスをそれぞれ作成します。TLS 接続を開始するエンティティは、「TLS クライアント」の役割に含まれます。TLS プロキシには、「クライアント」プロキシと「サーバ」プロキシの厳密な定義があるため、いずれかのエンティティが接続を開始できる場合、2 つの TLS プロキシ インスタンスを定義する必要があります。

	コマンド	目的
ステップ 1	! Local entity to remote entity hostname(config)# **tls-proxy** proxy_name **Example:** hostname(config)# tls-proxy ent_x_to_y	TLS プロキシ インスタンスを作成します。
ステップ 2	hostname(config-tlsp)# **server trust-point** *proxy_name* **Example:** hostname(config-tlsp)# server trust-point ent_y_proxy	TLS ハンドシェイク中に提示されるプロキシ トラストポイント証明書を指定します。 証明書は、ASA が所有している必要があります（ID 証明書）。 **server trust-point** コマンドの *proxy_name* には、リモート エンティティのプロキシ名を指定します。
ステップ 3	hostname(config-tlsp)# **client trust-point** *proxy_trustpoint* **Example:** hostname(config-tlsp)# client trust-point ent_x_proxy	ASA が TLS クライアントの役割と見なす場合に、ASA が TLS ハンドシェイクで使用するトラストポイントおよび関連付けられた証明書を指定します。 証明書は、ASA が所有している必要があります（ID 証明書）。 **client trust-point** コマンドの *proxy_trustpoint* には、ローカル エンティティのプロキシを指定します。

	コマンド	目的
ステップ 4	hostname(config-tlsp)# **client cipher-suite** *cipher_suite* **Example:** hostname(config-tlsp)# client cipher-suite aes128-sha1 aes256-sha1 3des-sha1 null-sha1	暗号スイートの設定を指定します。 クライアント プロキシ（サーバに対して TLS クライアントとして機能するプロキシ）の場合、ユーザ定義の暗号スイートによってデフォルトの暗号スイートが置き換えられます。
ステップ 5	! Remote entity to local entity hostname(config)# **tls-proxy** *proxy_name* **Example:** tls-proxy ent_y_to_x	TLS プロキシ インスタンスを作成します。
ステップ 6	hostname(config-tlsp)# **server trust-point** *proxy_name* **Example:** hostname(config-tlsp)# server trust-point ent_x_proxy	TLS ハンドシェイク中に提示されるプロキシ トラストポイント証明書を指定します。 **server trust-point** コマンドの *proxy_name* には、ローカル エンティティのプロキシ名を指定します。
ステップ 7	hostname(config-tlsp)# **client trust-point** *proxy_trustpoint* **Example:** hostname(config-tlsp)# client trust-point ent_y_proxy	ASA が TLS クライアントの役割と見なす場合に、ASA が TLS ハンドシェイクで使用するトラストポイントおよび関連付けられた証明書を指定します。 **client trust-point** コマンドの *proxy_trustpoint* には、リモート エンティティのプロキシを指定します。
ステップ 8	hostname(config-tlsp)# **client cipher-suite** *cipher_suite* **Example:** hostname(config-tlsp)# client cipher-suite aes128-sha1 aes256-sha1 3des-sha1 null-sha1	暗号スイートの設定を指定します。

次の作業

TLS プロキシ インスタンスを作成したら、そのインスタンスを SIP インスペクション用にイネーブルにします。「SIP インスペクションでの TLS プロキシのイネーブル化」（P.17-14）を参照してください。

SIP インスペクションでの TLS プロキシのイネーブル化

SIP インスペクションで TLS プロキシをイネーブルにし、接続を開始できる両方のエンティティのポリシーを定義します。

	コマンド	目的
ステップ 1	hostname(config)# **access-list** *id* **extended permit tcp host** *src_ip* **host** *dest_ip* **eq** *port* **Examples:** access-list ent_x_to_y extended permit tcp host 10.0.0.2 host 192.0.2.254 eq 5061 access-list ent_y_to_x extended permit tcp host 192.0.2.254 host 192.0.2.1 eq 5061	アクセス コントロール エントリを追加します。ACL を使用して、検査するトラフィックのクラスを指定します。
ステップ 2	hostname(config)# **class-map** *class_map_name* **Example:** hostname(config)# class-map ent_x_to_y	検査するセキュア SIP クラスのトラフィックを設定します。 *class_map_name* には、SIP クラス マップの名前を指定します。

