ユーザアクセス、認証およびアカウンティング

この章の内容は、次のとおりです。

アクセス権のワークフローの依存関係

Cisco Application Centric Infrastructure (ACI) RBAC のルールによって、ファブリック全体へのアクセスを有効にするか、一部へのアクセスに制限します。たとえば、ベア メタル サーバ アクセス用のリーフ スイッチを設定するには、ログインしている管理者が `infra` ドメインに対する権限を持っている必要があります。デフォルトでは、テナント管理者は `infra` ドメインに対する権限を持っていません。この場合、リーフ スイッチに接続されているベア メタル サーバの使用を計画しているテナント管理者は、そのために必要なすべての手順を実行することはできません。テナント管理者は、`infra` ドメインに対する権限を持っているファブリック管理者と連携する必要があります。ファブリック管理者は、テナント管理者が ACI リーフ スイッチに

接続されたベア メタル サーバを使用するアプリケーション ポリシーを導入するために使用するスイッチ設定ポリシーをセットアップします。

ユーザ アクセス、認可およびアカウンティング

Application Policy Infrastructure Controller（APIC）ポリシーは、Cisco Application Centric Infrastructure（ACI）ファブリックの認証、認可、およびアカウンティング（AAA）機能を管理します。ユーザ権限、ロール、およびドメインとアクセス権限の継承を組み合わせることにより、管理者は細分化された方法で管理対象オブジェクト レベルで AAA 機能を設定することができます。これらの設定は、REST API、CLI、または GUI を使用して実行できます。

マルチテナントのサポート

コア Application Policy Infrastructure Controller（APIC）の内部データ アクセス コントロール システムにより、マルチテナント分離が提供され、テナント間での個人情報の漏洩が防止されます。読み取り/書き込みの制約により、テナントによる他のテナントの設定、統計情報、障害、またはイベント データの参照が防止されます。管理者によって読み取り権限が割り当てられない限り、テナントはファブリックの設定、ポリシー、統計情報、障害、またはイベントの読み取りが制限されます。

ユーザ アクセス：ロール、権限、セキュリティ ドメイン

APIC では、ロールベース アクセス コントロール (RBAC) を介してユーザのロールに従ってアクセスが提供されます。Cisco Application Centric Infrastructure (ACI) ファブリック ユーザは、次に関連付けられています。

- ロールのセット
- ロールごとの権限タイプ：アクセスなし、読み取り専用、または読み取り/書き込み
- ユーザがアクセスできる管理情報ツリー (MIT) の一部を識別する 1 つ以上のセキュリティ ドメイン タグ

ACI ファブリックは、管理対象オブジェクト (MO) レベルでアクセス権限を管理します。権限は、システム内の特定の機能に対するアクセスを許可または制限する MO です。たとえば、ファブリック機器は権限ビットです。このビットは、Application Policy Infrastructure Controller (APIC) によって物理ファブリックの機器に対応するすべてのオブジェクトで設定されます。

ロールは権限ビットの集合です。たとえば、「管理者」ロールが「ファブリック機器」と「テナントセキュリティ」に対する権限ビットに設定されていると、「管理者」ロールにはファブリックの機器とテナントセキュリティに対応するすべてのオブジェクトへのアクセス権があります。

セキュリティ ドメインは、ACI MIT オブジェクト階層の特定のサブツリーに関連付けられたタグです。たとえば、デフォルトのテナント「common」にはドメイン タグ `common` が付いています。同様に、特殊なドメイン タグ `all` の場合、MIT オブジェクト ツリー全体が含まれます。

管理者は、MIT オブジェクト階層にカスタム ドメイン タグを割り当てることができます。たとえば、管理者は「solar」という名前のテナントにドメイン タグ「solar」を割り当てることができます。MIT 内では、特定のオブジェクトだけがセキュリティ ドメインとしてタグ付けできます。たとえば、テナントはセキュリティ ドメインとしてタグ付けすることができますが、テナント内のオブジェクトはできません。

（注） 作成することによって、セキュリティ ドメインのパスワード強度のパラメータを設定できます **カスタム条件** かを選択して **任意の 3 つの条件** が提供されます。

ユーザを作成してロールを割り当てても、アクセス権は有効になりません。1 つ以上のセキュリティ ドメインにそのユーザを割り当てることも必要です。デフォルトでは、ACI ファブリックには事前作成された次の 2 つの特殊なドメインが含まれています。

- `All`：MIT 全体へのアクセスを許可
- `Infra`：ファブリック アクセス ポリシーなどの、ファブリック インフラストラクチャのオブジェクトおよびサブツリーへのアクセスを許可

（注） ユーザのクレデンシャルが許可しない管理対象オブジェクトの読み取り操作の場合、「DN/Class Unauthorized to read」ではなく「DN/Class Not Found」というエラーが返されます。ユーザのクレデンシャルが許可しない管理対象オブジェクトへの書き込み操作の場合、「HTTP 401 Unauthorized」というエラーが返されます。GUI では、ユーザのクレデンシャルが許可しないアクションの場合、表示されないか、またはグレー表示されます。

事前に定義された一連の管理対象オブジェクト クラスをドメインに関連付けることができます。これらのクラスがオーバーラップすることはできません。ドメインの関連付けをサポートするクラスの例：

- レイヤ 2 およびレイヤ 3 のネットワークで管理されたオブジェクト
- ネットワーク プロファイル (物理、レイヤ 2、レイヤ 3、管理など)
- Quality of Service（QoS）ポリシー

ドメインに関連付けることができるオブジェクトが作成されると、ユーザは、ユーザのアクセス権の範囲内でオブジェクトにドメインを割り当てる必要があります。ドメインの割り当てはいつでも変更できます。

仮想マシン管理 (VMM) ドメインがセキュリティ ドメインとしてタグ付けされている場合、セキュリティ ドメイン内のユーザは、対応するタグ付き VMM ドメインにアクセスできます。たとえば、solar という名前のテナントに sun というセキュリティ ドメインのタグが付いており、VMM ドメインにも sun というセキュリティ ドメインのタグが付いている場合、solar テナント内のユーザは各自のアクセス権限に従って VMM ドメインにアクセスできます。

ローカル ユーザの設定

初期の設定スクリプトで、管理者アカウントが設定され、管理者はシステム起動時の唯一のユーザとなります。APIC は、きめ細かなロールベースのアクセス コントロール システムをサポートしており、そのシステムでは、権限が少ない管理者以外のユーザを含め、ユーザ アカウントをさまざまなロールで作成することができます。

GUI を使用したローカル ユーザの設定

始める前に

- ACI ファブリックが設置され、APIC コントローラがオンラインになっており、APIC クラスタが形成されて正常に動作していること。
- 必要に応じて、ユーザがアクセスするセキュリティ ドメインが定義されていること。たとえば、新しい使用アカウントがテナントにアクセスすることを制限する場合は、それに従ってテナント ドメインにタグ付けします。
- 以下を行うことができる APIC ユーザ アカウントを使用できること。
  - TACACS+ プロバイダーの作成。
  - ターゲット セキュリティ ドメインでのローカル ユーザ アカウントの作成。ターゲット ドメインが `all` である場合、新しいローカル ユーザの作成に使用するログイン アカウントは、`all` にアクセスできるファブリック全体の管理者である必要があります。ターゲット ドメインがテナントである場合、新しいローカル ユーザの作成に使用するログイン アカウントは、ターゲット テナント ドメインに対する完全な読み取り/書き込みアクセス権を持つテナント管理者である必要があります。

