



アクセス、認証およびアカウントティング

この章は、次の項で構成されています。

- [概要](#) (1 ページ)
- [設定 \(Configuration\)](#) (25 ページ)

概要

ユーザアクセス、認可およびアカウントティング

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、およびアカウントティング (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) セキュリティドメインのパスワード強度パラメータは、**[Custom Conditions]** を作成するか、または提供されている **[Any Three Conditions]** を選択して設定できます。

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。

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。

AAA RBAC の役割および権限

Application Policy Infrastructure Controller (APIC) では次の AAA の役割および権限を提供します。

ロール	権限	説明
aaa	aaa	ポリシーの認証、許可、アカウントिंग、インポート/エクスポートの設定に使用されます。
admin	admin	すべてのファブリックの機能へのフルアクセスを提供します。管理者権限は、その他のすべての権限を組み合わせたものとみなされます。

Role: access-admin	
権限	説明
access-connectivity-l1	インフラでレイヤ 1 の設定に使用します。例：セクタとポート レイヤ 1 のポリシー設定。
access-connectivity-l2	インフラでレイヤ 2 の設定に使用します。例：セクタおよび接続可能なエンティティ設定をカプセル化します。
access-connectivity-l3	インフラでレイヤ 3 の設定に使用され、テナントの L3Out でスタティック ルートの設定に使用されます。
access-connectivity-mgmt	管理インフラ ポリシーに使用されます。
access-connectivity-util	テナント ERSPAN ポリシーに使用されます。
access-equipment	アクセス ポート設定に使用されます。
access-protocol-l1	インフラでレイヤ 1 プロトコル設定に使用されます。
access-protocol-l2	インフラでレイヤ 2 プロトコル設定に使用されます。
access-protocol-l3	インフラでレイヤ 3 プロトコル設定に使用されます。
access-protocol-mgmt	NTP、SNMP、DNS、およびイメージ管理のためファブリック全体のポリシーに使用されます。
access-protocol-ops	クラスタポリシーおよびファームウェアポリシーなどの動作に関連するアクセス ポリシーに使用されます。
access-qos	CoPP および QoS に関連するポリシーの変更に使用されます。
Role: fabric-admin	
権限	説明
fabric-connectivity-l1	ファブリックでレイヤ 1 の設定に使用されます。例：セクタおよびポート レイヤ 1 のポリシーと vPC 保護。
fabric-connectivity-l2	ポリシー展開の影響を想定して警告を発生させるファームウェアおよび展開ポリシーで使用されます。
fabric-connectivity-l3	ファブリックでレイヤ 3 の設定に使用されます。例：ファブリック IPv4、IPv6、および MAC 保護グループ。
fabric-connectivity-mgmt	リーフ スイッチおよびスパイン スイッチのアトミック カウンタおよび診断ポリシーに使用されます。

Role: fabric-admin	
権限	説明
fabric-connectivity-util	リーフ スイッチおよびスパイン スイッチのアトミック カウンタ、診断、およびイメージ管理ポリシーに使用されます。
fabric-equipment	リーフ スイッチおよびスパイン スイッチのアトミック カウンタ、診断、およびイメージ管理ポリシーに使用されます。
fabric-protocol-l1	ファブリックでレイヤ 1 プロトコル設定に使用されます。
fabric-protocol-l2	ファブリックでレイヤ 2 プロトコル設定に使用されます。
fabric-protocol-l3	ファブリックでレイヤ 3 プロトコル設定に使用されます。
fabric-protocol-mgmt	NTP、SNMP、DNS、およびイメージ管理のためファブリック全体のポリシーに使用されます。
fabric-protocol-ops	ERSPAN および健全性のスコア ポリシーに使用されます。
fabric-protocol-util	ファームウェア管理トレースルートおよびエンドポイント トラッキング ポリシーに使用されます。
tenant-connectivity-util	リーフ スイッチおよびスパイン スイッチのアトミック カウンタ、診断、およびイメージ管理ポリシーに使用されます。
tenant-connectivity-l2	ブリッジドメインおよびサブネットを含む、レイヤ 2 接続の変更で使用されます。
tenant-connectivity-l3	VRF を含むレイヤ 3 接続の変更で使用されます。
tenant-protocol-ops	テナント トレースルート ポリシーに使用されます。

ロール	権限	説明
nw-svc-admin	nw-svc-device	レイヤ 4 ~ レイヤ 7 のサービスの管理に使用されます。
	nw-svc-devshare	共有のレイヤ 4 ~ レイヤ 7 のサービス デバイスの管理に使用されます。
	nw-svc-policy	レイヤ 4 ~ レイヤ 7 のネットワーク サービス オーケストレーションの管理に使用されます。
nw-svc-params	nw-svc-params	レイヤ 4 ~ レイヤ 7 のサービス ポリシーの管理に使用されます。

Role: ops	
権限	説明
ops	アトミック カウンタ、SPAN、TSW、技術サポート、トレースルート、分析、コアポリシーなど、ポリシーのモニタリングとトラブルシューティングを含む動作ポリシーに使用されます。
Role: read-all	
権限	説明
access-connectivity-l1	インフラでレイヤ 1 の設定に使用します。例：セクタとポート レイヤ 1 のポリシー設定。
access-connectivity-l2	インフラでレイヤ 2 の設定に使用します。例：セクタおよび接続可能なエンティティ設定をカプセル化します。
access-connectivity-l3	インフラでレイヤ 3 の設定に使用され、テナントの L3Out でスタティック ルートの設定に使用されます。
access-connectivity-mgmt	管理インフラ ポリシーに使用されます。
access-connectivity-util	テナント ERSPAN ポリシーに使用されます。
access-equipment	アクセス ポート設定に使用されます。
access-protocol-l1	インフラでレイヤ 1 プロトコル設定に使用されます。
access-protocol-l2	インフラでレイヤ 2 プロトコル設定に使用されます。
access-protocol-l3	インフラでレイヤ 3 プロトコル設定に使用されます。
access-protocol-mgmt	NTP、SNMP、DNS、およびイメージ管理のためファブリック全体のポリシーに使用されます。
access-protocol-ops	クラスタ ポリシーおよびファームウェア ポリシーなどの動作に関連するアクセス ポリシーに使用されます。
access-qos	CoPP および QoS に関連するポリシーの変更に使用されます。
fabric-connectivity-l1	ファブリックでレイヤ 1 の設定に使用されます。例：セクタおよびポート レイヤ 1 のポリシーと vPC 保護。
fabric-connectivity-l2	ポリシー展開の影響を想定して警告を発生させるファームウェアおよび展開ポリシーで使用されます。

