CHAPTER 15

暗号化音声インスペクションの TLS プロキシの設定

この章では、暗号化音声インスペクション機能の TLS プロキシ用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。

暗号化音声インスペクションの TLS プロキシに関する情報

エンドツーエンド暗号化では、ネットワーク セキュリティ アプライアンスがメディアやシグナリング トラフィックに対して「受信停止」になることがよくあります。これにより、アクセス コントロールや脅威回避セキュリティの機能が低下する場合があります。このように可視性が失われると、ファイアウォール機能と暗号化された音声間の相互運用性が失われ、企業では両方の主要なセキュリティ要件に対応できない状態が続く可能性があります。

ASA は、シスコ暗号化エンドポイントから Cisco Unified Communications Manager（Cisco UCM）までの暗号化されたシグナリングを代行受信して復号化し、必要な脅威回避とアクセス コントロールを適用します。また、Cisco UCM サーバへのトラフィックを再暗号化して機密性を保証することもできます。

通常、ASA の TLS プロキシ機能は、構内の統合された通信ネットワークに展開されます。このソリューションは、エンドツーエンド暗号化とファイアウォールを利用して Unified Communications Manager サーバを保護する構成に最も適しています。

統合された通信の暗号化されたシグナリングの復号化と検査

暗号化音声インスペクションを使用すると、セキュリティ アプライアンスは、音声シグナリング トラフィックの復号化、検査、変更（必要に応じて NAT フィックスアップの実行など）、および再暗号化を行う一方で、Skinny および SIP プロトコルに対する既存の VoIP インスペクション機能はすべて維持します。音声シグナリングが復号化されると、プレーンテキストのシグナリング メッセージが既存のインスペクション エンジンに渡されます。

セキュリティ アプライアンスは、Cisco IP Phone と Cisco UCM の間の TLS プロキシとして機能します。プロキシは、電話と Cisco UCM の間の音声通話に対しては透過的です。Cisco IP Phone は、登録の前に Cisco UCM から Certificate Trust List（CTL; 証明書信頼リスト）をダウンロードします。CTL には、TFTP サーバや Cisco UCM サーバなど、電話が信頼すべきデバイスの ID（証明書）が含まれています。サーバ プロキシをサポートするには、CTL ファイルに、セキュリティ アプライアンスが Cisco UCM 用に作成した証明書が含まれている必要があります。セキュリティ アプライアンスが Cisco IP Phone に代わってコールをプロキシするには、セキュリティ アプライアンス上にある、認証局によって発行され、Cisco UCM が確認可能な証明書（電話のローカル ダイナミック証明書）を提示する必要があります。

TLS プロキシは、Cisco Unified CallManager Release 5.1 以降でサポートされています。ユーザは Cisco UCM のセキュリティ機能について詳しく知っておく必要があります。Cisco UCM のセキュリティの背景と詳細な説明については、次の Cisco Unified CallManager のマニュアルを参照してください。

http://www.cisco.com/univercd/cc/td/doc/product/voice/c_callmg/5_0/sec_vir/ae/sec504/index.htm

TLS プロキシは、暗号化レイヤに適用されるので、アプリケーション レイヤ プロトコル インスペクションを設定する必要があります。ユーザは ASA のインスペクション機能、特に Skinny インスペクションと SIP インスペクションについて詳しく知っておく必要があります。

TLS プロキシでサポートされる Cisco UCM および IP Phone

Cisco Unified Communications Manager

次のリリースの Cisco Unified Communications Manager が TLS プロキシでサポートされています。

- Cisco Unified CallManager バージョン 4.*x*
- Cisco Unified CallManager バージョン 5.0
- Cisco Unified CallManager バージョン 5.1
- Cisco Unified Communications Manager 6.1
- Cisco Unified Communications Manager 7.0
- Cisco Unified Communications Manager 8.0

Cisco Unified IP Phone

Cisco Unified IP Phones 7900 シリーズの次の IP Phone が TLS プロキシでサポートされています。

- Cisco Unified IP Phone 7985
- Cisco Unified IP Phone 7975
- Cisco Unified IP Phone 7971
- Cisco Unified IP Phone 7970
- Cisco Unified IP Phone 7965
- Cisco Unified IP Phone 7962

- Cisco Unified IP Phone 7961
- Cisco Unified IP Phone 7961G-GE
- Cisco Unified IP Phone 7960
- Cisco Unified IP Phone 7945
- Cisco Unified IP Phone 7942
- Cisco Unified IP Phone 7941
- Cisco Unified IP Phone 7941G-GE
- Cisco Unified IP Phone 7940
- Cisco Unified Wireless IP Phone 7921
- Cisco Unified Wireless IP Phone 7925
- Cisco IP Communicator（CIPC）ソフトウェア電話

