HTTPS アクセス

この章は、次の項で構成されています。

- 概要 （1 ページ）
- カスタム証明書の設定のガイドライン （1 ページ）
- SSL 暗号設定を変更する （2 ページ）
- GUI を使用した Cisco ACI HTTPS アクセス用カスタム証明書の設定 （3 ページ）
- NX-OS CLI を使用した証明書ベースの認証の有効化 （6 ページ）

概要

この記事は、Cisco ACI を使用する際の HTTPS アクセスのカスタム証明書を設定する方法の例を示します。

カスタム証明書の設定のガイドライン

- Cisco Application Policy Infrastructure Controller（APIC）で証明書署名要求（CSR）を生成するために使用される秘密キーのエクスポートはサポートされていません。証明書の CSR を生成するために使用された秘密キーを共有することにより、「Subject Alternative Name（SAN）」フィールドのワイルドカード（「* cisco.com」など）を介して複数のサーバで同じ証明書を使用する場合は、秘密キーを Cisco Application Centric Infrastructure（ACI）ファブリックの外部に配置し、Cisco ACI ファブリックにインポートします。
- 証明書署名要求（CSR）を生成する前に、公開中間証明書とルート CA 証明書をダウンロードしてインストールする必要があります。ルート CA 証明書は技術的には CSR を生成するために必要ではありませんが、シスコでは、対象とする CA 機関と CSR への署名に使用される実物の間の不一致を防ぐために、CSR を生成する前にルート CA 証明書が必要です。Cisco APIC は、送信された証明書が設定された CA によって署名されていることを確認します。
- 更新された証明書の生成に同じ公開キーと秘密キーを使用するには、次のガイドラインを満たす必要があります。

- 元の CSR にはキー リング内の秘密キーとペアになる公開キーが含まれているため、元の CSR を維持する必要があります。
- Cisco APIC で公開キーと秘密キーを再使用する場合は、元の証明書に使用されたものと同じ CSR を更新された証明書に再送信する必要があります。
- 更新された証明書に同じ公開キーと秘密キーを使用する場合は、元のキー リングを削除しないでください。キー リングを削除すると、CSR で使用されている関連秘密キーが自動的に削除されます。

- Cisco ACI マルチサイト、VCPlugin、VRA、および SCVMM は、証明書ベースの認証ではサポートされません。
- Cisco APIC クラスタごとに 1つのSSL証明書のみが許可されます。
- 以降のリリースからリリース 4.0(1) にダウングレードする前に、証明書ベースの認証を無効にする必要があります。
- 証明書ベースの認証セッションを終了するには、ログアウトして CAC カードを削除する必要があります。
- Cisco APIC に設定されたカスタム証明書は、リーフ スイッチとスパイン スイッチに展開されます。ファブリック ノードに接続するために使用される URL または DN が **[サブジェクト（Subject）]** または **[サブジェクト代替名（Subject Alternative Name）]** フィールド内にある場合、ファブリック ノードは証明書でカバーされます。
- Cisco APIC GUI は、最大サイズが 4k バイトの証明書を受け入れることができます。

SSL 暗号設定を変更する

SSL 暗号は、有効化、無効化、または完全に削除できます。必要な暗号設定に応じて、必要な正確な組み合わせを理解する必要があります。暗号が残らない方法で暗号を無効化および有効化することは設定ミスであり、NGINX の検証に失敗します。

NGINX は OpenSSL 暗号リスト形式を使用します。形式については、OpenSSL Web サイトにアクセスしてください。

Cisco APIC SSL 設定オプションを暗号リスト形式化にマッピングする

暗号を有効にすると、その暗号が NGINX 構成ファイルに書き込まれます。暗号を無効にすると、その暗号が NGINX 構成ファイルに前に感嘆符 (!) を付けて書き込まれます。たとえば、「EEDCH」を無効にすると、「!EEDCH」と書き込まれます。暗号を削除すると、その暗号が NGINX 構成ファイルに暗号がまったく書き込まれなくなります。

（注） OpenSSL 暗号リスト形式のドキュメントには次のように記載されています。「(!) が使用されている場合、暗号はリストから完全に削除されます。削除された暗号は、明示的に指定されていても、リストに再び表示されることはありません」これにより、暗号の「有効」状態に関係なく、「無効」に設定された暗号を参照する組み合わせ暗号が削除される可能性があります。

例：「EEDCH」を無効にし、「EECDH+aRSA+SHA384」を有効にします。これにより、次がNGINX 構成ファイルに書き込まれます：「!EEDCH:EECDH+aRSA+SHA384」。「!EEDCH」は、「EECDH+aRSA+SHA384」が追加されないようにします。これにより、暗号が使用されないことでNGINX検証に失敗するため、NGINXの更新（カスタムHTTPS証明書の適用など）が成功しなくなります。

Cisco APIC SSL 設定を変更する前の暗号リスト形式のテスト

Cisco Application Policy Infrastructure Controller（APIC）に暗号の変更を加える前に、`openssl ciphers -V 'cipher_list'` コマンドを使用して、計画した暗号の組み合わせの結果を検証し、暗号出力が目的の結果と一致することを確認します。