Role: read-all	
権限	説明
fabric-connectivity-l3	ファブリックでレイヤ3の設定に使用されます。例：ファブリック IPv4、IPv6、および MAC 保護グループ。
fabric-protocol-l1	ファブリックでレイヤ1プロトコル設定に使用されます。
fabric-protocol-l2	ファブリックでレイヤ2プロトコル設定に使用されます。
fabric-protocol-l3	ファブリックでレイヤ3プロトコル設定に使用されます。
nw-svc-device	レイヤ4～レイヤ7のサービスの管理に使用されます。
nw-svc-devshare	共有のレイヤ4～レイヤ7のサービス デバイスの管理に使用されます。
nw-svc-params	レイヤ4～レイヤ7のサービス ポリシーの管理に使用されます。
nw-svc-policy	レイヤ4～レイヤ7のネットワーク サービス オーケストレーションの管理に使用されます。
ops	アトミックカウンタ、SPAN、TSW、技術サポート、トレースルート、分析、コアポリシーなど、ポリシーのモニタリングとトラブルシューティングを含む動作ポリシーに使用されます。
tenant-connectivity-util	リーフスイッチおよびスパインスイッチのアトミックカウンタ、診断、およびイメージ管理ポリシーに使用されます。
tenant-connectivity-l2	ブリッジドメインおよびサブネットを含む、レイヤ2接続の変更に使用されます。
tenant-connectivity-l3	VRF を含むレイヤ3接続の変更に使用されます。
tenant-connectivity-mgmt	テナント インバンドおよびアウトオブバンド管理接続の設定、アトミックカウンタや健全性スコアなどデバッグやモニタリングポリシーに使用されます。
tenant-epg	エンドポイントグループ、VRF、ブリッジドメインの削除/作成など、テナント設定の管理に使用されます。
tenant-ext-connectivity-l1	書き込みアクセスファームウェアポリシーに使用されます。
tenant-ext-connectivity-l2	テナント L2Out 設定の管理に使用されます。
tenant-ext-connectivity-l3	テナント L3Out 設定の管理に使用されます。

Role: read-all	
権限	説明
tenant-ext-connectivity-mgmt	ファームウェア ポリシーの書き込みアクセスとして使用されます。
tenant-ext-connectivity-util	トレースルート、ping、oam、eptrkなどのデバッグ/モニタリング/オブザーバ ポリシーに使用されます。
tenant-ext-protocol-l1	テナント外部レイヤ1プロトコルの管理に使用されます。通常、ファームウェア ポリシーの書き込みアクセスにのみ使用されます。
tenant-ext-protocol-l2	テナント外部レイヤ2プロトコルの管理に使用されます。通常、ファームウェア ポリシーの書き込みアクセスにのみ使用します。
tenant-ext-protocol-l3	BGP、OSPF、PIM、IGMPなどのテナント外部レイヤ3プロトコルの管理に使用されます。
tenant-ext-protocol-mgmt	ファームウェア ポリシーの書き込みアクセスとして使用されます。
tenant-ext-protocol-util	トレースルート、ping、oam、eptrkなどのデバッグ/モニタリング/オブザーバ ポリシーに使用されます。
tenant-network-profile	ネットワーク プロファイルの削除および作成、エンドポイント グループの削除および作成など、テナント設定の管理に使用されます。
tenant-protocol-l1	テナントでレイヤ1プロトコルの設定の管理に使用されます。
tenant-protocol-l2	テナントでレイヤ2プロトコルの設定の管理に使用されます。
tenant-protocol-l3	テナントでレイヤ3プロトコルの設定の管理に使用されます。
tenant-protocol-mgmt	ファームウェア ポリシーの書き込みアクセスとしてのみ使用されます。
tenant-protocol-ops	テナント トレースルート ポリシーに使用されます。
tenant-QoS	テナントの QoS に関連する設定に使用されます。
tenant-security	テナントのコントラクトに関連する設定に使用されます。

Role: read-all	
権限	説明
vmm-connectivity	仮想マシン接続に必要な APIC の VMM インベントリ内のすべてのオブジェクトを読み取るのに使用されます。
vmm-ep	APIC の VMM インベントリ内の仮想マシンとハイパーバイザ エンドポイントを読み取るために使用されます。
vmm-policy	仮想マシン ネットワーキングのポリシー管理に使用されます。
vmm-protocol-ops	VMM ポリシーでは使用されません。
vmm-security	VMware vCenter のユーザー名やパスワードなど VMM 認証ポリシーの管理に使用されます。