CTL クライアントの概要

Cisco Unified CallManager Release 5.1 以降で提供される CTL クライアント アプリケーションは、CTL ファイルで TLS プロキシ サーバ（ファイアウォール）をサポートします。図 15-1 から図 15-4 は、CTL クライアントでサポートされる TLS プロキシ機能を示しています。

図 15-1　　CTL クライアントの TLS プロキシ機能：ファイアウォールの追加

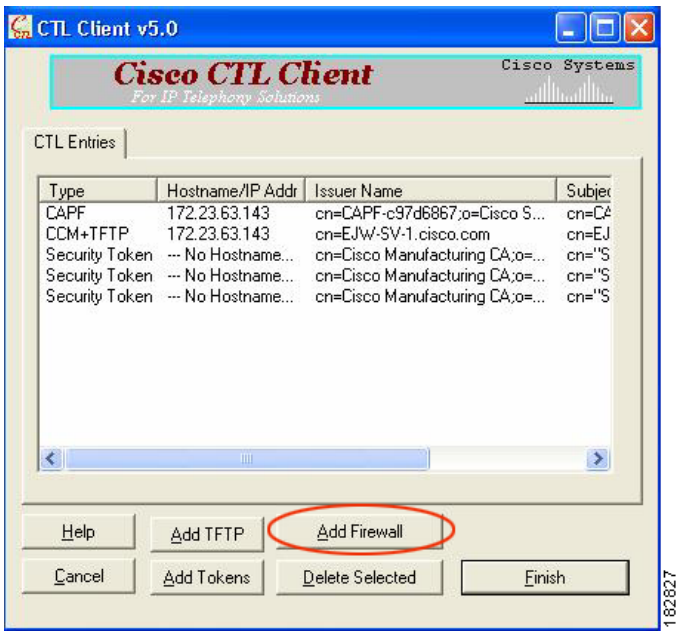


図 15-1 のように、TLS プロキシとしてのセキュリティ アプライアンスで構成される CTL エントリを追加できます。

図 15-2　CTL クライアントの TLS プロキシ機能：ASA の IP アドレスまたはドメイン名

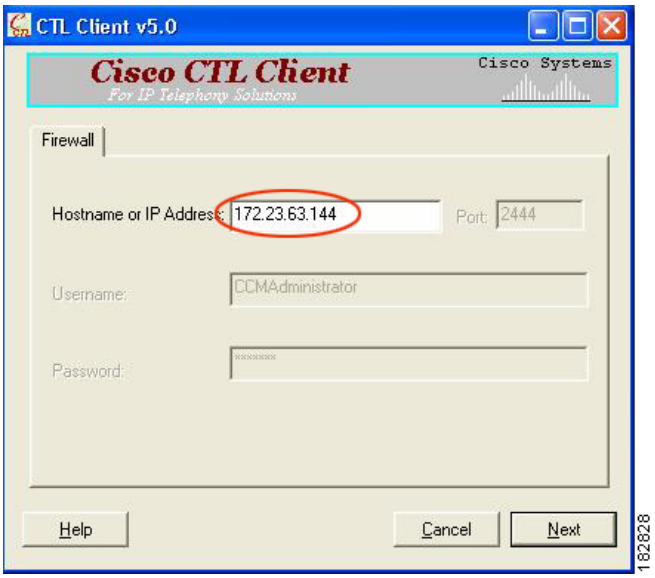


図 15-2 のように、CTL クライアントにセキュリティ アプライアンスの IP アドレスまたはドメイン名を入力できます。

図 15-3　CTL クライアントの TLS プロキシ機能：ASA の CTL エントリ

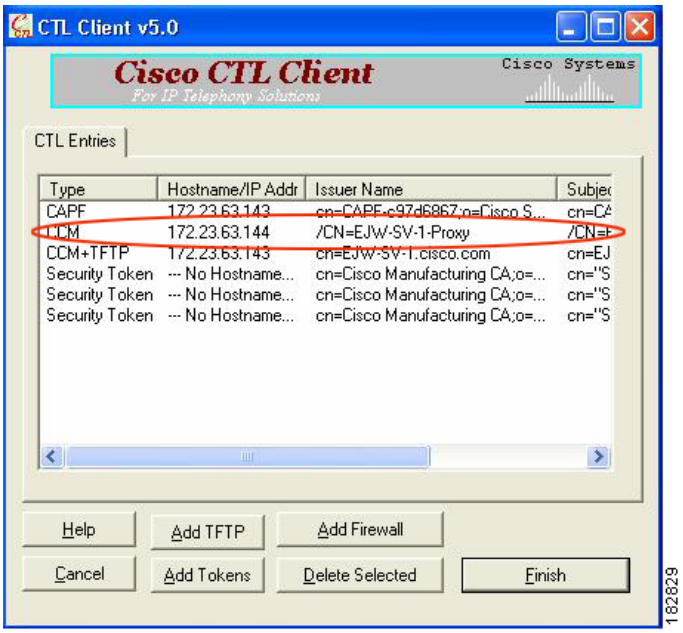
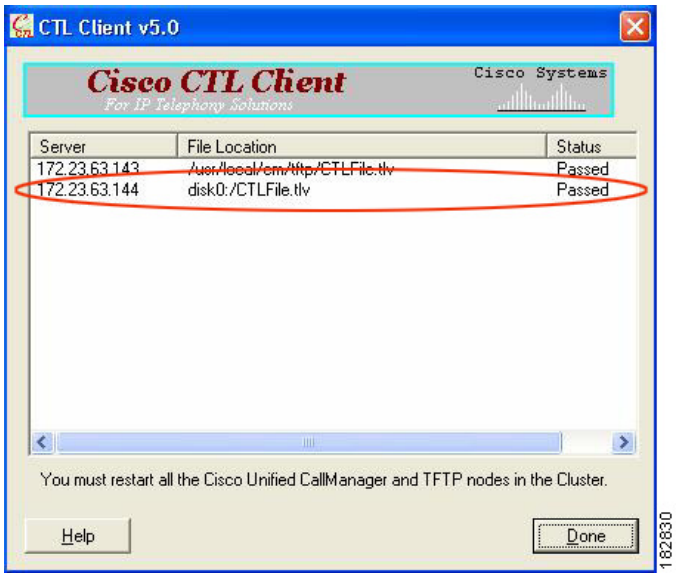


図 15-3 は、TLS プロキシとしてのセキュリティ アプライアンスの CTL エントリが追加されたことを示しています。CTL エントリは、CTL クライアントがセキュリティ アプライアンスの CTL プロバイダー サービスに接続してプロキシ証明書を取得した後に追加されます。

図 15-4　CTL クライアントの TLS プロキシ機能：ASA にインストールされている CTL ファイル



セキュリティ アプライアンスでは、CTL ファイルをそのままフラッシュ メモリに保存することはなく、CTL ファイルを解析して適切なトラストポイントをインストールします。図 15-4 は、インストールが正常に完了したことを示しています。

TLS プロキシのライセンス

ASA でサポートされる暗号化音声インスペクション機能の TLS プロキシには、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。*

モデル	ライセンス要件[1]
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。*
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。

表 15-1 に、TLS セッションのデフォルト数と最大数の詳細をプラットフォーム別に示します。

表 15-1　セキュリティ アプライアンス上での TLS セッションのデフォルト数と最大数

セキュリティ アプライアンス プラットフォーム	TLS セッションのデフォルト数	TLS セッションの最大数