ステップ 1 メニュー バーで、**[ADMIN]** > **[AAA]** を選択します。

ステップ 2 [Navigation] ペインの [Work] ペインで、[Users] と [Local Users] をクリックします。

ステップ 3 [Work] ペインの [Local Users] タブに自身が含まれていることを確認します。

デフォルトでは admin ユーザが表示されます。

ステップ 4 [Work] ペインで、タスク アイコン ドロップダウンリストをクリックして [Create Local User] を選択します。

ステップ 5 [User Identity] ダイアログボックスで、次の操作を実行します。

a) [Login ID] フィールドで、ID を追加します。

b) [Password] フィールドにパスワードを入力します。

ユーザがパスワードを設定する時点で、APIC によって以下の基準が検証されます。

- パスワードの最小長は 8 文字です。
- パスワードの最大長は 64 文字です。

- 連続して繰り返される文字は 3 文字未満です。
- 小文字、大文字、数字、記号の文字種のうち少なくとも 3 種類の文字が含まれている必要があります。
- 簡単に推測できるパスワードは使用しません。
- ユーザ名やユーザ名を逆にしたものは使用できません。
- cisco、isco、またはこれらの文字列の並べ替えを変化させたものや、それらの文字の大文字化の変更により取得される変形語であってはなりません。

c) [Confirm Password] フィールドで、パスワードを確認します。
d) （オプション）証明書ベースの認証の場合は、[User Certificate Attribute] フィールドに認証証明書のユーザ ID を入力します。
e) [Finish] をクリックします。

ステップ 6 [Security] ダイアログボックスで、ユーザに必要なセキュリティ ドメインを選択し、[Next] をクリックします。

ステップ 7 [Roles] ダイアログボックスで、ユーザのロールを選択するためのオプション ボタンをクリックし、[Next] をクリックします。

読み取り専用または読み取り/書き込み権限を提供できます。

ステップ 8 [Navigation] ペインで、作成したユーザの名前をクリックします。[Work] ペインで、[Security Domains] 領域のユーザの横にある [+] 記号を展開します。
ユーザのアクセス権限が表示されます。

GUI を使用した SSH 公開キー認証の設定

始める前に

- ターゲット セキュリティ ドメインでローカル ユーザ アカウントを作成します。ターゲット ドメインが `all` である場合、新しいローカル ユーザの作成に使用するログイン アカウントは、`all` にアクセスできるファブリック全体の管理者である必要があります。ターゲット ドメインがテナントである場合、新しいローカル ユーザの作成に使用するログイン アカウントは、ターゲット テナント ドメインに対する完全な読み取り/書き込みアクセス権を持つテナント管理者である必要があります。
- UNIX コマンド **ssh-keygen** を使用して公開キーを生成します。

  デフォルトのログイン ドメインは **local** に設定する必要があります。

ステップ 1 メニュー バーで **[ADMIN]** > **[Users]** を選択し、[Local Users] タブに自身が含まれていることを確認します。

ステップ 2 [Navigation] ペインで、事前に作成したユーザの名前をクリックします。

ステップ 3 [Work] ペインで、[SSH Keys] テーブルを展開して次の情報を入力します。

a) [Name] フィールドにキーの名前を入力します。

b) [Key] フィールドに、事前に作成した公開キーを入力します。[Update] をクリックします。

（注） ファイルを作成する、SSH 秘密キーをメニュー バーで、リモートの場所をダウンロードするため、展開 **ファームウェア** > **ダウンロード タスク** 。

NX-OS スタイル CLI を使用したローカル ユーザの設定

手順の概要

1. NX-OS CLI で、次に示すようにしてコンフィギュレーション モードを開始します。
2. 新しいユーザを次に示すように作成します。

手順の詳細

ステップ 1 NX-OS CLI で、次に示すようにしてコンフィギュレーション モードを開始します。

例：

```
apic1# configure
apic1(config)#
```

ステップ 2 新しいユーザを次に示すように作成します。

例：

```
apic1(config)# username
 WORD       User name (Max Size 28)
 admin
 cli-user
 jigarshah
 test1
 testUser

apic1(config)# username test
apic1(config-username)#
 account-status      Set The status of the locally-authenticated user account.
 certificate         Create AAA user certificate in X.509 format.
 clear-pwd-history   Clears the password history of a locally-authenticated user
 domain              Create the AAA domain to which the user belongs.
 email               Set The email address of the locally-authenticated user.
 exit                Exit from current mode
 expiration          If expires enabled, Set expiration date of locally-authenticated user account.

 expires             Enable expiry for locally-authenticated user account
 fabric              show fabric related information
 first-name          Set the first name of the locally-authenticated user.
 last-name           Set The last name of the locally-authenticated user.
 no                  Negate a command or set its defaults
 password            Set The system user password.
```

```
 phone                Set The phone number of the locally-authenticated user.
 pwd-lifetime         Set The lifetime of the locally-authenticated user password.
 pwd-strength-check   Enforces the strength of the user password
 show                 Show running system information
 ssh-key              Update ssh key for the user for ssh authentication
 where                show the current mode

apic1(config-username)# exit
```

REST API を使用したローカル ユーザの設定

手順の概要

1. ローカル ユーザを作成します。

手順の詳細

ローカル ユーザを作成します。

例：

```
URL: https://apic-ip-address/api/policymgr/mo/uni/userext.xml
POST CONTENT:
        <aaaUser name="operations" phone="" pwd="<strong_password>" >
            <aaaUserDomain childAction="" descr="" name="all" rn="userdomain-all" status="">
                <aaaUserRole childAction="" descr="" name="Ops" privType="writePriv"/>
            </aaaUserDomain>
      </aaaUser>
```

リモート ユーザの設定

ローカル ユーザを設定する代わりに、APIC を一元化された企業クレデンシャルのデータセンターに向けることができます。APIC は、Lightweight Directory Access Protocol（LDAP）、Active Directory、RADIUS、および TACACS+ をサポートしています。

（注） APICが少数側である（クラスタから切断されている）場合、ACIは分散システムであり、ユーザ情報がAPICSに分散されるため、リモートログインは失敗する可能性があります。ただし、ローカル ログインは APIC に対してローカルであるため、この場合も機能します。

3.1 (1) のリリース以降、**サーバ モニタリング** はRADIUS、TACACS +、LDAP、およびRSA を介して設定され、個別の AAA サーバがアクティブかを判断できます。サーバ モニタリング機能は、サーバがアクティブかどうか確認するためそれぞれのプロトコルのログインを使用しま

す。たとえば、LDAP サーバは ldap lログインを使用し、Radius サーバはサーバがアクティブか判断するサーバ モニタリング機能を持つ radius のログインを使用します。

外部認証プロバイダーを通じて認証されたリモートユーザを設定するには、次の前提条件を満たす必要があります。

- DNS 設定は、RADIUS サーバのホスト名ですでに名前解決されている必要があります。
- 管理サブネットを設定する必要があります。

外部認証サーバの AV ペア

Cisco APIC では、管理者が外部認証サーバで Cisco AV ペアを設定する必要があります。Cisco AV ペアは、ユーザの RBAC ロールおよび権限に必要な APIC を指定します。Cisco AV ペアの形式は、RADIUS、LDAP、または TACACS+ のものと同じです。

外部認証サーバで Cisco AV ペアを設定するには、管理者が既存のユーザ レコードに Cisco AV ペアを追加します。Cisco AV ペアの形式は次のとおりです。

```
shell:domains =
domainA/writeRole1|writeRole2|writeRole3/readRole1|readRole2,
domainB/writeRole1|writeRole2|writeRole3/readRole1|readRole2
shell:domains =
domainA/writeRole1|writeRole2|writeRole3/readRole1|readRole2,
domainB/writeRole1|writeRole2|writeRole3/readRole1|readRole2(16003)
```

Cisco APIC リリース 2.1 より、AV ペアで UNIX ID が指定されていない場合は、APIC が固有の UNIX ユーザー ID を内部的に割り当てます。