	コマンド	目的
ステップ 3	hostname(config-cmap)# **match access-list** *access_list_name* **Example:** hostname(config-cmap)# match access-list ent_x_to_y	検査するトラフィックを指定します。
ステップ 4	hostname(config-cmap)# **exit**	クラス マップ コンフィギュレーション モードを終了します。
ステップ 5	hostname(config)# **policy-map type inspect sip** *policy_map_name* **Example:** hostname(config)# policy-map type inspect sip sip_inspect	SIP インスペクション アプリケーション トラフィックの特別なアクションを定義します。
ステップ 6	hostname(config-pmap)# **parameters** ! SIP inspection parameters	SIP インスペクションのパラメータを指定します。パラメータは、インスペクション エンジンの動作に影響します。 パラメータ コンフィギュレーション モードで使用できるコマンドは、アプリケーションによって異なります。
ステップ 7	hostname(config-pmap)# **exit**	ポリシー マップ コンフィギュレーション モードを終了します。
ステップ 8	hostname(config)# **policy-map** name **Example:** hostname(config)# policy-map global_policy	ポリシー マップを設定し、アクションをトラフィック クラスに関連付けます。
ステップ 9	hostname(config-pmap)# **class** classmap_*name* **Example:** hostname(config-pmap)# class ent_x_to_y	クラス マップ トラフィックにアクションを割り当てることができるように、クラス マップをポリシー マップに割り当てます。 *classmap_name* には、SIP クラス マップの名前を指定します。
ステップ 10	hostname(config-pmap)# **inspect sip** *sip_map* **tls-proxy** *proxy_name* hostname(config-pmap)# inspect sip sip_inspect tls-proxy ent_x_to_y	指定された SIP インスペクション セッションで TLS プロキシをイネーブルにします。
ステップ 11	hostname(config-pmap)# **exit**	ポリシー マップ コンフィギュレーション モードを終了します。
ステップ 12	hostname(config)# **service-policy** *policy_map_name* **global** **Example:** hostname(config)# service-policy global_policy global	すべてのインターフェイスで SIP インスペクションに対するサービス ポリシーをイネーブルにします。 policy-map コマンドの名前には、グローバル ポリシー マップの名前を指定します。

Cisco Unified Presence のモニタリング

デバッグは、IP テレフォニーの TLS プロキシのデバッグと似ています。TLS プロキシ接続の問題をデバッグするために、SSL の syslog とともに TLS プロキシのデバッグ フラグをイネーブルにできます。

たとえば、TLS プロキシ関連のデバッグと syslog 出力だけをイネーブルにするには、次のコマンドを使用します。

```
hostname(config)# debug inspect tls-proxy events
hostname(config)# debug inspect tls-proxy errors
hostname(config)# logging enable
hostname(config)# logging timestamp
```

```
hostname(config)# logging list loglist message 711001
hostname(config)# logging list loglist message 725001-725014
hostname(config)# logging list loglist message 717001-717038
hostname(config)# logging buffer-size 1000000
hostname(config)# logging buffered loglist
hostname(config)# logging debug-trace
```

TLS プロキシのデバッグ方法および出力例については、「TLS プロキシのモニタリング」（P.15-16）を参照してください。

SIP インスペクション エンジンのデバッグを行うには、**debug sip** コマンドをイネーブルにします。コマンド リファレンスを参照してください。

また、次のコマンドを入力して、未加工のデータおよび復号化されたデータを TLS プロキシでキャプチャできます。

```
hostname# capture mycap interface outside (capturing raw packets)
hostname# capture mycap-dec type tls-proxy interface outside (capturing decrypted data)
hostname# show capture capture_name
hostname# copy /pcap capture:capture_name tftp://tftp_location
```