ASA 5505	10	80
ASA 5510	100	200
ASA 5520	300	1200
ASA 5540	1000	4500
ASA 5550	2000	4500
ASA 5580	4000	13,000

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。

暗号化音声インスペクションの TLS プロキシの前提条件

TLS プロキシを設定する前に、次の前提条件を満たす必要があります。

- TLS プロキシを設定する前に、セキュリティ アプライアンスの時刻を設定する必要があります。手動で時刻を設定し、時刻を表示するには、**clock set** コマンドと **show clock** コマンドを使用します。セキュリティ アプライアンスでは Cisco Unified CallManager クラスタと同じ NTP サーバを使用することをお勧めします。セキュリティ アプライアンスと Cisco Unified CallManager サーバの間で時刻が同期していないと、証明書確認が失敗し、その結果、TLS ハンドシェイクが失敗する場合があります。
- Cisco Unified CallManager と相互運用するには、3DES-AES ライセンスが必要です。AES は、Cisco Unified CallManager と Cisco IP Phone で使用されるデフォルトの暗号です。
- Cisco UCM に保存されている次の証明書をインポートします。ASA で電話プロキシを使用するには、これらの証明書が必要です。
  - Cisco_Manufacturing_CA
  - CAP-RTP-001
  - CAP-RTP-002
  - CAPF 証明書（任意）

    LSC プロビジョニングが必要な場合や、IP 電話の LSC がイネーブルになっている場合は、Cisco UCM から CAPF 証明書をインポートする必要があります。Cisco UCM に CAPF 証明書が複数ある場合は、それらのすべてを ASA にインポートする必要があります。

  第 14 章「Cisco 電話プロキシの設定」を参照してください。たとえば、電話プロキシで IP 電話の証明書を検証するには、CA 製造業者証明書が必要です。

暗号化音声インスペクションの TLS プロキシの設定

この項では、次のトピックについて取り上げます。

暗号化音声インスペクションの TLS プロキシの設定のタスク フロー

セキュリティ アプライアンスに TLS プロキシを設定するには、次の手順を実行します。

ステップ 1　（任意）次のコマンドを使用してセキュリティ アプライアンスがサポートする最大 TLS プロキシ セッション数を設定します。次に例を示します。

```
hostname(config)# tls-proxy maximum-sessions 1200
```