（注） APIC の Cisco AV ペアの形式は互換性があり、他の Cisco AV ペアの形式と共存できます。APIC はすべての AV ペアから最初に一致した AV ペアを選択します。

APIC は、次の正規表現をサポートしています。

```
shell:domains\\s*[=:]\\s*((\\S+?/\\S*?/\\S*?)(,\\S+?/\\S*?/\\S*?){0,31})(\\(\\d+\\))$
shell:domains\\s*[=:]\\s*((\\S+?/\\S*?/\\S*?)(,\\S+?/\\S*?/\\S*?){0,31})$
```

例：

- 例 1：writeRole のみを持つ単一のログイン ドメインを含む Cisco AV ペア

```
shell:domains=domainA/writeRole1|writeRole2/
```

- 例 2：readRole のみを持つ単一のログイン ドメインを含む Cisco AV ペア

```
shell:domains=domainA//readRole1|readRole2
```

（注） 文字「/」はログイン ドメインごとに writeRole と readRole の間を区切る記号で、使用するロールの種類が 1 つのみである場合も必要です。

Cisco AV ペアの文字列は、大文字と小文字が区別されます。エラーが表示されなくても、使用する大文字と小文字がドメイン名またはロールに一致していない場合は、予期しない権限が付与されることがあります。

オープン RADIUS サーバ（/etc/raddb/users）の設定例は次のとおりです。

```
aaa-network-admin Cleartext-Password := "<password>"
Cisco-avpair = "shell:domains = all/aaa/read-all(16001)"
```

AV ペアを割り当てるためのベスト プラクティス

ベスト プラクティスとして、

Cisco は、bash シェルでユーザに割り当てられる AV ペアには 16000 ～ 23999 の範囲の一意の UNIX ユーザ ID を割り当てることを推奨します（SSH、Telnet または Serial/KVM のコンソールを使用）。Cisco AV ペアが UNIX ユーザ ID を提供しない状況が発生すると、そのユーザにはユーザ ID 23999 または範囲内の類似した番号が割り当てられます。これにより、そのユーザのホーム ディレクトリ、ファイル、およびプロセスに UNIX ID 23999 を持つリモート ユーザがアクセスできるようになってしまいます。

リモート認証サーバがその av ペアを cisco 応答 UNIX ID を明示的に指定していないことを確認するには、(リモート ユーザ アカウントを使用) は、管理者として、APIC とログインへの SSH セッションを開きます。ログインすると、次のコマンド (置換」ユーザ id「ログに記録するユーザ名と) を実行します。

```
admin@apic1:remoteuser-userid> cd /mit/uni/userext/remoteuser-userid
admin@apic1:remoteuser-userid> cat summary
```

Cisco AV ペアの文字列は、大文字と小文字が区別されます。エラーが表示されなくても、使用する大文字と小文字がドメイン名またはロールに一致していない場合は、予期しない権限が付与されることがあります。

外部認証サーバの AV ペアの設定

属性/値（AV）のペア文字列のカッコ内の数字は、セキュア シェル（SSH）または Telnet を使用してログインしたユーザの UNIX ユーザ ID として使用されます。

手順の概要

1. 外部認証サーバの AV ペアを設定します。

手順の詳細

外部認証サーバの AV ペアを設定します。

Cisco AV ペアの定義は次のとおりです（シスコは、UNIX ユーザ ID が指定されているかどうかにかかわらず AV ペアをサポートします）

例：

```
* shell:domains =
domainA/writeRole1|writeRole2|writeRole3/readRole1|readRole2,domainB/writeRole1|writeRole2|writeRole3/readRole1|readRole2
         * shell:domains =
domainA/writeRole1|writeRole2|writeRole3/readRole1|readRole2,domainB/writeRole1|writeRole2|writeRole3/readRole1|readRole2(8101)

These are the boost regexes supported by APIC:
uid_regex("shell:domains\\s*[=:]\\s*((\\S+?/\\S*?/\\S*?)(,\\S+?/\\S*?/\\S*?){0,31})(\\(\\d+\\))$");
regex("shell:domains\\s*[=:]\\s*((\\S+?/\\S*?/\\S*?)(,\\S+?/\\S*?/\\S*?){0,31})$");
```

次に、例を示します。

```
shell:domains = coke/tenant-admin/read-all,pepsi//read-all(16001)
```

TACACS+ アクセス用の APIC の設定

始める前に

- Cisco Application Centric Infrastructure（ACI）ファブリックが設置され、Application Policy Infrastructure Controller（APIC）がオンラインになっており、APIC クラスタが形成されて正常に動作していること。
- TACACS+ サーバのホスト名または IP アドレス、ポート、およびキーを使用できること。
- APIC 管理エンドポイント グループを使用できること。

ステップ 1 APIC で、**TACACS+ プロバイダー**を作成します。

a) メニュー バーで、**[Admin]** > **[AAA]** の順に選択します。
b) [Navigation] ペインで、**[TACACS+ Managment]** > **[TACACS+ Providers]** の順に選択します。
c) [Work] ペインで、**[Actions]** > **[Create TACACS+ Provider]** の順に選択します。
d) TACACS+ ホスト名（または IP アドレス）、ポート、認証プロトコル、キー、および管理エンドポイント グループを指定します。

（注） APICがインバンド管理に接続するために設定されている場合、アウトオブバンド管理は認証に機能しません。APIC リリース 2.1(1x) では、APIC サーバとその他の外部管理デバイス間のデフォルト管理接続として、インバンドおよびアウトオブ バンド間のグローバル トグルを設定できます。

APIC GUI のインバンドまたはアウトオブバンド管理のトグル：

- リリース 2.2(1x) 以前、**[ナビゲーション]** ペインでは、**[ファブリック]** > **[ファブリック ポリシー]** > **[グローバル ポリシー]** > **[接続設定]** を選択します。**[作業ペイン]** で **[インバンド]** または **[アウトバウンド]** のどちらかを選択します。
- リリース 2.2(x) および 2.3(x): では、**[ナビゲーション]** ペインで、**[ファブリック]** > **[ファブリック ポリシー]** > **[グローバル ポリシー]** > **[APIC 接続設定]** を選択します。**[作業ペイン]** で **[インバンド]** または **[アウトバウンド]** のどちらかを選択します。
- リリース 3.0(1x) 以降、**[ナビゲーション]** ペインで、**[システム]** > **[システム設定]** > **[APIC 接続設定]** を選択します。**[作業ペイン]** で **[インバンド]** または **[アウトバウンド]** のどちらかを選択します。

ステップ 2 [TACACS+ Provider Group] を作成します。

a) [Navigation] ペインで、**[TACACS+ Managment]** > **[TACACS+ Provider Groups]** の順に選択します。
b) [Work] ペインで、**[Actions]** > **[Create TACACS+ Provider Group]** の順に選択します。
c) 必要に応じて、TACACS+ プロバイダー グループ名、説明、およびプロバイダーを指定します。

ステップ 3 TACACS+ の [Login Domain] を作成します。

a) **[Navigation]** ペインで、**[AAA Authentication]** > **[Login Domains]** の順に選択します。
b) **[Work]** ペインで、**[Actions]** > **[Create Login Domain]** の順に選択します。
c) 必要に応じて、ログイン ドメイン名、説明、レルム、およびプロバイダー グループを指定します。

次のタスク

これで、APIC の TACACS+ 設定手順は完了です。次に、RAIDUS サーバも使用する場合は、RADIUS 用の APIC の設定も行います。TACACS+ サーバのみを使用する場合は、次の ACS サーバ設定に関するトピックに進みます。