Role: tenant-admin	
権限	説明
aaa	ポリシーの認証、許可、アカウントिंग、インポート/エクスポートの設定に使用されます。
access-connectivity-l1	インフラでレイヤ 1 の設定に使用します。例：セクタとポート レイヤ 1 のポリシー設定。
access-connectivity-l2	インフラでレイヤ 2 の設定に使用します。例：セクタおよび接続可能なエンティティ設定をカプセル化します。
access-connectivity-l3	インフラでレイヤ 3 の設定に使用され、テナントの L3Out でスタティック ルートの設定に使用されます。
access-connectivity-mgmt	管理インフラ ポリシーに使用されます。
access-connectivity-util	テナント ERSPAN ポリシーに使用されます。
access-equipment	アクセス ポート設定に使用されます。
access-protocol-l1	インフラでレイヤ 1 プロトコル設定に使用されます。
access-protocol-l2	インフラでレイヤ 2 プロトコル設定に使用されます。
access-protocol-l3	インフラでレイヤ 3 プロトコル設定に使用されます。
access-protocol-mgmt	NTP、SNMP、DNS、およびイメージ管理のためファブリック全体のポリシーに使用されます。
access-protocol-ops	クラスタ ポリシーおよびファームウェア ポリシーなどの動作に関連するアクセス ポリシーに使用されます。

Role: tenant-admin	
権限	説明
access-qos	CoPP および QoS に関連するポリシーの変更で使用されます。
fabric-connectivity-l1	ファブリックでレイヤ1の設定に使用されます。例：セレクトアおよびポート レイヤ1のポリシーと vPC 保護。
fabric-connectivity-l2	ポリシー展開の影響を想定して警告を発生させるファームウェアおよび展開ポリシーで使用されます。
fabric-connectivity-l3	ファブリックでレイヤ3の設定に使用されます。例：ファブリック IPv4、IPv6、および MAC 保護グループ。
fabric-connectivity-mgmt	リーフ スイッチおよびスパインスイッチのアトミック カウンタおよび診断ポリシーに使用されます。
fabric-connectivity-util	リーフ スイッチおよびスパインスイッチのアトミック カウンタ、診断、およびイメージ管理ポリシーに使用されます。
fabric-equipment	リーフ スイッチおよびスパインスイッチのアトミック カウンタ、診断、およびイメージ管理ポリシーに使用されます。
fabric-protocol-l1	ファブリックでレイヤ1プロトコル設定に使用されます。
fabric-protocol-l2	ファブリックでレイヤ2プロトコル設定に使用されます。
fabric-protocol-l3	ファブリックでレイヤ3プロトコル設定に使用されます。
fabric-protocol-mgmt	NTP、SNMP、DNS、およびイメージ管理のためファブリック全体のポリシーに使用されます。
fabric-protocol-ops	ERSPAN および健全性のスコア ポリシーに使用されます。
fabric-protocol-util	ファームウェア管理トレースルートおよびエンドポイントトラッキング ポリシーに使用されます。
nw-svc-device	レイヤ4～レイヤ7のサービスの管理に使用されます。
nw-svc-devshare	共有のレイヤ4～レイヤ7のサービス デバイスの管理に使用されます。
nw-svc-params	レイヤ4～レイヤ7のサービス ポリシーの管理に使用されます。
nw-svc-policy	レイヤ4～レイヤ7のネットワーク サービス オーケストレーションの管理に使用されます。

Role: tenant-admin	
権限	説明
ops	アトミックカウンタ、SPAN、TSW、技術サポート、トレースルート、分析、コアポリシーなど、ポリシーのモニタリングとトラブルシューティングを含む動作ポリシーに使用されます。
tenant-connectivity-util	リーフスイッチおよびスパインスイッチのアトミックカウンタ、診断、およびイメージ管理ポリシーに使用されます。
tenant-connectivity-l2	ブリッジドメインおよびサブネットを含む、レイヤ2接続の変更に使用されます。
tenant-connectivity-l3	VRFを含むレイヤ3接続の変更に使用されます。
tenant-connectivity-mgmt	テナントインバンドおよびアウトオブバンド管理接続の設定、アトミックカウンタや健全性スコアなどデバッグやモニタリングポリシーに使用されます。
tenant-epg	エンドポイントグループ、VRF、ブリッジドメインの削除/作成など、テナント設定の管理に使用されます。
tenant-ext-connectivity-l1	書き込みアクセスファームウェアポリシーに使用されます。
tenant-ext-connectivity-l2	テナントL2Out設定の管理に使用されます。
tenant-ext-connectivity-l3	テナントL3Out設定の管理に使用されます。
tenant-ext-connectivity-mgmt	ファームウェアポリシーの書き込みアクセスとして使用されます。
tenant-ext-connectivity-util	トレースルート、ping、oam、eprkなどのデバッグ/モニタリング/オブザーバポリシーに使用されます。
tenant-ext-protocol-l1	テナント外部レイヤ1プロトコルの管理に使用されます。通常、ファームウェアポリシーの書き込みアクセスにのみ使用されます。
tenant-ext-protocol-l2	テナント外部レイヤ2プロトコルの管理に使用されます。通常、ファームウェアポリシーの書き込みアクセスにのみ使用します。
tenant-ext-protocol-l3	BGP、OSPF、PIM、IGMPなどのテナント外部レイヤ3プロトコルの管理に使用されます。
tenant-ext-protocol-mgmt	ファームウェアポリシーの書き込みアクセスとして使用されます。

Role: tenant-admin	
権限	説明
tenant-ext-protocol-util	トレースルート、ping、oam、eptrkなどのデバッグ/モニタリング/オブザーバポリシーに使用されます。
tenant-network-profile	ネットワークプロファイルの削除および作成、エンドポイントグループの削除および作成など、テナント設定の管理に使用されます。
tenant-protocol-l1	テナントでレイヤ1プロトコルの設定の管理に使用されます。
tenant-protocol-l2	テナントでレイヤ2プロトコルの設定の管理に使用されます。
tenant-protocol-l3	テナントでレイヤ3プロトコルの設定の管理に使用されます。
tenant-protocol-mgmt	ファームウェアポリシーの書き込みアクセスとしてのみ使用されます。
tenant-protocol-ops	テナントトレースルートポリシーに使用されます。
tenant-QoS	テナントのQoSに関連する設定に使用されます。
tenant-security	テナントのコントラクトに関連する設定に使用されます。
vmm-connectivity	仮想マシン接続に必要なAPICのVMMインベントリ内のすべてのオブジェクトを読み取るのに使用されます。
vmm-ep	APICのVMMインベントリ内の仮想マシンとハイパーバイザエンドポイントを読み取るために使用されます。
vmm-policy	仮想マシンネットワークキングのポリシー管理に使用されます。
vmm-protocol-ops	VMMポリシーでは使用されません。
vmm-security	VMware vCenterのユーザー名やパスワードなどVMM認証ポリシーの管理に使用されます。
Role: tenant-ext-admin	
権限	説明
tenant-connectivity-util	リーフスイッチおよびスパインスイッチのアトミックカウンタ、診断、およびイメージ管理ポリシーに使用されます。