（注）　**tls-proxy maximum-sessions** コマンドは、TLS プロキシのような暗号化アプリケーション用に予約されているメモリ サイズを制御します。暗号化メモリは、システムのブート時に予約されます。設定した最大セッション数がその時点で予約されているものよりも大きい場合、コンフィギュレーションを有効にするためにセキュリティ アプライアンスのリブートが必要になることもあります。

ステップ 2　トラストポイントを作成し、暗号化音声インスペクションの TLS プロキシの証明書を生成します。「トラストポイントの作成と証明書の生成」（P.15-10）を参照してください。

ステップ 3　Cisco IP Phone 用の LDC に署名する内部 CA を作成します。「内部 CA の作成」（P.15-11）を参照してください。

ステップ 4　CTL プロバイダー インスタンスを作成します。「CTL プロバイダー インスタンスの作成」（P.15-12）を参照してください。

ステップ 5　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.15-13）を参照してください。

ステップ 6　TLS プロキシを SIP インスペクションと Skinny インスペクションでイネーブルにします。「Skinny または SIP インスペクションでの TLS プロキシ インスタンスのイネーブル化」（P.15-14）を参照してください。

ステップ 7　ローカル CA 証明書（ldc_server）をエクスポートし、信頼できる証明書として Cisco UCM サーバにインストールします。

a.　**proxy-ldc-issuer** で指定されたトラストポイントがダイナミック証明書の署名者として使用される場合、次のコマンドを使用して証明書をエクスポートします。次に例を示します。

```
hostname(config)# crypto ca export ldc_server identity-certificate
```

b.　埋め込みローカル CA サーバ LOCAL-CA-SERVER の場合、次のコマンドを使用して証明書をエクスポートします。次に例を示します。

```
hostname(config)# show crypto ca server certificate
```

出力をファイルに保存し、Cisco UCM に証明書をインポートします。詳細については、Cisco Unified CallManager のマニュアル（http://www.cisco.com/univercd/cc/td/doc/product/voice/c_callmg/5_0/iptp_adm/504/iptpch6.htm#wp1040848）を参照してください。

この手順の後、次の Cisco Unified CallManager GUI の Display Certificates 機能を使用して、インストールされた証明書を確認することもできます。

http://www.cisco.com/univercd/cc/td/doc/product/voice/c_callmg/5_0/iptp_adm/504/iptpch6.htm#wp1040354

ステップ 8　CTL クライアント アプリケーションを実行してサーバ プロキシ証明書（ccm_proxy）を CTL ファイルに追加し、その CTL ファイルをセキュリティ アプライアンスにインストールします。CTL クライアントの設定方法と使用方法については、次の Cisco Unified CallManager のマニュアルを参照してください。

http://www.cisco.com/univercd/cc/td/doc/product/voice/c_callmg/5_1/nci/p08/secuauth.htm

（注）　セキュリティ アプライアンスと相互運用するためには、Cisco Unified CallManager Release 5.1 と一緒にリリースされた CTL クライアントが必要です。TLS プロキシ サポートの詳細については、「CTL クライアントの概要」（P.15-3）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

Cisco UCM プロキシ証明書は、自己署名の場合と、サードパーティ CA 発行の場合があります。証明書は CTL クライアントにエクスポートされます。

前提条件

Cisco UCM に保存されている、必要な証明書をインポートします。「Cisco UCM の証明書」（P.14-7）および「Cisco UCM からの証明書のインポート」（P.14-17）を参照してください。

	コマンド	目的
ステップ 1	hostname(config)# **crypto key generate rsa label** *key-pair-label* **modulus** *size* **Examples:** hostname(config)# **crypto key generate rsa label ccm_proxy_key modulus 1024** hostname(config)# **crypto key generate rsa label ldc_signer_key modulus 1024** hostname(config)# **crypto key generate rsa label phone_common modulus 1024**	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。 （注）　役割ごとに異なるキー ペアを作成することをお勧めします。
ステップ 2	hostname(config)# **crypto ca trustpoint** *trustpoint_name* **Example:** hostname(config)# **! for self-signed CCM proxy certificate** hostname(config)# **crypto ca trustpoint ccm_proxy**	Cisco UMA サーバ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。