RADIUS アクセス用の APIC の設定

始める前に

- ACI ファブリックが設置され、Application Policy Infrastructure Controller (APIC) がオンラインになっており、APIC クラスタが形成されて正常に動作していること。
- RADIUS サーバのホスト名または IP アドレス、ポート、認証プロトコル、およびキーを使用できること。

• APIC 管理エンドポイント グループを使用できること。

ステップ 1 APIC で、RADIUS プロバイダーを作成します。

a) メニュー バーで、**[Admin]** > **[AAA]** の順に選択します。
b) [Navigation] ペインで [Authentication] をクリックし、[RADIUS] タブをクリックします。
c) **[Work]** ペインで、**[Actions]** > **[Create RADIUS Provider]** の順に選択します。
d) RADIUS ホスト名 (または IP アドレス)、ポート、プロトコル、および管理エンドポイント グループを指定します。

（注） APIC がインバンド管理接続用に設定されている場合、アウトバンド管理は認証のために機能しません。APIC リリース 2.1(1x) では、APIC サーバとその他の外部管理デバイス間のデフォルト管理接続として、インバンドおよびアウトオブ バンド間のグローバル トグルを設定できます。

APIC GUI のインバンドまたはアウトオブバンド管理のトグル：

- リリース 2.2(1x) 以前、**[ナビゲーション]** ペインでは、**[ファブリック]** > **[ファブリック ポリシー]** > **[グローバル ポリシー]** > **[接続設定]** を選択します。**[作業ペイン]** で **[インバンド]** または **[アウトバウンド]** のどちらかを選択します。
- リリース 2.2(x) および 2.3(x): では、**[ナビゲーション]** ペインで、**[ファブリック]** > **[ファブリック ポリシー]** > **[グローバル ポリシー]** > **[APIC 接続設定]** を選択します。**[作業ペイン]** で **[インバンド]** または **[アウトバウンド]** のどちらかを選択します。
- リリース 3.0(1x) 以降、**[ナビゲーション]** ペインで、**[システム]** > **[システム設定]** > **[APIC 接続設定]** を選択します。**[作業ペイン]** で **[インバンド]** または **[アウトバウンド]** のどちらかを選択します。

ステップ 2 RADIUS のログイン ドメインを作成します。

a) **[Navigation]** ペインで、**[AAA Authentication]** > **[Login Domains]** の順に選択します。
b) **[Work]** ペインで、**[Actions]** > **[Create Login Domain]** の順に選択します。
c) 必要に応じて、ログイン ドメイン名、説明、レルム、およびプロバイダー グループを指定します。

次のタスク

これで、APIC RADIUS 設定手順は完了です。次に、RADIUS サーバを設定します。

APIC への RADIUS および TACACS+ アクセス用の Cisco Secure Access Control Server の設定

始める前に

- Cisco Secure Access Control Server (ACS) バージョン 5.5 がインストールされ、オンラインになっていること。

（注） ここでは手順の説明に ACS v5.5 が使用されています。ACS の他のバージョンでもこのタスクを実行できる可能性がありますが、GUI の手順はバージョンによって異なる場合があります。

- Cisco Application Policy Infrastructure Controller (Cisco APIC) の RADIUS キーまたは TACACS+ キーを使用できること (両方を設定する場合は両方のキー) 。
- Cisco APIC がインストールされ、オンラインになっていること。Cisco APIC クラスタが形成されて正常に動作していること。
- RADIUS または TACACS+ のポート、認証プロトコル、およびキーを使用できること。

ステップ 1 ACS サーバにログインして、Cisco APIC をクライアントとして設定します。

a) **[Network Resources]** > **[Network Devices Groups]** > **[Network Devices and AAA Clients]** に移動します。
b) クライアント名と Cisco APIC インバンド IP アドレスを指定し、TACACS+ または RADIUS (または両方) の認証オプションを選択します。

（注） RADIUS または TACACS+ のみの認証が必要な場合は、必要なオプションのみを選択します。

c) 共有秘密 (キー) や認証オプションに適したポートなど、認証の詳細を指定します。

（注） **[Shared Secret]** は Cisco APIC **[Provider]** キーと一致する必要があります。

ステップ 2 ID グループを作成します。

a) **[Users and Identity Stores]** > **[Internal Groups]** オプションに移動します。
b) 必要に応じて、**[Name]** と **[Parent Group]** を指定します。

ステップ 3 ユーザを ID グループにマッピングします。

a) **[Navigation]** ペインで、**[Users and Identity Stores]** > **[Internal Identity Stores]** > **[Users]** オプションをクリックします。
b) 必要に応じて、ユーザの **[Name]** と **[Identity Group]** を指定します。

ステップ 4 ポリシー要素を作成します。

a) **[Policy Elements]** オプションに移動します。

b) RADIUS の場合、[Authorization and Permissions] > [Network Access] > [Authorization Profiles **Name]** を指定します。TACACS+ の場合、必要に応じて、[Authorization and Permissions] > [Device Administration] > [Shell Profile **Name]** を指定します。

c) RADIUS の場合、必要に応じて、**[Attribute]** には「`cisco-av-pair`」、**[Type]** には「string」、**[Value]** には「**`shell:domains = <domain>/<role>/,<domain>// role`**」と指定します。TACACS+ の場合、必要に応じて、**[Attribute]** には「`cisco-av-pair`」、**[Requirement]** には「Mandatory」、**[Value]** には「**`shell:domains = <domain>/<role>/,<domain>// role`**」と指定します。

たとえば、*cisco-av-pair* の値が `shell:domains = solar/admin/,common// read-all(16001)` である場合、「`solar`」はセキュリティ ドメイン、「`admin`」は `solar` というセキュリティ ドメインに対する書き込み権限をこのユーザに付与するロール、「`common`」はCisco Application Centric Infrastructure (Cisco ACI) テナント common、「`read-all(16001)`」は Cisco ACI テナント common のすべてに対する読み取り権限をこのユーザに付与するロールです。

ステップ 5 サービス選択ルールを作成します。

a) RADIUS の場合、サービス選択ルールを作成して ID グループをポリシー要素に関連付けるには、**[Access Policies]** > **[Default Device Network Access Identity]** > **[Authorization]** に移動し、ルールの [Name]、[Status]、および [Conditions] を指定し、必要に応じて「`Internal Users:UserIdentityGroup in ALL Groups:<identity group name>`」を追加します。

b) TACACS+ の場合、サービス選択ルールを作成して ID グループをシェル プロファイルに関連付けるには、**[Access Policies]** > **[Default Device Admin Identity]** > **[Authorization]** に移動します。ルールの [Name] と [Conditions] を指定し、必要に応じて [Shell Profile] を選択します。

次のタスク

新しく作成された RADIUS および TACACS+ のユーザを使用して、Cisco APIC にログインします。割り当てられた RBAC ロールと権限に従って正しい Cisco APIC セキュリティ ドメインにアクセスできることを確認します。ユーザは、明示的に許可されていない項目にアクセスできてはなりません。読み取り/書き込みアクセス権が、そのユーザに設定されたものと一致している必要があります。

Cisco AVPair を使用した APIC アクセス用の Windows Server 2008 LDAP の設定

始める前に

- 最初に LDAP サーバを設定し、次に Cisco Application Policy Infrastructure Controller（Cisco APIC）を LDAP アクセス用に設定する。
- Microsoft Windows Server 2008 がインストールされ、オンラインになっていること。

- Microsoft Windows Server 2008 サーバ マネージャの ADSI Edit ツールがインストールされていること。ADSI Edit をインストールするには、Windows Server 2008 サーバ マネージャのヘルプに記載されている手順に従ってください。
- `CiscoAVPair` の属性の指定：Common Name = **`CiscoAVPair`**, LDAP Display Name = **`CiscoAVPair`**, Unique X500 Object ID = 1.3.6.1.4.1.9.22.1, Description = **`CiscoAVPair`**, Syntax = **`Case Sensitive String`**。