Role: tenant-ext-admin	
権限	説明
tenant-connectivity-l2	ブリッジドメインおよびサブネットを含む、レイヤ2接続の変更に使用されます。
tenant-connectivity-l3	VRFを含むレイヤ3接続の変更に使用されます。
tenant-connectivity-mgmt	テナントインバンドおよびアウトオブバンド管理接続の設定、アトミックカウンタや健全性スコアなどデバッグやモニタリングポリシーに使用されます。
tenant-epg	エンドポイントグループ、VRF、ブリッジドメインの削除/作成など、テナント設定の管理に使用されます。
tenant-ext-connectivity-l1	書き込みアクセスファームウェアポリシーに使用されます。
tenant-ext-connectivity-l2	テナントL2Out設定の管理に使用されます。
tenant-ext-connectivity-l3	テナントL3Out設定の管理に使用されます。
tenant-ext-connectivity-mgmt	ファームウェアポリシーの書き込みアクセスとして使用されます。
tenant-ext-connectivity-util	トレースルート、ping、oam、eprtkなどのデバッグ/モニタリング/オブザーバポリシーに使用されます。
tenant-ext-protocol-l1	テナント外部レイヤ1プロトコルの管理に使用されます。通常、ファームウェアポリシーの書き込みアクセスにのみ使用されます。
tenant-ext-protocol-l2	テナント外部レイヤ2プロトコルの管理に使用されます。通常、ファームウェアポリシーの書き込みアクセスにのみ使用します。
tenant-ext-protocol-l3	BGP、OSPF、PIM、IGMPなどのテナント外部レイヤ3プロトコルの管理に使用されます。
tenant-ext-protocol-mgmt	ファームウェアポリシーの書き込みアクセスとして使用されます。
tenant-ext-protocol-util	トレースルート、ping、oam、eprtkなどのデバッグ/モニタリング/オブザーバポリシーに使用されます。
tenant-network-profile	ネットワークプロファイルの削除および作成、エンドポイントグループの削除および作成など、テナント設定の管理に使用されます。

Role: tenant-ext-admin	
権限	説明
tenant-protocol-l1	テナントでレイヤ 1 プロトコルの設定の管理に使用されます。
tenant-protocol-l2	テナントでレイヤ 2 プロトコルの設定の管理に使用されます。
tenant-protocol-l3	テナントでレイヤ 3 プロトコルの設定の管理に使用されます。
tenant-protocol-mgmt	ファームウェア ポリシーの書き込みアクセスとしてのみ使用されます。
tenant-protocol-ops	テナント トレースルート ポリシーに使用されます。
tenant-QoS	テナントの QoS に関連する設定に使用されます。
tenant-security	テナントのコントラクトに関連する設定に使用されます。
vmm-connectivity	仮想マシン接続に必要な APIC の VMM インベントリ内のすべてのオブジェクトを読み取るのに使用されます。
vmm-ep	APIC の VMM インベントリ内の仮想マシンとハイパーバイザ エンドポイントを読み取るために使用されます。
vmm-policy	仮想マシン ネットワーキングのポリシー管理に使用されます。
vmm-protocol-ops	VMM ポリシーでは使用されません。
vmm-security	VMware vCenter のユーザー名やパスワードなど VMM 認証ポリシーの管理に使用されます。

Role: vmm-admin	
権限	説明
vmm-connectivity	仮想マシン接続に必要な APIC の VMM インベントリ内のすべてのオブジェクトを読み取るのに使用されます。
vmm-ep	APIC の VMM インベントリ内の仮想マシンとハイパーバイザ エンドポイントを読み取るために使用されます。
vmm-policy	仮想マシン ネットワーキングのポリシー管理に使用されます。
vmm-protocol-ops	VMM ポリシーでは使用されません。

Role: vmm-admin	
権限	説明
vmm-security	VMware vCenter のユーザー名やパスワードなど VMM 認証ポリシーの管理に使用されます。

カスタム ロール

カスタムロールを作成し、ロールに権限を割り当てることができます。インターフェイスは、すべての管理対象オブジェクトクラスに1つ以上の権限を内部的に割り当てます。XML モデルで、権限はアクセス属性に割り当てられています。権限のビット数は、コンパイル時に割り当てられ、クラスのインスタンスまたはオブジェクトごとではなく、クラスごとに適用されます。

45 権限ビットだけでなく、「aaa」権限ビットはすべての AAA サブシステムの設定と読み取り操作に適用されます。次の表は、サポートされている権限の組み合わせの一覧を提供します。表の行は Cisco Application Centric Infrastructure (ACI) モジュールを表し、列は特定のモジュールの機能を表します。セルの「o」の値は、モジュールがアクセス可能な機能と、機能にアクセスするための権限ビットが存在することを示します。空のセルは、権限ビットでアクセスできないモジュールの特定の機能を示します。権限ビットについての詳細は、各ビットの機能について参照してください。

	Connectivity	QoS	セキュリティ	アプリケーション	Fault	Stats	Provider	サービスプロファイル	サービスチェーン
VMM	対応		対応		対応	対応	対応		
ファブリック	対応	対応	対応	対応	対応	対応	対応		
External	対応	対応	対応		対応	対応			対応
テナント	対応	対応	対応	EPG、NP	対応	対応			対応
Infra	対応	対応	対応	対応	対応	対応			対応
操作					対応	対応			
ストレージ	対応	対応	対応	対応	対応	対応			