例：

```
apic# openssl ciphers -V 'EECDH+aRSA+SHA256:EECDH+aRSA+SHA384'
          0xC0,0x27 - ECDHE-RSA-AES128-SHA256 TLSv1.2 Kx=ECDH     Au=RSA  Enc=AES(128)
 Mac=SHA256
          0xC0,0x28 - ECDHE-RSA-AES256-SHA384 TLSv1.2 Kx=ECDH     Au=RSA  Enc=AES(256)
 Mac=SHA384
```

テストした暗号リストがエラーまたは「暗号が一致しません」という結果になった場合は、この設定を Cisco APIC に適用しないでください。これを行うと、Cisco APIC GUI にアクセスできなくなったり、カスタム証明書アプリケーションが壊れたりするなど、NGINX の問題が発生する可能性があります。

例：

```
apic# openssl ciphers -V '!EECDH:EECDH+aRSA+SHA256:EECDH+aRSA+SHA384'
Error in cipher list
132809172158128:error:1410D0B9:SSL routines:SSL_CTX_set_cipher_list:no cipher
match:ssl_lib.c:1383:
```

GUI を使用した Cisco ACI HTTPS アクセス用カスタム証明書の設定

注意：ダウンタイムの可能性があるため、メンテナンス時間中にのみこのタスクを実行してください。ダウンタイムは外部ユーザまたはシステムからの Cisco Application Policy Infrastructure Controller（APIC）APIC クラスタおよびスイッチへのアクセスには影響しますが、Cisco APIC とスイッチの接続には影響しませスイッチ上の NGINX プロセスも影響を受けますが、外部接続のみでファブリックのデータプレーンには影響ありません。Cisco APIC、設定、管理、トラ

ブルシューティングなどへのアクセスは影響を受けることになります。Cisco APIC および スイッチで実行されている NGINX Web サーバーは、この操作中に再起動されます。

始める前に

適切な認証局を作成できるように、信頼できる証明書を取得する機関を決定します。

ステップ 1 メニュー バーで、**[Admin]** > **[AAA]** の順に選択します。

ステップ 2 [Navigation] ペインで、[Security] を選択します。

ステップ 3 **作業**ペインで、**[認証局（Certificate Authorities）]** > **[アクション（Actions）]** > **[認証局の作成（Create Certificate Authority）]** の順に選択します。

ステップ 4 [**認証局の作成（Create Certificate Authority**）] 画面で、[**Name（名前**）] フィールドに、認証局の名前を入力します。

ステップ 5 （オプション）認証局の [**説明（Description**）] を入力します。

ステップ 6 **[証明書チェーン（Certificate Chain）]** フィールドで、Cisco APIC の証明書署名要求（CSR）に署名する認証局の中間証明書とルート証明書をコピーします。

証明書は、Base64 エンコード X.509（CER）形式である必要があります。中間証明書はルート CA 証明書の前に配置されます。次の例のようになります。

```
-----BEGIN CERTIFICATE-----
<Intermediate Certificate>
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
<Root CA Certificate>
-----END CERTIFICATE-----
```

ステップ 7 [**保存（Save**）] をクリックします。

ステップ 8 **作業**ペインで、**[キーリング（Key Rings）]** > **[アクション（Actions）]** > **[キーリングの作成（Create Key Ring）]** の順に選択します。

キー リングを使用すると、秘密キー（外部デバイスからインポートされるか、APIC で内部的に生成される）、秘密キーによって生成される CSR、および CSR によって署名された証明書を管理できます。

ステップ 9 [Create Key Ring] ダイアログボックスで、[Name] フィールドに、名前を入力します。

ステップ 10 （オプション）キーリングの [**説明（Description**）] を入力します。

ステップ 11 [**認証局（Certificate Authority**）] フィールドで、[**認証局の選択（Select Certificate Authority**）] をクリックし、以前に作成した認証局を選択するか、[**認証局の作成（Create Certificate Authority**）] を選択します。

ステップ 12 [**秘密キー（Private Key**）] フィールドで必要なラジオボタンをクリックします。オプションは、[新しいキーの生成（Generate New Key）]、[既存キーのインポート（Import Existing Key）] です。

ステップ 13 秘密キーを入力します。このオプションは、**秘密キー**に [既存キーのインポート（Import Existing Key）] オプションを選択した場合にのみ表示されます。

キーリングから Cisco APIC を使用して CSR を生成する場合は、コンテンツを追加しないでください。

署名付き証明書と秘密キーを入力していない場合は、**[作業（Work）]**ペインの**[キー リング（Key Rings）]** 領域で、作成されたキー リングの **[管理状態（Admin State）]** に **[開始（Started）]** と表示され、CSR が生成されるのを待ちます。手順 17 に進みます。

署名付き証明書と秘密キーの両方を入力した場合は、**[キーリング（Key Rings）]** 領域に、作成されたキー リングの **[管理状態（Admin State）]** が **[完了（Completed）]** と表示されます。手順 26 に進みます。