Cisco Unified Presence の設定例

ここでは、次の内容について説明します。

SIP フェデレーション配置の設定例

次の例は、図 17-5 に示されている Cisco Unified Presence の TLS プロキシを実行するために必要な ASA の設定を示しています。この例では、単一の Cisco UP（エンティティ X）がローカル ドメイン内にあり、自己署名した証明書がエンティティ X と ASA 間で使用されていることを前提にしています。

（SIP SUBSCRIBE を送信して）外部サーバとの接続を開始できる Cisco UP ごとに、スタティック PAT も設定する必要があります。また、そのアドレス（この例では 10.0.0.3）を使用する別の Cisco UP がある場合は、PAT ポートの異なるセット（45062 や 45070 など）を使用する必要があります。ダイナミック NAT または PAT は、発信接続または TLS ハンドシェイクに使用できます。ASA SIP インスペクション エンジンは、必要な変換（フィックスアップ）を処理します。

必要な RSA キー ペアを作成すると、エンティティ X（エンティティ Y のプロキシ）に提示されるキーペアは自己署名した証明書で使用されます。エンティティ Y に対するプロキシ証明書を作成すると、証明書はエンティティ X トラストストアにインストールされます。この証明書は、エンティティ X が信頼するローカル CA に登録することもできます。

エンティティ X で ASA を認証するには、ASA の自己署名した証明書（ent_y_proxy）をエクスポートし、それをエンティティ X に信頼できる証明書としてインストールする必要があります。エンティティ X とのハンドシェイク中に ASA でエンティティ X を認証するには、エンティティ X の証明書をエクスポートし、それを ASA にインストールする必要があります。エンティティ X で自己署名した証明書を使用する場合は、自己署名した証明書をインストールする必要があります。エンティティ X で CA によって発行された証明書を使用する場合は、CA 証明書をインストールする必要があります。

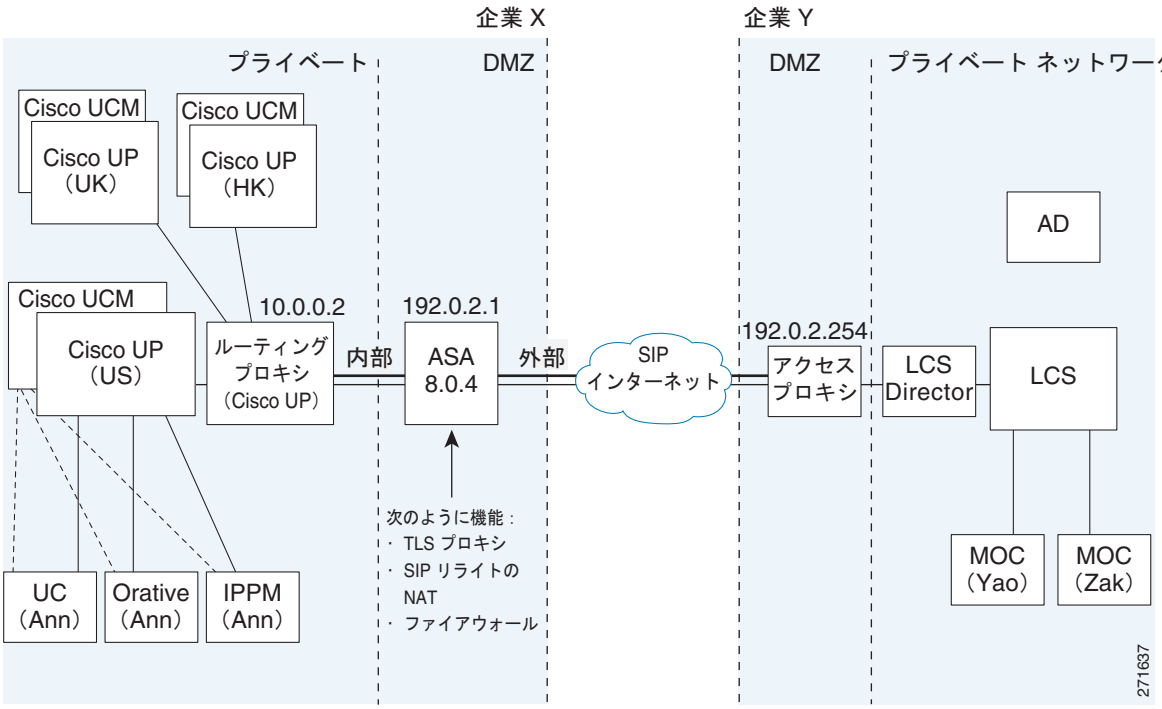
信頼できる CA から証明書を取得する方法については、一般的な操作のコンフィギュレーション ガイドの “Configuring Digital Certificates” section on page 36-11 を参照してください。