	Connectivity	QoS	セキュリティ	アプリケーション	Fault	Stats	Provider	サービスプロファイル	サービスチェーン
ネットワークサービス	対応	対応	対応	対応	対応	対応		対応	

複数のセキュリティドメイン間で物理リソースを選択的に公開する

ファブリック全体の管理者は、RBAC規則を使用して、異なるセキュリティドメインにあるため他の方法ではアクセス不可能なユーザに対し、物理リソースを選択的に公開します。

たとえば、ソーラーというテナントのユーザが仮想マシン管理（VMM）ドメインへのアクセスを必要とする場合、ファブリック全体の管理者によって、これを許可するRBAC規則を作成することができます。RBAC規則は、次の2つの部分から構成されます。アクセス対象オブジェクトを検索する識別名（DN）と、オブジェクトにアクセスするユーザを含むセキュリティドメインの名前です。したがって、この例では、ソーラーというセキュリティドメイン内の指定ユーザがログインすると、このルールにより、VMMドメインおよびツリーの内の子オブジェクトすべてへのアクセスが許可されます。VMMドメインへのアクセスを複数のセキュリティドメイン内のユーザに許可するには、ファブリック全体の管理者は、セキュリティドメインそれぞれについて、VMMドメインのDNとセキュリティドメインを含むRBAC規則を作成します。



- (注) 管理情報ツリー内の異なる部分に存在するユーザに対し、RBAC規則によりオブジェクトを公開することは可能ですが、CLIの使用によってツリーの構造を横断することでそのようなオブジェクトに移動することはできません。ただし、RBAC規則に含まれるオブジェクトのDNをユーザが把握していれば、ユーザはMO検索コマンドにより、CLIを使用してそれを見つけることができます。

複数のセキュリティドメイン間でのサービス共有を有効にする

ファブリック全体の管理者は、RBAC規則を使用して、テナント間の共有サービスを可能にするトランステナントEPG通信をプロビジョニングします。

APIC ローカル ユーザ

管理者は、外部AAAサーバを使用しないことを選択し、APIC自体でユーザを設定することができます。これらのユーザは、APIC ローカル ユーザと呼ばれます。

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。

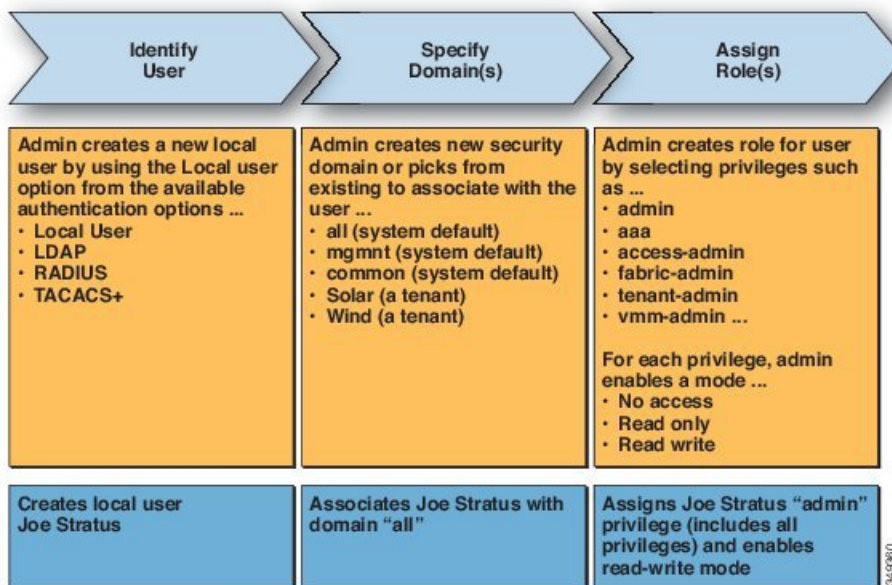
- 連続して繰り返される文字は 3 文字未満です。
- 小文字、大文字、数字、記号の文字種のうち少なくとも 3 種類の文字が含まれている必要があります。
- 簡単に推測できるパスワードは使用しません。
- ユーザ名やユーザ名を逆にしたものは使用できません。
- cisco、isco、またはこれらの文字列の並べ替えを変化させたものや、それらの文字の大文字化の変更により取得される変形語であってはなりません。

また APIC により、管理者は、外部で管理されている認証 Lightweight Directory Access Protocol (LDAP)、RADIUS、TACACS+、または SAML サーバで設定されたユーザにアクセス権を付与できます。ユーザは、異なる認証システムに属し、APIC に同時にログインできます。

さらに、30 秒ごとに変更したワンタイムパスワードはローカルユーザの OTP を有効にできます。OTP を有効にすると、APIC は、ランダムな人間判読可能な 162 進数オクテット base32 OTP キーであるを生成します。この OTP キーは、ユーザの OTP を生成するために使用します。