（注） LDAP 設定のベスト プラクティスは、属性文字列として **`CiscoAVPair`** を使用することです。顧客がオブジェクト ID 1.3.6.1.4.1.9.22.1 を使用して問題が発生した場合、その他のオブジェクト ID 1.3.6.1.4.1.9.2742.1-5 が LDAPサーバにも使用できます。

- 以下を行うことができる Microsoft Windows Server 2008 ユーザ アカウントを使用できること。
  - ADSI Edit を実行して `CiscoAVPair` 属性を Active Directory（AD）スキーマに追加します。
  - `CiscoAVPair` 属性パラメータに対するアクセス許可を持つように Active Directory LDAP ユーザーを設定します。
- ポート 636 は、SSll/TLS と LDAP の連携設定に必要です。

ステップ 1 ドメイン管理者として Active Directory（AD）サーバにログインします。

ステップ 2 AD スキーマに `CiscoAVPair` 属性を追加します。

a) **[Start]** > **[Run]** に移動し、「**`mmc`**」と入力し、Enter を押します。
Microsoft Management Console（MMC）が開きます。

b) **[File]** > **[Add/Remove Sanp-in]** > **[Add]** に移動します。

c) [Add Standalonee Snap-in] ダイアログボックスで、[Active Directory Schema] を選択し、[Add] をクリックします。
MMC コンソールが開きます。

d) **[属性]** フォルダを右クリックし、**[属性の作成]** オプションを選択します。
[Create New Attribute] ダイアログボックスが開きます。

e) **[共通名]** に「**`CiscoAVPair`**」、**[LDAP 表示名]** に「**`CiscoAVPair`**」、[Unique X500 Object ID] に「**`1.3.6.1.4.1.9.22.1`**」と入力し、**[構文]**で「**`Case Sensitive String`**」を選択します。

f) [OK] をクリックして、属性を保存します。

ステップ 3 [User Properties] クラスを [CiscoAVPair] 属性が含まれるように更新します。

a) MMC コンソールで、[Classes] フォルダを展開し、[user] クラスを右クリックし、[Properties] を選択します。
[user Properties] ダイアログボックスが開きます。

b) **[属性]** タブをクリックし、**[追加]** をクリックして **[スキーマのオブジェクトを選択する]** ウィンドウを開きます。
c) [Select a schema object:] リストで、「**CiscoAVPair**」を選択し、[Apply] をクリックします。
d) MMC コンソールで、[Active Directory Schema] を右クリックし、[Reload the Schema] を選択します。

ステップ 4 **CiscoAVPair** 属性のアクセス許可を設定します。

LDAP には **CiscoAVPair** 属性が含まれているため、LDAP ユーザーに Cisco APIC RBAC ロールを割り当てることにより Cisco APIC アクセス許可を付与する必要があります。

a) [ADSI Edit] ダイアログボックスで、Cisco APIC にアクセスする必要があるユーザを見つけます。
b) ユーザ名を右クリックし、[Properties] を選択します。
[<user> Properties] ダイアログボックスが開きます。
c) **[属性エディタ]** タブをクリックし、「*CiscoAVPair*」属性を選択し、*[値]* に「 **shell:domains = <domain>/<role>/,<domain>// role**」と入力します。

たとえば、*CiscoAVPair* の値が `shell:domains = solar/admin/,common// read-all(16001)` である場合、`solar` はセキュリティ ドメイン、`admin` は `solar` というセキュリティ ドメインに対する書き込み権限をこのユーザーに付与するロールであり、`common` は Cisco Application Centric Infrastructure（Cisco ACI）テナント共通であり `read-all(16001)` は Cisco ACI テナント共通のすべてに対する読み取り権限をこのユーザーに付与するロールです。

d) [OK] をクリックして変更を保存し、[<user> Properties] ダイアログボックスを閉じます。

Cisco APIC にアクセスするように LDAP サーバが設定されます。

次のタスク

Cisco APIC を LDAP アクセス用に設定します。

LDAP アクセス用の APIC の設定

始める前に

- Cisco Application Centric Infrastructure（ACI）ファブリックが設置され、Application Policy Infrastructure Controller（APIC）がオンラインになっており、APIC クラスタが形成されて正常に動作していること。
- LDAP サーバのホスト名または IP アドレス、ポート、バインド DN、ベース DN、およびパスワードを使用できること。
- APIC 管理エンドポイント グループを使用できること。

ステップ 1 APIC で、LDAP プロバイダーを設定します。
a) メニュー バーで、**[Admin]** > **[AAA]** の順に選択します。

b) [Navigation] ペインで [Authentication] を選択し、[Work] ペインで [LDAP] タブをクリックします。
c) [Work] ペインで、**[Actions]** > **[Create LDAP Provider]** の順に選択します。
d) LDAP ホスト名（または IP アドレス）、ポート、バインド DN、ベース DN、パスワード、属性、および管理エンドポイント グループを指定します。

（注）
- バインド DN は、APIC が LDAP サーバにログインするために使用する文字列です。APIC は、ログインしようとするリモート ユーザの検証にこのアカウントを使用します。ベース DN は、APIC がリモート ユーザ アカウントを検索する LDAP サーバのコンテナ名とパスです。これはパスワードが検証される場所です。フィルタを使用して、APIC が *cisco-av-pair* に使用するために要求している属性を見つけます。これには、APIC で使用するユーザ認証と割り当て済み RBAC ロールが含まれます。APIC は、この属性を LDAP サーバから要求します。
- **[属性]** フィールド：次のうちいずれかを入力します。
  - LDAPサーバの設定では、Cisco AVPair、入力 **`CiscoAVPair`** 。
  - LDAP グループ マップ LDAPサーバ設定、入力 **`memberOf`** 。
- APIC がインバンド管理接続用に設定されている場合、LDAP アクセス用にアウトオブバンド管理エンドポイント グループを選択しても有効にはなりません。また、インバンド管理エンドポイント グループ上のアウトオブバンドで LDAP サーバに接続することはできますが、LDAP サーバのスタティック ルートの設定が必要です。本書の設定手順例では、APIC インバンド管理エンドポイント グループを使用します。

ステップ 2 APIC で、LDAP のログイン ドメインを設定します。
a) [Navigation] ペインで、**[Authentication]** > **[Login Domains]** を選択します。
b) **[Work]** ペインで、**[Actions]** > **[Create Login Domain]** の順に選択します。
c) 必要に応じて、ログイン ドメイン名、説明、レルム、およびプロバイダー グループを指定します。

次のタスク

これで、APIC の LDAP 設定手順は完了です。次に、APIC の LDAP ログイン アクセスをテストします。

Cisco AV ペアが欠落しているか不良であるリモート ユーザのデフォルトの動作の変更

ステップ 1 メニュー バーで、**[ADMIN]** > **[AAA]** の順にクリックします。
ステップ 2 [Navigation] ペインで、[Users] をクリックします。

ステップ 3 [Work] ペインの [Remote Users] 領域で、[Remote user login policy] ドロップダウンリストから [Assign Default Role] を選択します。

デフォルト値は **[No Login]** です。**[Assign Default Role]** オプションは、Cisco AV ペアが欠落しているか不良であるユーザに最小限の読み取り専用権限を割り当てます。不正な AV ペアは、解析ルール適用時に問題があった AV ペアです。