	コマンド	目的
ステップ 5	hostname(config-ca-trustpoint)# **subject-name** *X.500_name* **Example:** hostname(config-ca-trustpoint)# **subject-name cn=EJW-SV-1-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。 Cisco IP Phone では、CTL ファイルを参照して証明書を確認するために、X.509v3 証明書に特定のフィールドが存在している必要があります。そのため、プロキシ証明書トラストポイントの **subject-name** エントリを設定する必要があります。このサブジェクト名は、CN、OU、O の各フィールドを順に連結して構成する必要があります。CN フィールドは必須で、それ以外はオプションです。 **（注）** 連結したフィールド（存在する場合）はそれぞれセミコロンで区切られ、次のいずれかの形式になります。 CN=xxx;OU=yyy;O=zzz CN=xxx;OU=yyy CN=xxx;O=zzz CN=xxx
ステップ 6	hostname(config-ca-trustpoint)# **keypair** *keyname* **Example:** hostname(config-ca-trustpoint)# **keypair ccm_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 ccm_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

トランスポイントを作成し、証明書を生成したら、Cisco IP Phone の LDC に署名する内部 CA を作成します。「内部 CA の作成」（P.15-11）を参照してください。

内部 CA の作成

Cisco IP Phone の LDC に署名する内部ローカル CA を作成します。

このローカル CA は、**proxy-ldc-issuer** がイネーブルな標準の自己署名トラストポイントとして作成されます。LDC を発行するには、ASA 上の埋め込みローカル CA である LOCAL-CA-SERVER を使用できます。

	コマンド	目的
ステップ 1	hostname(config)# **crypto ca trustpoint** *trustpoint_name* **Example:** hostname(config)# **! for the internal local LDC issuer** hostname(config)# **crypto ca trustpoint ldc_server**	LDC 発行元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。
ステップ 2	hostname(config-ca-trustpoint)# **enrollment self**	自己署名した証明書を生成します。

	コマンド	目的
ステップ 3	hostname(config-ca-trustpoint)# **proxy-ldc-issuer**	TLS プロキシのローカル ダイナミック証明書を発行します。**proxy-ldc-issuer** コマンドは、暗号トラストポイントに、LDC を発行するためのローカル CA としての役割を付与します。このコマンドには、Crypto ca トラストポイント コンフィギュレーション モードからアクセスできます。 **proxy-ldc-issuer** コマンドは、TLS プロキシのダイナミック証明書を発行するトラストポイントに、ローカル CA の役割を定義します。このコマンドは、「自己登録」を使用するトラストポイントでのみ設定できます。
ステップ 4	hostname(config-ca-trustpoint)# **fqdn** *fqdn* **Example:** hostname(config-ca-trustpoint)# **fqdn my-ldc-ca.exmaple.com**	登録時に、指定した FQDN を証明書の Subject Alternative Name 拡張子に含めます。
ステップ 5	hostname(config-ca-trustpoint)# **subject-name** *X.500_name* **Example:** hostname(config-ca-trustpoint)# **subject-name cn=FW_LDC_SIGNER_172_23_45_200**	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 ldc_signer_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 ldc_server**	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

内部 CA を作成したら、CTL プロバイダー インスタンスを作成します。「CTL プロバイダー インスタンスの作成」（P.15-12）を参照してください。

CTL プロバイダー インスタンスの作成

CTL クライアントからの接続に備えて CTL プロバイダー インスタンスを作成します。

CTL プロバイダーのデフォルトの受信ポート番号は TCP 2444 です。これは、Cisco UCM のデフォルトの CTL ポートです。Cisco UCM クラスタが異なるポートを使用している場合は、**service port** コマンドを使用してポート番号を変更します。

	コマンド	目的
ステップ 1	hostname(config)# **ctl-provider** *ctl_name* **Example:** hostname(config)# **ctl-provider my_ctl**	証明書信頼リスト プロバイダー インスタンスを作成するには、CTL プロバイダー コンフィギュレーション モードに入ります。
ステップ 2	hostname(config-ctl-provider)# **client interface** *if_name ipv4_addr* **Example:** hostname(config-ctl-provider)# **client interface inside address 172.23.45.1**	証明書信頼リスト プロバイダーに接続できるクライアントを指定します。 **interface** *if_name* には、接続できるインターフェイスを指定し、*ipv4_addr* には、クライアントの IP アドレスを指定します。 複数のコマンドを発行して、複数のクライアントを定義できます。
ステップ 3	hostname(config-ctl-provider)# **client username** *user_name* **password** *password* **encrypted** **Example:** hostname(config-ctl-provider)# **client username CCMAdministrator password XXXXXX encrypted**	クライアント認証のユーザ名とパスワードを指定します。 このユーザ名とパスワードは、Cisco UCM の管理用ユーザ名とパスワードと同じでなければなりません。
ステップ 4	hostname(config-ctl-provider)# **export certificate** *trustpoint_name* **Example:** hostname(config-ctl-provider)# **export certificate**	クライアントにエクスポートする証明書を指定します。証明書は、CTL クライアントで構成された証明書信頼リスト ファイルに追加されます。 **export** コマンドのトラストポイント名は、Cisco UCM サーバのプロキシ証明書です。
ステップ 5	hostname(config-ctl-provider)# **ctl install**	CTL プロバイダーで CTL クライアントから CTL ファイルを解析し、CTL ファイルからのエントリのトラストポイントをインストールできるようにします。このコマンドでインストールされたトラストポイントには、"_internal_CTL_<ctl_name>" のプレフィックスが付いた名前が設定されます。