次の図は、ACI ファブリック全体へのフルアクセス権があるローカル APIC 認証データベース内の管理ユーザを設定するプロセスがどのように動作するかを示します。

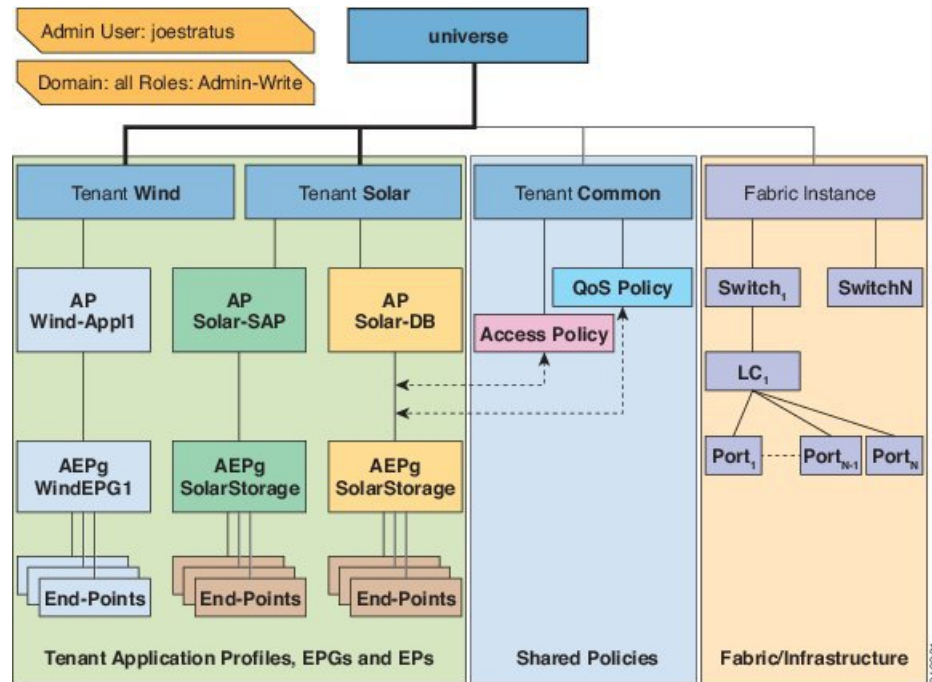
図 1: APIC ローカル ユーザの設定プロセス



(注) セキュリティ ドメイン「all」は、管理情報ツリー (MIT) 全体を表します。このドメインには、システム内のすべてのポリシーと APIC によって管理されるすべてのノードが含まれます。テナント ドメインには、テナントのすべてのユーザおよび管理対象オブジェクトが含まれます。テナント管理者には、「all」ドメインへのアクセス権を付与しないでください。