NX-OS スタイル CLI を使用した欠落または不良 Cisco AV ペアを持つリモート ユーザのデフォルトの動作の変更

Cisco APIC では、管理者が外部認証サーバで Cisco AV ペアを設定する必要があります。これを行うには、管理者は既存のユーザ レコードに Cisco AV ペアを追加します。Cisco AV ペアは、ユーザの RBAC ロールおよび権限に必要な APIC を指定します。Cisco AV ペアの形式は、RADIUS、LDAP、または TACACS+ のものと同じです。AV ペアの形式には Cisco UNIX ユーザ ID が含まれるものと含まれないものがあります。すべてのリモート ユーザが同じロールを持ち、相互ファイル アクセスが許可される場合はどちらの形式でも問題ありません。UNIX ユーザ ID を指定しないと、APIC システムによって ID 23999 が適用され、AV ペア ユーザに対して複数のロールまたは読み取り権限が指定されます。これは、グループ設定で設定された権限より高いかまたは低い権限がユーザに付与される原因になることがあります。このトピックでは、許可されない動作を変更する方法について説明します。

NX-OS スタイル CLI を使用して欠落または不良 Cisco AV ペアを持つリモート ユーザのデフォルトの動作を変更するには、次の手順を実行します。

ステップ 1 NX-OS CLI で、コンフィギュレーション モードで開始します。

例：

```
apic1#
apic1# configure
```

ステップ 2 aaa ユーザ デフォルト ロールを設定します。

例：

```
apic1(config)# aaa user default-role
 assign-default-role  assign-default-role
 no-login             no-login
```

ステップ 3 aaa 認証ログイン メソッドを設定します。

例：

```
apic1(config)# aaa authentication
 login  Configure methods for login
```

```
apic1(config)# aaa authentication login
 console  Configure console methods
 default  Configure default methods
 domain   Configure domain methods

apic1(config)# aaa authentication login console
 <CR>

apic1(config)# aaa authentication login domain
 WORD      Login domain name
 fallback
```

署名ベースのトランザクションについて

Cisco ACI ファブリックの APIC コントローラは、ユーザを認証するためにさまざまな方法を提供します。

主要な認証方式ではユーザ名とパスワードが使用され、APIC REST API は APIC に対するその後のアクセスに使用できる認証トークンを返します。これは、HTTPS が使用不可であるか有効でない状況では安全でないと見なされます。

提供されている別の認証形式では、トランザクションごとに計算される署名が活用されます。その署名の計算には秘密キーが使用され、そのキーは安全な場所に保管して秘密にしておく必要があります。APIC がトークン以外の署名が付いた要求を受信すると、APIC は X.509 証明書を活用して署名を確認します。署名ベースの認証では、APICに対するすべてのトランザクションに新しく計算された署名が必要です。これは、ユーザがトランザクションごとに手動で行うタスクではありません。理想的には、この機能は APIC と通信するスクリプトまたはアプリケーションで使用する必要があります。この方法では、攻撃者がユーザクレデンシャルを偽装またはなりすますためには RSA/DSA キーを解読する必要があるため、最も安全です。

（注） また、リプレイ攻撃を防ぐためには HTTPS を使用する必要があります。

認証に X.509 証明書ベースの署名を使用する前に、次の必須タスクが完了していることを確認します。

1. OpenSSL または同様のツールを使用して X.509 証明書と秘密キーを作成します。
2. APIC のローカル ユーザを作成します（ローカル ユーザがすでに利用可能である場合、このタスクはオプションです）。
3. APIC のローカル ユーザに X.509 証明書を追加します。

注意事項と制約事項

次の注意事項と制約事項に従ってください。

- ローカル ユーザはサポートされます。リモート AAA ユーザはサポートされません。
- APIC GUI は証明書認証方式をサポートしません。
- WebSocket と eventchannel は X.509 要求では動作しません。
- サードパーティにより署名された証明書はサポートされません。自己署名証明書を使用します。

X.509 証明書と秘密キーの生成

ステップ 1 OpenSSL コマンドを入力して、X.509 証明書と秘密キーを生成します。

例：

```
$ openssl req -new -newkey rsa:1024 -days 36500 -nodes -x509 -keyout userabc.key -out userabc.crt
-subj '/CN=User ABC/O=Cisco Systems/C=US'
```

（注）
- X.509 証明書が生成されると、APIC のユーザ プロファイルに追加され、署名の確認に使用されます。秘密キーは、署名を生成するためにクライアントによって使用されます。
- 証明書には公開キーは含まれていますが、秘密キーは含まれていません。公開キーは、計算された署名を確認するために APIC によって使用される主要な情報です。秘密キーが APIC に保存されることはありません。このキーを秘密にしておく必要があります。

ステップ 2 OpenSSL を使用して証明書のフィールドを表示します。

例：

```
$ openssl x509 -text -in userabc.crt
        Certificate:
            Data:
                Version: 3 (0x2)
                Serial Number:
                    c4:27:6c:4d:69:7c:d2:b6
                Signature Algorithm: sha1WithRSAEncryption
                Issuer: CN=User ABC, O=Cisco Systems, C=US
                Validity
                    Not Before: Jan 12 16:36:14 2015 GMT
                    Not After : Dec 19 16:36:14 2114 GMT
                Subject: CN=User ABC, O=Cisco Systems, C=US
                Subject Public Key Info:
                    Public Key Algorithm: rsaEncryption
                    RSA Public Key: (1024 bit)
                        Modulus (1024 bit):
                            00:92:35:12:cd:2b:78:ef:9d:ca:0e:11:77:77:3a:
                            99:d3:25:42:94:b5:3e:8a:32:55:ce:e9:21:2a:ff:
                            e0:e4:22:58:6d:40:98:b1:0d:42:21:db:cd:44:26:
                            50:77:e5:fa:b6:10:57:d1:ec:95:e9:86:d7:3c:99:
                            ce:c4:7f:61:1d:3c:9e:ae:d8:88:be:80:a0:4a:90:
                            d2:22:e9:1b:25:27:cd:7d:f3:a5:8f:cf:16:a8:e1:
                            3a:3f:68:0b:9c:7c:cb:70:b9:c7:3f:e8:db:85:d8:
                            98:f6:e3:70:4e:47:e2:59:03:49:01:83:8e:50:4a:
                            5f:bc:35:d2:b1:07:be:ec:e1
                        Exponent: 65537 (0x10001)
                X509v3 extensions:
```

```
                X509v3 Subject Key Identifier:
                    0B:E4:11:C7:23:46:10:4F:D1:10:4C:C1:58:C2:1E:18:E8:6D:85:34
                X509v3 Authority Key Identifier:
                    keyid:0B:E4:11:C7:23:46:10:4F:D1:10:4C:C1:58:C2:1E:18:E8:6D:85:34
                    DirName:/CN=User ABC/O=Cisco Systems/C=US
                    serial:C4:27:6C:4D:69:7C:D2:B6

                X509v3 Basic Constraints:
                    CA:TRUE
        Signature Algorithm: sha1WithRSAEncryption
            8f:c4:9f:84:06:30:59:0c:d2:8a:09:96:a2:69:3d:cf:ef:79:
            91:ea:cd:ae:80:16:df:16:31:3b:69:89:f7:5a:24:1f:fd:9f:
            d1:d9:b2:02:41:01:b9:e9:8d:da:a8:4c:1e:e5:9b:3e:1d:65:
            84:ff:e8:ad:55:3e:90:a0:a2:fb:3e:3e:ef:c2:11:3d:1b:e6:
            f4:5e:d2:92:e8:24:61:43:59:ec:ea:d2:bb:c9:9a:7a:04:91:
            8e:91:bb:9d:33:d4:28:b5:13:ce:dc:fe:c3:e5:33:97:5d:37:
            cc:5f:ad:af:5a:aa:f4:a3:a8:50:66:7d:f4:fb:78:72:9d:56:
            91:2c
    [snip]
```