ASA でエンティティ Y を認証するには、ASA にエンティティ Y の証明書に署名する CA 証明書をインストールする必要があります。

エンティティ X とエンティティ Y の TLS プロキシ インスタンスを作成する場合、TLS 接続を開始するエンティティは、「TLS クライアント」の役割に含まれます。TLS プロキシには、「クライアント」プロキシと「サーバ」プロキシの厳密な定義があるため、いずれかのエンティティが接続を開始できる場合、2 つの TLS プロキシ インスタンスを定義する必要があります。

SIP インスペクションで TLS プロキシをイネーブルにする場合は、接続を開始できる両方のエンティティに対してポリシーを定義する必要があります。

図 17-5　標準的な Cisco Unified Presence/LCS フェデレーション シナリオ




```
object network obj-10.0.0.2-01
    host 10.0.0.2
    nat (inside,outside) static 192.0.2.1 service tcp 5061 5061
object network obj-10.0.0.2-02
    host 10.0.0.2
    nat (inside,outside) static 192.0.2.1 service tcp 5062 5062
object network obj-10.0.0.2-03
    host 10.0.0.2
    nat (inside,outside) static 192.0.2.1 service udp 5070 5070
object network obj-10.0.0.3-01
    host 10.0.0.3
    nat (inside,outside) static 192.0.2.1 service tcp 5062 45062
object network obj-10.0.0.3-02
    host 10.0.0.3
    nat (inside,outside) static 192.0.2.1 service udp 5070 45070
object network obj-0.0.0.0-01
    subnet 0.0.0.0 0.0.0.0
    nat (inside,outside) dynamic 192.0.2.1
crypto key generate rsa label ent_y_proxy_key modulus 1024
    !for self-signed Entity Y proxy certificate
crypto ca trustpoint ent_y_proxy
    enrollment self
```

```
    fqdn none
    subject-name cn=Ent-Y-Proxy
    keypair ent_y_proxy_key
crypto ca enroll ent_y_proxy
crypto ca export ent_y_proxy identity-certificate
!for Entity X's self-signed certificate
crypto ca trustpoint ent_x_cert
    enrollment terminal
crypto ca authenticate ent_x_cert
Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself
    [ certificate data omitted ]
quit
!for Entity Y's CA certificate
crypto ca trustpoint ent_y_ca
    enrollment terminal
crypto ca authenticate ent_y_ca
Enter the base 64 encoded CA certificate.
End with a blank line or the word "quit" on a line by itself
MIIDRTCCAu+gAwIBAgIQKVcqP/KW74VP0NZzL+JbRTANBgkqhkiG9w0BAQUFADCB
    [ certificate data omitted ]
/7QEM8izy0EOTSErKu7Nd76jwf5e4qttkQ==
quit
!Entity X to Entity Y
tls-proxy ent_x_to_y
    server trust-point ent_y_proxy
    client trust-point ent_x_proxy
    client cipher-suite aes128-sha1 aes256-sha1 3des-sha1 null-sha1
!Entity Y to Entity X
tls-proxy ent_y_to_x
    server trust-point ent_x_proxy
    client trust-point ent_y_proxy
    client cipher-suite aes128-sha1 aes256-sha1 3des-sha1 null-sha1
access-list ent_x_to_y extended permit tcp host 10.0.0.2 host 192.0.2.254 eq 5061
access-list ent_y_to_x extended permit tcp host 192.0.2.254 host 192.0.2.1 eq 5061
class-map ent_x_to_y
    match access-list ent_x_to_y
class-map ent_y_to_x
    match access-list ent_y_to_x
policy-map type inspect sip sip_inspect
    parameters
        !SIP inspection parameters
policy-map global_policy
    class ent_x_to_y
        inspect sip sip_inspect tls-proxy ent_x_to_y
    class ent_y_to_x
        inspect sip sip_inspect tls-proxy ent_y_to_x
service-policy global_policy global
```