次の図は、管理ユーザ Joe Stratus が持つシステムへのアクセス権を示します。

図 2: 「all」ドメインへ管理ユーザを設定した結果

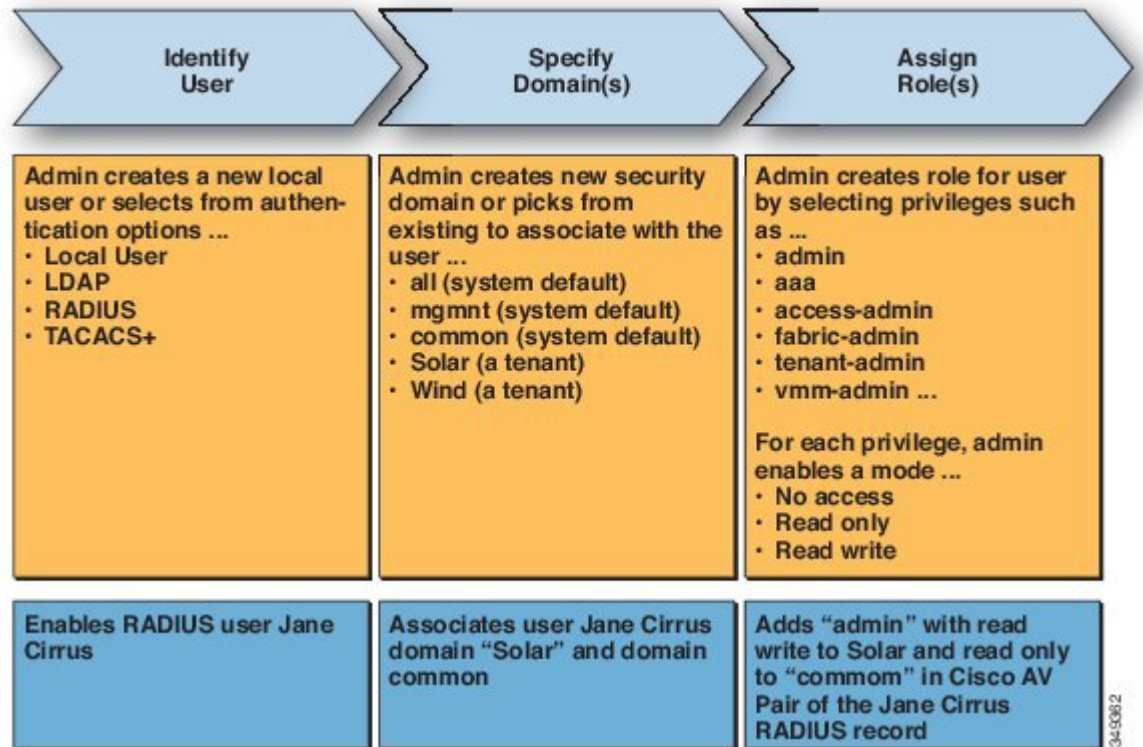


読み取り/書き込み「管理者」権限を持つユーザ Joe Stratus は、ドメイン「all」に割り当てられ、システム全体へのフルアクセス権が与えられます。

外部管理されている認証サーバのユーザ

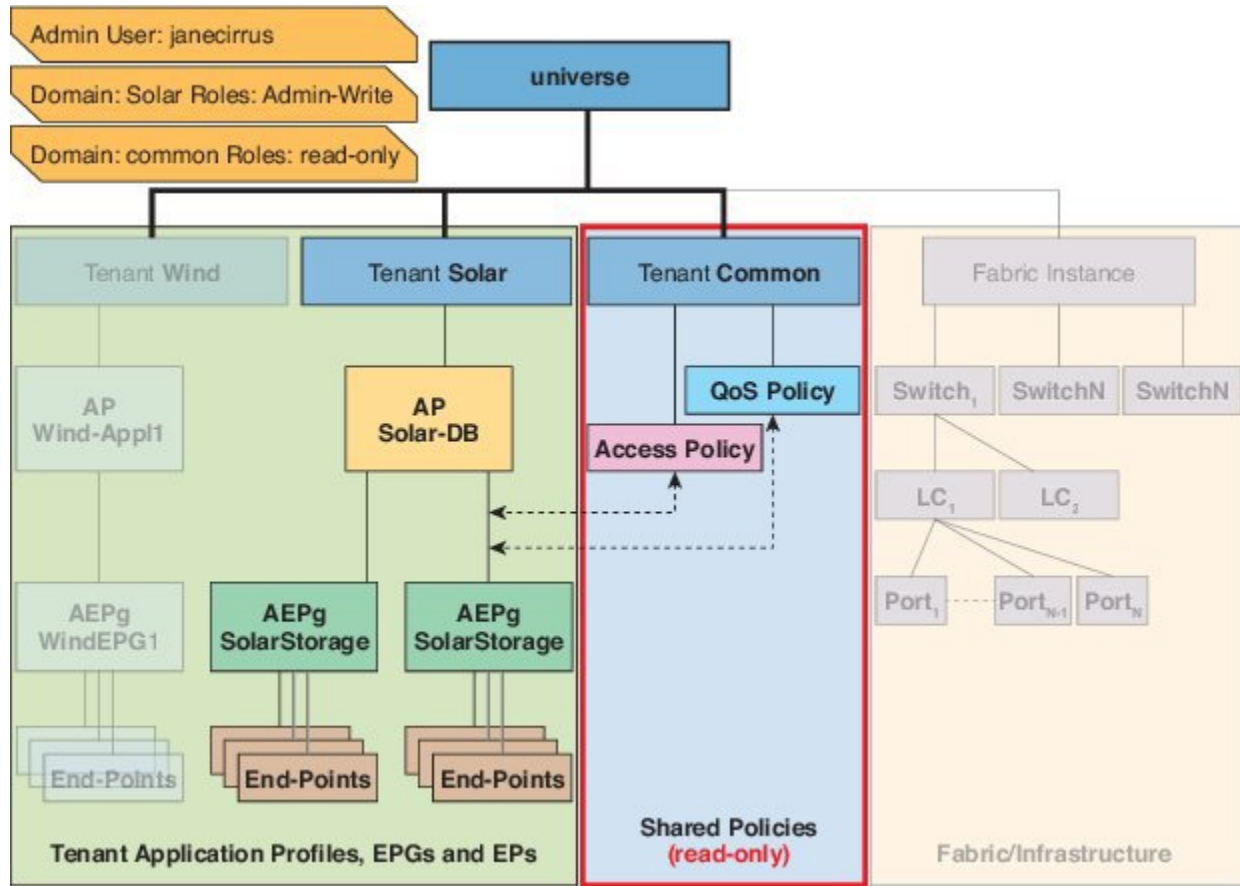
次の図は、テナント Solar へのフルアクセス権がある外部 RADIUS サーバ内の管理ユーザを設定するプロセスがどのように動作するかを示します。

図 3: 外部認証サーバでのユーザ設定のプロセス



次の図は、管理ユーザ Jane Cirrus が持つシステムへのアクセス権を示します。

図 4: テナント **Solar** へ管理ユーザを設定した結果



この例では、Solar テナントの管理者には、Solar テナントに含まれるすべてのオブジェクトへのフルアクセス権と、テナント Common への読み取り専用アクセス権があります。テナント管理者 Jane Cirrus には、テナント Solar へのフルアクセス権があり、テナント Solar で新しいユーザを作成する機能などがあります。テナントユーザは、自身が所有し制御する ACI ファブリックの設定パラメータを変更できます。また、エンドポイント、エンドポイントグループ (EPG) およびアプリケーションプロファイルなどの適用されるエンティティ (管理対象オブジェクト) の統計情報の読み取り、障害およびイベントのモニタもできます。

上記の例では、ユーザ Jane Cirrus は外部 RADIUS 認証サーバで設定されました。外部認証サーバで AV ペアを設定するには、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、APIC 上のユーザに対するロールベースアクセスコントロール (RBAC) のロールと権限を指定します。次に RADIUS サーバは、ユーザ権限を APIC コントローラに伝播します。

上記の例のオープン RADIUS サーバ (/etc/raddb/users) の設定は次のとおりです。

```
janecirrus Cleartext-Password := "<password>"
Cisco-avpair = "shell:domains = solar/admin/,common//read-all(16001)"
```

この例には、次の要素が含まれています。

- janecirrus はテナント管理者です。

- solar はテナントです。
- admin は書き込み権限があるロールです。
- common は、テナント共通サブツリーで、すべてのユーザがそのサブツリーへの読み取り専用アクセス権を持っています。
- read-all は、読み取り権限があるロールです。

Cisco AV ペアの形式

Cisco APIC は、管理者が外部認証サーバで Cisco AV ペアを設定し、1 個の AV ペアの文字列のみを検索することを要求しています。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。

AV ペア文字列を機能させるため、次の形式にする必要があります。

```
shell:domains =
ACI_Security_Domain_1/ACI_Write_Role_1|ACI_Write_Role_2|ACI_Write_Role_3/ACI_Read_Role_1|ACI_Read_Role_2,
ACI_Security_Domain_2/ACI_Write_Role_1|ACI_Write_Role_2|ACI_Write_Role_3/ACI_Read_Role_1|ACI_Read_Role_2,
ACI_Security_Domain_3/ACI_Write_Role_1|ACI_Write_Role_2|ACI_Write_Role_3/ACI_Read_Role_1|ACI_Read_Role_2
```