ローカル ユーザの設定

GUI を使用したローカル ユーザの作成とユーザ証明書の追加

ステップ 1 メニュー バーで、**[ADMIN]** > **[AAA]** を選択します。

ステップ 2 [Navigation] ペインの [Work] ペインで、[Users] と [Local Users] をクリックします。

ステップ 3 [Work] ペインの [Local Users] タブに自身が含まれていることを確認します。

デフォルトでは admin ユーザが表示されます。

ステップ 4 [Work] ペインで、タスク アイコン ドロップダウンリストをクリックして [Create Local User] を選択します。

ステップ 5 [Security] ダイアログボックスで、ユーザに必要なセキュリティ ドメインを選択し、[Next] をクリックします。

ステップ 6 [Roles] ダイアログボックスで、ユーザのロールを選択するためのオプション ボタンをクリックし、[Next] をクリックします。

読み取り専用または読み取り/書き込み権限を提供できます。

ステップ 7 [User Identity] ダイアログボックスで、次の操作を実行します。

a) [Login ID] フィールドで、ID を追加します。
b) [Password] フィールドにパスワードを入力します。
c) [Confirm Password] フィールドで、パスワードを確認します。
d) （オプション）証明書ベースの認証の場合は、[User Certificate Attribute] フィールドに認証証明書のユーザ ID を入力します。
e) [Finish] をクリックします。

ステップ 8 [Navigation] ペインで、作成したユーザの名前をクリックします。[Work] ペインで、[Security Domains] 領域のユーザの横にある [+] 記号を展開します。

ユーザのアクセス権限が表示されます。

ステップ 9 [Work] ペインの [User Certificates] 領域で、ユーザ証明書の [+] 記号をクリックし、[Create X509 Certificate] ダイアログ ボックスで次の操作を実行します。

a) [Name] フィールドに、証明書の名前を入力します。

b) [Data] フィールドに、ユーザ証明書の詳細を入力します。

c) **Submit** をクリックします。

X509 証明書がローカル ユーザ用に作成されます。

REST API を使用したローカル ユーザの作成とユーザ証明書の追加

ローカル ユーザを作成し、ユーザ証明書を追加します。

例：

```
method: POST
        url: http://apic/api/node/mo/uni/userext/user-userabc.json
        payload:
        {
            "aaaUser": {
                "attributes": {
                    "name": "userabc",
                    "firstName": "Adam",
                    "lastName": "BC",
                    "phone": "408-525-4766",
                    "email": "userabc@cisco.com",
                },
                "children": [{
                    "aaaUserCert": {
                        "attributes": {
                            "name": "userabc.crt",
                          "data": "-----BEGIN CERTIFICATE-----\nMIICjjCCAfegAwIBAgIJAMQnbE <snipped
 content> ==\n-----END CERTIFICATE-----",
                        },
                        "children": []
                    },
                    "aaaUserDomain": {
                        "attributes": {
                            "name": "all",
                        },
                        "children": [{
                            "aaaUserRole": {
                                "attributes": {
                                    "name": "aaa",
                                    "privType": "writePriv",
                                },
                                "children": []
                            }
                        }, {
                            "aaaUserRole": {
                                "attributes": {
                                    "name": "access-admin",
                                    "privType": "writePriv",
                                },
                                "children": []
                            }
                        }, {
```

```
    "aaaUserRole": {
        "attributes": {
            "name": "admin",
            "privType": "writePriv",
        },
        "children": []
    }
}, {
    "aaaUserRole": {
        "attributes": {
            "name": "fabric-admin",
            "privType": "writePriv",
        },
        "children": []
    }
}, {
    "aaaUserRole": {
        "attributes": {
            "name": "nw-svc-admin",
            "privType": "writePriv",
        },
        "children": []
    }
}, {
    "aaaUserRole": {
        "attributes": {
            "name": "ops",
            "privType": "writePriv",
        },
        "children": []
    }
}, {
    "aaaUserRole": {
        "attributes": {
            "name": "read-all",
            "privType": "writePriv",
        },
        "children": []
    }
}, {
    "aaaUserRole": {
        "attributes": {
            "name": "tenant-admin",
            "privType": "writePriv",
        },
        "children": []
    }
}, {
    "aaaUserRole": {
        "attributes": {
            "name": "tenant-ext-admin",
            "privType": "writePriv",
        },
        "children": []
    }
}, {
    "aaaUserRole": {
        "attributes": {
            "name": "vmm-admin",
            "privType": "writePriv",
        },
        "children": []
    }
}]
```

```
                    }
                }]
            }
        }
```

Python SDK を使用したローカル ユーザの作成

ローカル ユーザを作成します。

例：

```
#!/usr/bin/env python
from cobra.model.pol import Uni as PolUni
from cobra.model.aaa import UserEp as AaaUserEp
from cobra.model.aaa import User as AaaUser
from cobra.model.aaa import UserCert as AaaUserCert
from cobra.model.aaa import UserDomain as AaaUserDomain
from cobra.model.aaa import UserRole as AaaUserRole
from cobra.mit.access import MoDirectory
from cobra.mit.session import LoginSession
from cobra.internal.codec.jsoncodec import toJSONStr

APIC = 'http://10.10.10.1'
username = 'admin'
password = 'p@$$w0rd'

session = LoginSession(APIC, username, password)
modir = MoDirectory(session)
modir.login()

def readFile(fileName=None, mode="r"):
   if fileName is None:
      return ""
   fileData = ""
   with open(fileName, mode) as aFile:
      fileData = aFile.read()
   return fileData

# Use a dictionary to define the domain and a list of tuples to define
# our aaaUserRoles (roleName, privType)
# This can further be abstracted by doing a query to get the valid
# roles, that is what the GUI does

userRoles = {'all': [
               ('aaa', 'writePriv'),
               ('access-admin', 'writePriv'),
               ('admin', 'writePriv'),
               ('fabric-admin', 'writePriv'),
               ('nw-svc-admin', 'writePriv'),
               ('ops', 'writePriv'),
               ('read-all', 'writePriv'),
               ('tenant-admin', 'writePriv'),
               ('tenant-ext-admin', 'writePriv'),
               ('vmm-admin', 'writePriv'),
           ],
}
```

```
uni = PolUni('') # '' is the Dn string for topRoot
aaaUserEp = AaaUserEp(uni)
aaaUser = AaaUser(aaaUserEp, 'userabc', firstName='Adam',
                  email='userabc@cisco.com')

aaaUser.lastName = 'BC'
aaaUser.phone = '555-111-2222'
aaaUserCert = AaaUserCert(aaaUser, 'userabc.crt')
aaaUserCert.data = readFile("/tmp/userabc.crt")
# Now add each aaaUserRole to the aaaUserDomains which are added to the
# aaaUserCert
for domain,roles in userRoles.items():
   aaaUserDomain = AaaUserDomain(aaaUser, domain)
   for roleName, privType in roles:
       aaaUserRole = AaaUserRole(aaaUserDomain, roleName,
                                 privType=privType)
print toJSONStr(aaaUser, prettyPrint=True)

cr = ConfigRequest()
cr.addMo(aaaUser)
modir.commit(cr)
# End of Script to create a user
```

秘密キーを使用した署名の計算

始める前に

次の情報が用意されている必要があります。

- HTTP メソッド：GET、POST、DELETE
- 要求される REST API URI（クエリ オプションを含む）
- POST 要求の場合、APIC に送信される実際のペイロード
- ユーザの X.509 証明書の生成に使用される秘密キー
- APIC のユーザ X.509 証明書の宛先名

ステップ 1 HTTP メソッド、REST API URI、およびペイロードをこの順序で連結し、ファイルに保存します。

OpenSSL で署名を計算するには、この連結データをファイルに保存する必要があります。この例では、ファイル名 payload.txt を使用します。秘密キーは userabc.key というファイルにあることに注意してください。