（注） キー リングは削除しないでください。キー リングを削除すると、CSR で使用されている関連秘密キーが自動的に削除されます。

ステップ 14 キー リングで Cisco APIC を使用して CSR を生成する場合は、**[証明書（Certificate）]** フィールドにコンテンツを追加しないでください。または、Cisco APIC 外の秘密キーおよび CSR を生成して前の手順で CA によって署名されたものがある場合は、署名された証明書の内容を追加します。

ステップ 15 [**モジュラス**（**Modulus**）] フィールドで、ドロップダウンリストから目的のキーの強さを選択します。このオプションは、**秘密**キーに [新しいキーの生成（Generate New Key）] オプションを選択した場合にのみ表示されます。

ステップ 16 [**保存**（**Save**）] ([**キーリングの作成**（**Create Key Ring**）] 画面) をクリックします。

ステップ 17 **ナビゲーション**ウィンドウで、**[キーリング（Key Rings）]** > ***[キーリング名（key_ring_name）]*** の順に選択します。

ステップ 18 [Work] ペインで、**[Actions]** > **[Create Certificate Request]** の順に選択します。

ステップ 19 **[サブジェクト（Subject）]** フィールドに、CSR の共通名（CN）を入力します。

ワイルドカードを使用して Cisco APIC の完全修飾ドメイン名（FQDN）を入力できますが、最新の証明書では、通常、識別可能な証明書の名前を入力し、**[代替サブジェクト名（Alternate Subject Name）]** フィールドにすべての Cisco APIC の FQDN を入力することを推奨します（多くの最新のブラウザは SAN フィールドに FQDN を想定しているため、*SAN*（サブジェクト代替名）とも呼ばれます。

ステップ 20 **[代替サブジェクト名（Alternate Subject Name）]** フィールドに、「DNS：apic1.example.com、DNS：apic2.example.com、DNS：apic3.example.com」や「DNS：* example.com」など、すべての Cisco APIC の FQDN を入力します。 。

ステップ 21 必要に応じて、残りのフィールドに入力します。

ステップ 22 [Submit] をクリックします。

同じキー リング内に、**[関連付けられた証明書要求（Associated Certificate Request）]** 領域が表示されます。このフィールドには、**[サブジェクト（Subject）]**、**[代替サブジェクト名（Alternate Subject Name）]**、およびこのフィールドに、このキー リングに関連付けられた CSR の内容を含む新しいフィールド **[要求（Request）]** が表示されます。**[要求（Request）]** フィールドからコンテンツをコピーし、署名用にこのキー リングに関連付けられている同じ認証局にコンテンツを送信します。

ステップ 23 **ナビゲーション**ウィンドウで、**[キーリング（Key Rings）]** > ***[キーリング名（key_ring_name）]*** の順に選択します。

ステップ 24 [Work] ペインの [Certificate] フィールドに、認証局から受信した署名付き証明書を貼り付けます。

ステップ 25 [Submit] をクリックします。

（注） CSR がキー リングで示されている認証局によって署名されていない場合、または証明書に MS-DOS 形式の行末が含まれている場合は、エラー メッセージが表示され、証明書は承認されません。MS-DOS 形式の行末を削除します。

キーが確認されて [Work] ペインの [Admin State] が [Completed] に変わり、HTTP ポリシーを使用できるようになります。

ステップ 26 メニュー バーで、**[Fabric]** > **[Fabric Policies]** の順に選択します。

ステップ 27 [Navigation] ペインで、**[Pod Policies]** > **[Policies]** > **[Management Access]** > **[default]** の順に選択します。

ステップ 28 [Work] ペインの [Admin Key Ring] ドロップダウン リストで目的のキー リングを選択します。

ステップ 29 （オプション）証明書ベースの認証では、[Client Certificate TP] ドロップダウン リストで、以前に作成したローカル ユーザ ポリシーを選択し、[Client Certificate Authentication state] の [Enabled] をクリックします。

ステップ 30 [Submit] をクリックします。
すべての Web サーバが再起動されます。証明書がアクティブになり、デフォルト以外のキー リングが HTTPS アクセスに関連付けられています。

次のタスク

証明書の失効日には注意しておき、期限切れになる前に対応する必要があります。更新された証明書に対して同じキーペアを維持するには、CSR を維持する必要があります。これは、CSR にはキー リング内の秘密キーとペアになる公開キーが含まれているためです。証明書が期限切れになる前に、同じ CSR を再送信する必要があります。キー リングを削除すると、Cisco APIC に内部的に保存されている秘密キーも削除されるため、新しいキー リングの削除または作成は行わないでください。

NX-OS CLI を使用した証明書ベースの認証の有効化

証明書ベースの認証を有効にするには、次の手順を実行します。

例：

```
To enable CAC for https access:
  configure terminal
   comm-policy default
    https
     client-cert-ca <ca name>
     client-cert-state-enable
 To disable:
  configure terminal
   comm-policy default
    https
     no client-cert-state-enable
     no client-cert-ca
```