XMPP フェデレーション用の ACL の設定例

例 1 : この ACL の設定例では、ポート 5269 上で任意のアドレスから任意のアドレスへの転送が許可されます。

```
access-list ALLOW-ALL extended permit tcp any any eq 5269
```

例 2 : この ACL の設定例では、ポート 5269 上で任意のアドレスから任意のシングル XMPP フェデレーション ノードへの転送が許可されます。この例では、次の値が使用されます。

- XMPP フェデレーションの Cisco Unified Presence Release 8.0 のプライベート IP アドレス = 1.1.1.1

- XMPP フェデレーションのリスニング ポート = 5269

```
access-list ALLOW-ALL extended permit tcp any host 1.1.1.1 eq 5269
```

例 3 : この ACL の設定例では、任意のアドレスから、DNS で公開された特定の XMPP フェデレーション ノードへの転送が許可されます。

（注） これらのパブリック アドレスは DNS で公開されますが、access-list コマンドにはプライベート アドレスが設定されます。

この設定例では、次の値が使用されます。

• XMPP フェデレーションの Cisco Unified Presence Release 8.0 のプライベート IP アドレス = 1.1.1.1

• 2 つ目の Cisco Unified Presence Release 8.0 のプライベート IP アドレス = 2.2.2.2

• 3 つ目の Cisco Unified Presence Release 7.x のプライベート IP アドレス = 3.3.3.3

• XMPP フェデレーションのリスニング ポート = 5269

```
access-list ALLOW-ALL extended permit tcp any host 1.1.1.1 eq 5269
access-list ALLOW-ALL extended permit tcp any host 2.2.2.2 eq 5269
access-list ALLOW-ALL extended permit tcp any host 3.3.3.3 eq 5269
```

例 4 : この ACL の設定例では、特定のフェデレーション ドメイン インターフェイスから、DNS で公開された特定の XMPP フェデレーション ノードへの転送だけが許可されます。

（注） これらのパブリック アドレスは DNS で公開されますが、access-list コマンドにはプライベート アドレスが設定されます。

この設定例では、次の値が使用されます。

- XMPP フェデレーションの Cisco Unified Presence Release 8.0 のプライベート IP アドレス = 1.1.1.1
- 2 つ目の Cisco Unified Presence Release 8.0 のプライベート IP アドレス = 2.2.2.2
- 3 つ目の Cisco Unified Presence Release 7.x のプライベート IP アドレス = 3.3.3.3
- XMPP フェデレーションのリスニング ポート = 5269
- 外部の XMPP 企業の外部インターフェイス = 100.100.100.100

```
access-list ALLOW-ALL extended permit tcp host 100.100.100.100 host 1.1.1.1 eq 5269
access-list ALLOW-ALL extended permit tcp host 100.100.100.100 host 2.2.2.2 eq 5269
access-list ALLOW-ALL extended permit tcp host 100.100.100.100 host 3.3.3.3 eq 5269
```

XMPP フェデレーション用の NAT の設定例

例 1 : XMPP フェデレーションがイネーブルのシングル ノード

この設定例では、次の値が使用されます。

- Cisco Unified Presence のパブリック IP アドレス = 10.10.10.10
- XMPP フェデレーションの Cisco Unified Presence Release 8.0 のプライベート IP アドレス = 1.1.1.1

- XMPP フェデレーションのリスニング ポート = 5269

```
nat (inside,outside) source static obj_host_1.1.1.1 obj_host_10.10.10.10 service
obj_udp_source_eq_5269 obj_udp_source_eq_5269
nat (inside,outside) source static obj_host_1.1.1.1 obj_host_10.10.10.10 service
obj_tcp_source_eq_5269 obj_tcp_source_eq_5269
```