次の作業

CTL プロバイダー インスタンスを作成したら、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.15-13）を参照してください。

TLS プロキシ インスタンスの作成

暗号化されたシグナリングを処理するための TLS プロキシ インスタンスを作成します。

	コマンド	目的
ステップ 1	hostname(config)# **tls-proxy** *proxy_name* **Example:** hostname(config)# tls-proxy my_proxy	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_trustpoint* **Example:** hostname(config-tlsp)# server trust-point ccm_proxy	TLS ハンドシェイク中に提示するプロキシ トラストポイント証明書を指定します。 **server** コマンドでは、元の TLS サーバのプロキシ パラメータを設定します。つまり、TLS ハンドシェイク時、または元の TLS クライアントと対話するときに、ASA がサーバとして機能できるようにするパラメータです。
ステップ 3	hostname(config-tlsp)# **client ldc issuer** *ca_tp_name* **Example:** hostname(config-tlsp)# client ldc issuer ldc_server	ローカル ダイナミック証明書の発行元を設定します。クライアントのダイナミック証明書を発行するローカル CA は、**crypto ca trustpoint** コマンドで定義され、トラストポイントでは **proxy-ldc-issuer** を設定するか、デフォルトのローカル CA サーバ（LOCAL-CA-SERVER）を使用する必要があります。 **ldc issuer** *ca_tp_name* には、クライアントのダイナミック証明書を発行するローカル CA トラストポイントを指定します。
ステップ 4	hostname(config-tlsp)# **client ldc key-pair** *key_label* **Example:** hostname(config-tlsp)# client ldc key-pair phone_common	キー ペアを設定します。 キー ペア値は、crypto key generate コマンドを使用して生成されている必要があります。
ステップ 5	hostname(config-tlsp)# **client cipher-suite** *cipher_suite* **Example:** hostname(config-tlsp)# client cipher-suite aes128-sha1 aes256-sha1	ユーザ定義の暗号スイートを設定します。 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、または ssl encryption コマンドで定義された暗号スイートが置き換えられます。このコマンドでは、2 つの TLS セッション間で異なる暗号を設定できます。CallManager サーバでは、AES 暗号を使用する必要があります。

次の作業

TLS プロキシ インスタンスを作成したら、Skinny および SIP インスペクションで TLS プロキシ インスタンスをイネーブルにします。「Skinny または SIP インスペクションでの TLS プロキシ インスタンスのイネーブル化」（P.15-14）を参照してください。

Skinny または SIP インスペクションでの TLS プロキシ インスタンスのイネーブル化

Skinny または SIP インスペクションで、Cisco IP Phone および Cisco UCM の TLS プロキシをイネーブルにします。次の手順は、Skinny インスペクションで TLS プロキシ インスタンスをイネーブルにする方法を示しています。

	コマンド	目的
ステップ 1	hostname(config)# **class-map** *class_map_name* **Example:** hostname(config)# **class-map sec_skinny**	検査するセキュア Skinny クラスのトラフィックを設定します。 *class_map_name* には、Skinny クラス マップの名前を指定します。
ステップ 2	hostname(config-cmap)# **match port tcp eq 2443**	セキュア Skinny インスペクションのアクションを適用する TCP ポート 2443 に照合します。
ステップ 3	hostname(config-cmap)# **exit**	
ステップ 4	hostname(config)# **policy-map type inspect skinny** *policy_map_name* **Example:** hostname(config)# **policy-map type inspect skinny skinny_inspect**	Skinny 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 5	hostname(config-pmap)# **parameters** hostname(config-pmap-p)# **!** Skinny inspection parameters	Skinny 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 6	hostname(config-pmap-p)# **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 7	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 8	hostname(config-pmap)# **class inspection_default**	デフォルトのクラス マップを指定します。 コンフィギュレーションには、デフォルト グローバル ポリシーで ASA が使用するデフォルトのレイヤ 3/4 クラス マップが含まれます。これは **inspection_default** と呼ばれ、デフォルト インスペクション トラフィックと一致します。
ステップ 9	hostname(config-pmap-c)# **inspect skinny** *skinny_map* **Example:** hostname(config-pmap-c)# **inspect skinny skinny_inspect**	SCCP（Skinny）アプリケーション インスペクションをイネーブルにします。
ステップ 10	hostname(config-pmap)# **class** *classmap_name* **Example:** hostname(config-pmap)# **class sec_skinny**	アクションをクラス マップ トラフィックに割り当てることができるポリシー マップにクラス マップを割り当てます。
ステップ 11	hostname(config-pmap-c)# **inspect skinny** *skinny_map* **tls-proxy** *proxy_name* **Example:** hostname(config-pmap-c)# **inspect skinny skinny_inspect tls-proxy my_proxy**	指定されたインスペクション 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 12	hostname(config-pmap-c)# **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 13	hostname(config)# **service-policy** *policymap_name* **global** **Example:** hostname(config)# **service-policy global_policy global**	すべてのインターフェイスでサービス ポリシーをイネーブルにします。