例：

GET の例：

```
GET http://10.10.10.1/api/class/fvTenant.json?rsp-subtree=children
```

POST の例：

```
POST http://10.10.10.1/api/mo/tn-test.json{"fvTenant": {"attributes": {"status": "deleted", "name":
 "test"}}}
```

ステップ 2 Payload.テキスト ファイルに正しい情報が含まれていることを確認します。

たとえば、前の手順で示したような取得例を使用します。

```
GET http://10.10.10.1/api/class/fvTenant.json?rsp-subtree=children
```

Payload.テキスト ファイルには、次の情報のみ含める必要があります。

```
GET/api/class/fvTenant.json?rsp-subtree=children
```

ステップ 3 Payload.ファイルを作成するときに新しい行を間違って作成していないことを確認します。

例：

```
# cat -e payload.txt
```

次と同じように出力の最後に **$** 記号があるか確認します。

```
GET/api/class/fvTenant.json?rsp=subtree=children$
```

ある場合、Payload.ファイルを作成したときに新しい行が作成されたことを意味します。Payload.ファイルの生成時に新しい行が作成されることを防ぐには、次のようなコマンドを使用します。

```
echo -n "GET/api/class/fvTenant.json?rsp-subtree=children" >payload.txt
```

ステップ 4 OpenSSL を使用して、秘密キーとペイロード ファイルを使用して署名を計算します。

例：

```
openssl dgst -sha256 -sign userabc.key payload.txt > payload_sig.bin
```

生成されたファイルには、複数行に印字された署名があります。

ステップ 5 base64 形式に署名を変換します。

例：

```
openssl base64 -A -in payload_sig.bin -out payload_sig.base64
```

ステップ 6 Bash を使用して、署名から改行文字を取り除きます。

例：

```
$ tr -d '\n' < payload_sig.base64
P+OTqK0CeAZjl7+Gute2R1Ww8OGgtzE0wsLlx8fIXXl4V79Zl7
Ou8IdJH9CB4W6CEvdICXqkv3KaQszCIC0+Bn07o3qF//BsIplZmYChD6gCX3f7q
IcjGX+R6HAqGeK7k97cNhXlWEoobFPe/oajtPjOu3tdOjhf/9ujG6Jv6Ro=
```

（注） これは、この特定の要求に関して APIC に送信される署名です。その他の要求では、独自の署名を計算する必要があります。

ステップ 7 署名を文字列内に配置し、APIC が署名をペイロードと照合して確認できるようにします。

この完全な署名が、要求のヘッダー内のクッキーとして APIC に送信されます。

例：

```
APIC-Request-Signature=P+OTqK0CeAZjl7+Gute2R1Ww8OGgtzE0wsLlx8f
IXXl4V79Zl7Ou8IdJH9CB4W6CEvdICXqkv3KaQszCIC0+Bn07o3qF//BsIplZmYChD6gCX3f
7qIcjGX+R6HAqGeK7k97cNhXlWEoobFPe/oajtPjOu3tdOjhf/9ujG6Jv6Ro=;
APIC-Certificate-Algorithm=v1.0; APIC-Certificate-Fingerprint=fingerprint;
APIC-Certificate-DN=uni/userext/user-userabc/usercert-userabc.crt
```

（注） ここで使用される DN が、次のステップの x509 証明書を含むユーザ認定オブジェクトの DN に一致する必要があります。

ステップ 8 署名を使用して APIC と通信するには、Python SDK の CertSession クラスを使用します。

次のスクリプトは、ACI Python SDK の CertSession クラスを使用して、署名を使用して APIC に要求する方法の例です。

例：

```
#!/usr/bin/env python
# It is assumed the user has the X.509 certificate already added to
# their local user configuration on the APIC
from cobra.mit.session import CertSession
from cobra.mit.access import MoDirectory

def readFile(fileName=None, mode="r"):
    if fileName is None:
        return ""
    fileData = ""
    with open(fileName, mode) as aFile:
        fileData = aFile.read()
    return fileData

pkey = readFile("/tmp/userabc.key")
csession = CertSession("https://ApicIPOrHostname/",
                       "uni/userext/user-userabc/usercert-userabc", pkey)

modir = MoDirectory(csession)
resp = modir.lookupByDn('uni/fabric')
pring resp.dn
# End of script
```

（注） 前のステップで使用した DN が、このステップの x509 証明書を含むユーザ認定オブジェクトの DN に一致する必要があります。

アカウンティング

ACI ファブリック アカウンティングは、障害およびイベントと同じメカニズムで処理される以下の 2 つの管理対象オブジェクト（MO）によって処理されます。

- `aaaSessionLR` MO は、APIC およびスイッチでのユーザ アカウントのログイン/ログアウト セッション、およびトークン更新を追跡します。ACI ファブリック セッション アラート機能は、次のような情報を保存します。
  - ユーザ名
  - セッションを開始した IP アドレス
  - タイプ（telnet、https、REST など）
  - セッションの時間と長さ
  - トークン更新：ユーザ アカウントのログイン イベントは、ユーザ アカウントが ACI ファブリックの権利を行使するために必要な、有効なアクティブ トークンを生成します。

（注） トークンはログインに関係なく期限切れになります。ユーザはログアウトできますが、トークンは含まれているタイマー値の期間に従って期限切れになります。

- aaaModLR MO は、ユーザがオブジェクトに対して行う変更、およびいつ変更が発生したかを追跡します。
- AAA サーバが ping 可能でない場合は、使用不可としてマークされ、エラーが表示されます。

aaaSessionLR と aaaModLR の両方のイベント ログが、APIC シャードに保存されます。データがプリセットされているストレージ割り当てサイズを超えると、先入れ先出し方式でレコードを上書きします。

（注） APIC クラスタ ノードを破壊するディスク クラッシュや出火などの破壊的なイベントが発生した場合、イベント ログは失われ、イベント ログはクラスタ全体で複製されません。

aaaModLR MO と aaaSessionLR MO は、クラスまたは識別名（DN）でクエリできます。クラスのクエリは、ファブリック全体のすべてのログ レコードを提供します。ファブリック全体の aaaModLR レコードはすべて、GUI の **[Fabric]** > **[Inventory]** > **[POD]** > **[History]** > **[Audit Log]** セクションから入手できます。APIC GUI の **[History]** > **[Audit Log]** オプションを使用すると、GUI に示された特定のオブジェクトのイベント ログを表示できます。

標準の syslog、callhome、REST クエリ、および CLI エクスポート メカニズムは、aaaModLR MO と aaaSessionLR MO のクエリ データで完全にサポートされます。このデータをエクスポートするデフォルト ポリシーはありません。

APIC には、一連のオブジェクトまたはシステム全体のデータの集約を報告する、事前設定されたクエリはありません。ファブリック管理者は、aaaModLR および aaaSessionLR のクエリ データを定期的に syslog サーバにエクスポートするエクスポート ポリシーを設定できます。エクスポートされたデータを定期的にアーカイブし、システムの一部またはシステム ログ全体のカスタム レポートを生成するために使用できます。

共有サービスとしての外部ネットワークへのルーテッド接続の課金と統計情報

APIC は、共有サービスとしての外部ネットワークへのルーテッド接続用に設定されたポート（l3extInstP EPG）からバイト カウントとパケット カウントでの課金統計情報を収集するように設定できます。任意のテナントの任意の EPG が、外部ネットワークへのルーテッド接続用に l3extInstP EPG を共有できます。課金統計情報は、共有サービスとして l3extInstP EPG を使用する任意のテナント内の EPG ごとに収集できます。l3extInstP がプロビジョニングさ

れているリーフ スイッチは課金統計情報を APIC に転送し、そこで課金情報が集約されます。定期的に課金統計情報をサーバにエクスポートするようにアカウンティング ポリシーを設定できます。