例 2 : XMPP フェデレーションが設定され、それぞれが DNS 内のパブリック IP アドレスを持つ複数のノード

この設定例では、次の値が使用されます。

- Cisco Unified Presence のパブリック IP アドレス = 10.10.10.10、20.20.20.20、30.30.30.30
- XMPP フェデレーションの Cisco Unified Presence Release 8.0 のプライベート IP アドレス = 1.1.1.1
- 2 つ目の Cisco Unified Presence Release 8.0 のプライベート IP アドレス = 2.2.2.2
- 3 つ目の Cisco Unified Presence Release 7.x のプライベート IP アドレス = 3.3.3.3
- XMPP フェデレーションのリスニング ポート = 5269

```
nat (inside,outside) source static obj_host_1.1.1.1 obj_host_10.10.10.10 service
obj_udp_source_eq_5269 obj_udp_source_eq_5269
nat (inside,outside) source static obj_host_1.1.1.1 obj_host_10.10.10.10 service
obj_tcp_source_eq_5269 obj_tcp_source_eq_5269

nat (inside,outside) source static obj_host_2.2.2.2 obj_host_20.20.20.20 service
obj_udp_source_eq_5269 obj_udp_source_eq_5269
nat (inside,outside) source static obj_host_2.2.2.2 obj_host_20.20.20.20 service
obj_tcp_source_eq_5269 obj_tcp_source_eq_5269

nat (inside,outside) source static obj_host_3.3.3.3 obj_host_30.30.30.30 service
obj_udp_source_eq_5269 obj_udp_source_eq_5269
nat (inside,outside) source static obj_host_3.3.3.3 obj_host_30.30.30.30 service
obj_tcp_source_eq_5269 obj_tcp_source_eq_5269
```

例 3 : XMPP フェデレーションが設定されているが、DNS 内のパブリック IP アドレスは単一で、DNS で公開された任意のポートを持つ複数のノード（PAT）

この設定例では、次の値が使用されます。

- Cisco Unified Presence のパブリック IP アドレス = 10.10.10.10
- XMPP フェデレーションの Cisco Unified Presence Release 8.0 のプライベート IP アドレス = 1.1.1.1、ポート 5269
- 2 つ目の Cisco Unified Presence Release 8.0 のプライベート IP アドレス = 2.2.2.2、任意のポート 25269
- 3 つ目の Cisco Unified Presence Release 7.x のプライベート IP アドレス = 3.3.3.3、任意のポート 35269

```
nat (inside,outside) source static obj_host_1.1.1.1 obj_host_10.10.10.10 service
obj_udp_source_eq_5269 obj_udp_source_eq_5269
nat (inside,outside) source static obj_host_1.1.1.1 obj_host_10.10.10.10 service
obj_tcp_source_eq_5269 obj_tcp_source_eq_5269

nat (inside,outside) source static obj_host_2.2.2.2 obj_host_10.10.10.10 service
obj_udp_source_eq_5269 obj_udp_source_eq_25269
nat (inside,outside) source static obj_host_2.2.2.2 obj_host_10.10.10.10 service
obj_tcp_source_eq_5269 obj_tcp_source_eq_25269

nat (inside,outside) source static obj_host_3.3.3.3 obj_host_10.10.10.10 service
obj_udp_source_eq_5269 obj_udp_source_eq_35269
```

```
nat (inside,outside) source static obj_host_3.3.3.3 obj_host_10.10.10.10 service
obj_tcp_source_eq_5269 obj_tcp_source_eq_35269
```

Cisco Unified Presence の機能履歴

表 17-1 に、この機能のリリース履歴を示します。

表 17-1　Cisco Unified Presence の機能履歴

機能名	リリース	機能情報
Cisco Presence Federation Proxy	8.0(4)	Cisco Unified Presence プロキシ機能が導入されました。
Cisco Presence Federation Proxy	8.3(1)	Unified Communications Wizard が ASDM に追加されました。このウィザードを使用することにより、Cisco Presence Federation Proxy を設定できます。 XMPP フェデレーションのサポートが導入されました。