TLS プロキシのモニタリング

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
```

次の出力例では、SIP 電話用に TLS プロキシ セッションのセットアップが完了したことが示されています。

```
hostname(config)# show log

Apr 17 2007 23:13:47: %ASA-6-725001: Starting SSL handshake with client
outside:133.9.0.218/49159 for TLSv1 session.
Apr 17 2007 23:13:47: %ASA-7-711001: TLSP cbad5120: Set up proxy for Client
outside:133.9.0.218/49159 <-> Server inside:195.168.2.201/5061
Apr 17 2007 23:13:47: %ASA-7-711001: TLSP cbad5120: Using trust point 'local_ccm' with the
Client, RT proxy cbae1538
Apr 17 2007 23:13:47: %ASA-7-711001: TLSP cbad5120: Waiting for SSL handshake from Client
outside:133.9.0.218/49159.
Apr 17 2007 23:13:47: %ASA-7-725010: Device supports the following 4 cipher(s).
Apr 17 2007 23:13:47: %ASA-7-725011: Cipher[1] : RC4-SHA
Apr 17 2007 23:13:47: %ASA-7-725011: Cipher[2] : AES128-SHA
Apr 17 2007 23:13:47: %ASA-7-725011: Cipher[3] : AES256-SHA
Apr 17 2007 23:13:47: %ASA-7-725011: Cipher[4] : DES-CBC3-SHA
Apr 17 2007 23:13:47: %ASA-7-725008: SSL client outside:133.9.0.218/49159 proposes the
following 2 cipher(s).
Apr 17 2007 23:13:47: %ASA-7-725011: Cipher[1] : AES256-SHA
Apr 17 2007 23:13:47: %ASA-7-725011: Cipher[2] : AES128-SHA
Apr 17 2007 23:13:47: %ASA-7-725012: Device chooses cipher : AES128-SHA for the SSL
session with client outside:133.9.0.218/49159
Apr 17 2007 23:13:47: %ASA-7-725014: SSL lib error.Function: SSL23_READ Reason: ssl
handshake failure
Apr 17 2007 23:13:47: %ASA-7-717025: Validating certificate chain containing 1
certificate(s).
Apr 17 2007 23:13:47: %ASA-7-717029: Identified client certificate within certificate
chain.serial number: 01, subject name: cn=SEP0017593F50A8.
Apr 17 2007 23:13:47: %ASA-7-717030: Found a suitable trustpoint
_internal_ejw-sv-2_cn=CAPF-08a91c01 to validate certificate.
Apr 17 2007 23:13:47: %ASA-6-717022: Certificate was successfully validated.serial number:
01, subject name:  cn=SEP0017593F50A8.
Apr 17 2007 23:13:47: %ASA-6-717028: Certificate chain was successfully validated with
warning, revocation status was not checked.
Apr 17 2007 23:13:47: %ASA-6-725002: Device completed SSL handshake with client
outside:133.9.0.218/49159
Apr 17 2007 23:13:47: %ASA-6-725001: Starting SSL handshake with server
inside:195.168.2.201/5061 for TLSv1 session.
Apr 17 2007 23:13:47: %ASA-7-725009: Device proposes the following 2 cipher(s) to server
inside:195.168.2.201/5061
Apr 17 2007 23:13:47: %ASA-7-725011: Cipher[1] : AES128-SHA
Apr 17 2007 23:13:47: %ASA-7-725011: Cipher[2] : AES256-SHA
Apr 17 2007 23:13:47: %ASA-7-711001: TLSP cbad5120: Generating LDC for client
'cn=SEP0017593F50A8', key-pair 'phone_common', issuer 'LOCAL-CA-SERVER', RT proxy cbae1538
```

```
Apr 17 2007 23:13:47: %ASA-7-711001: TLSP cbad5120: Started SSL handshake with Server
Apr 17 2007 23:13:47: %ASA-7-711001: TLSP cbad5120: Data channel ready for the Client
Apr 17 2007 23:13:47: %ASA-7-725013: SSL Server inside:195.168.2.201/5061 choose cipher :
AES128-SHA
Apr 17 2007 23:13:47: %ASA-7-717025: Validating certificate chain containing 1
certificate(s).
Apr 17 2007 23:13:47: %ASA-7-717029: Identified client certificate within certificate
chain.serial number: 76022D3D9314743A, subject name: cn=EJW-SV-2.inside.com.
Apr 17 2007 23:13:47: %ASA-6-717022: Certificate was successfully validated.Certificate is
resident and trusted, serial number: 76022D3D9314743A, subject name:
cn=EJW-SV-2.inside.com.
Apr 17 2007 23:13:47: %ASA-6-717028: Certificate chain was successfully validated with
revocation status check.
Apr 17 2007 23:13:47: %ASA-6-725002: Device completed SSL handshake with server
inside:195.168.2.201/5061
Apr 17 2007 23:13:47: %ASA-7-711001: TLSP cbad5120: Data channel ready for the Server
```

アクティブな TLS プロキシ セッションをチェックするには、**show tls-proxy** コマンドにさまざまなオプションを指定して使用します。以下にいくつかの出力例を示します。

```
hostname(config-tlsp)# show tls-proxy
Maximum number of sessions: 1200

TLS-Proxy 'sip_proxy': ref_cnt 1, seq# 3
    Server proxy:
        Trust-point: local_ccm
    Client proxy:
        Local dynamic certificate issuer: LOCAL-CA-SERVER
        Local dynamic certificate key-pair: phone_common
        Cipher suite:  aes128-sha1 aes256-sha1
    Run-time proxies:
         Proxy 0xcbae1538: Class-map: sip_ssl, Inspect: sip
            Active sess 1, most sess 3, byte 3456043

TLS-Proxy 'proxy': ref_cnt 1, seq# 1
    Server proxy:
        Trust-point: local_ccm
    Client proxy:
        Local dynamic certificate issuer: ldc_signer
        Local dynamic certificate key-pair: phone_common
        Cipher-suite: <unconfigured>
    Run-time proxies:
        Proxy 0xcbadf720: Class-map: skinny_ssl, Inspect: skinny
            Active sess 1, most sess 1, byte 42916

hostname(config-tlsp)# show tls-proxy session count
2 in use, 4 most used

hostname(config-tlsp)# show tls-proxy session
2 in use, 4 most used
outside 133.9.0.211:50437 inside 195.168.2.200:2443 P:0xcbadf720(proxy) S:0xcbc48a08 byte
42940
outside 133.9.0.218:49159 inside 195.168.2.201:5061 P:0xcbae1538(sip_proxy) S:0xcbad5120
byte 8786

hostname(config-tlsp)# show tls-proxy session detail
2 in use, 4 most used
outside 133.9.0.211:50437 inside 195.168.2.200:2443 P:0xcbadf720(proxy) S:0xcbc48a08 byte
42940
    Client: State SSLOK  Cipher AES128-SHA Ch 0xca55e498 TxQSize 0 LastTxLeft 0 Flags 0x1
    Server: State SSLOK  Cipher AES128-SHA Ch 0xca55e478 TxQSize 0 LastTxLeft 0 Flags 0x9
Local Dynamic Certificate
    Status: Available
    Certificate Serial Number: 29
```

```
    Certificate Usage: General Purpose
    Public Key Type: RSA (1024 bits)
    Issuer Name:
        cn=TLS-Proxy-Signer
    Subject Name:
        cn=SEP0002B9EB0AAD
        o=Cisco Systems Inc
        c=US
    Validity Date:
        start date: 09:25:41 PDT Apr 16 2007
        end   date: 09:25:41 PDT Apr 15 2008
    Associated Trustpoints:

outside 133.9.0.218:49159 inside 195.168.2.201:5061 P:0xcbae1538(sip_proxy) S:0xcbad5120
byte 8786
    Client: State SSLOK  Cipher AES128-SHA Ch 0xca55e398 TxQSize 0 LastTxLeft 0 Flags 0x1
    Server: State SSLOK  Cipher AES128-SHA Ch 0xca55e378 TxQSize 0 LastTxLeft 0 Flags 0x9
Local Dynamic Certificate
    Status: Available
    Certificate Serial Number: 2b
    Certificate Usage: General Purpose
    Public Key Type: RSA (1024 bits)
    Issuer Name:
        cn=F1-ASA.default.domain.invalid
    Subject Name:
        cn=SEP0017593F50A8
    Validity Date:
        start date: 23:13:47 PDT Apr 16 2007
        end   date: 23:13:47 PDT Apr 15 2008
    Associated Trustpoints:
```

暗号化音声インスペクションの TLS プロキシの機能履歴

表 15-2 に、この機能のリリース履歴を示します。

表 15-2　Cisco 電話プロキシの機能履歴

機能名	リリース	機能情報
TLS プロキシ	8.0(2)	TLS プロキシ機能が導入されました。