- **shell:domains=** : ACI が正常に文字列を読み取るために必要です。シェル文字列を常にブリーペンドする必要があります。
- **ACI_Security_Domain_1/admin** : 管理者にこのセキュリティドメインのテナントへの読み取り専用アクセス権を付与します。
- **ACI_Security_Domain_2/admin** : 管理者にこのセキュリティドメインのテナントへの書き込みアクセス権を付与します。
- **ACI_Security_Domain_3/read-all** : このセキュリティドメインのテナントへの読み取り/書き込みすべてのアクセス権を付与します。



(注) /により区別される文字列のセキュリティドメイン、書き込み、読み取りセクション同じセキュリティドメイン内の | により区別される複数の書き込みまたは読み取り権限



(注) Cisco APIC リリース 2.1 より、AV ペアに UNIX ID が指定されていない場合、APIC は UNIX の固有ユーザー ID を内部的に割り当てます。

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=ACI_Security_Domain_1/Write_Role_1|Write_Role_2/
```

- 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=Security_Domain_1//Read_Role_1|Read_Role_2
```



(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。

AV ペア GUI の設定

セキュリティ ドメインは、[Admin] > [AAA] > [Security Management] > [Security Domains] の ACI GUI で定義されており、[テナント] > [Tenant_Name] > [ポリシー] のテナントに割り当てられています。

セキュリティドメインには読み取りまたは書き込み権限のいずれかが必須です。これらの権限は、[APIC] > [Admin] > [Security Management] > [Roles] で定義されています。権限が書き込みセクションに入力される場合、ACI_Security_Domain_1/admin/admin/admin を使用する必要がないため、自動的に同じレベルの読み取り権限を付与します。

リモート ユーザー ロールの変更

ユーザー権限を「動的」に変更可能で、ユーザーがロール変更の要求を行うことが可能になり、ローカルまたはリモートで保存されている情報に基づいて、要求ロールが許可または拒否されます。

ロール変更は次の 2 種類の方法で行うことができます。

- ログインの日付/時間に基づくロールの割り当て
- 明示的な「要求」に基づくロールの割り当て

ACI ファブリックは、Radius、TACACS +、LDAP プロトコルを使用して外部認証をサポートします。上記の両方の方法で、リモート認証サーバにロール変更機能をサポートするコンポーネントが含まれていると仮定します。

Cisco Secure ACS サーバは、Radius/TACACS+ および LDAP プロトコルのリモート認証、認証、アカウンティング機能を提供します。

デフォルト デバイス管理またはデフォルト ネットワーク アクセス サービスのどちらかにルールが一致する必要があります。

認証で、別のルール設定が設定されています。

- **AVPairOps** : tacacs + ユーザー名および AVPair 値と一致します (cisco-av-pair*newrole)。ルールに一致すると、ACI_OPS シェル プロファイルが返されます
- **NoAVPair** : tacacs + ユーザー名のみ一致し、一致で ACI_ADMIN シェル プロファイルを返します
- **opsuser** : プロトコルのみ一致し、ACI_OPS シェル プロファイルを返します

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ー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。

アカウントティング

ACI ファブリック アカウントティングは、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せん。

`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記録を上書きします。



(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

クシヨ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トングポリシーを設定できます。

設定 (Configuration)

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+ と TACACS+ プロバイダー グループの作成。
- 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 a11 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、a11 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。

手順

-
- 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]** ペインで、**[AAA Authentication]** をクリックします。
- ステップ 3** **[Work]** ペインのデフォルトの **[Authentication]** フィールドで、**[Realm]** フィールドが **[Local]** と表示されていることを確認します。
- ステップ 4** **[Navigation]** ペインで、**[Security Management] > [Local Users]** を展開します。
管理ユーザはデフォルトで存在しています。
- ステップ 5** **[Navigation]** ペインで、**[Create Local User]** を右クリックします。
- ステップ 6** **ユーザ アイデンティティ** ダイアログボックス、入力、**ログイン ID** および **パスワード** ユーザ、および] をクリック 次。
- ステップ 7** **[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 8** **[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 9** **[User Identity]** ダイアログボックスで、次の操作を実行します。
- [Login ID]** フィールドで、**ID** を追加します。
 - [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。

- [Confirm Password] フィールドで、パスワードを確認します。
- [Finish] をクリックします。

ステップ 10 [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。

手順

ステップ 1 メニューバーで、[ADMIN] > [Security Management] > [Local Users] の順に選択します。

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。

- [Name] フィールドにキーの名前を入力します。
- [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) リモートの場所にダウンロードするための SSH 秘密キー ファイルを作成するには、メニューバーで、[Firmware] > [Download Tasks] を展開します。

NX-OS スタイル CLI を使用したローカルユーザの設定

手順

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

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>
```

X.509 証明書と秘密キーの生成

手順

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]

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

手順

- 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]** ペインで、**[AAA Authentication]** をクリックします。
- ステップ 3 **[Work]** ペインのデフォルトの **[Authentication]** フィールドで、**[Realm]** フィールドが **[Local]** と表示されていることを確認します。
- ステップ 4 **[Navigation]** ペインで、**[Security Management]** > **[Local Users]** を展開します。
管理ユーザはデフォルトで存在しています。
- ステップ 5 **[Navigation]** ペインで、**[Local Users]** をクリックし、**[Create Local User]** 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 **[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) **[Finish]** をクリックします。
- ステップ 9 **[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 10 [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

手順

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-----\nMIICjjCAafegAwIBAgIJAMQnbE <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

手順

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

手順

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 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 3 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+OTqK0CeAZj17+Gute2R1Ww8OGgtzE0wsLlx8fIXX14V79Z17
Ou8IdJH9CB4W6CEvdICXqkv3KaQszCIC0+Bn07o3qF//BsIpl2mYChD6gCX3f7q
IcJGX+R6HAqGeK7k97cNhX1WEoobFPe/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 4 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+OTqK0CeAZj17+Gute2R1Ww8OGgtzE0wsLlx8f
IXX14V79Z17Ou8IdJH9CB4W6CEvdICXqkv3KaQszCIC0+Bn07o3qF//BsIpl2mYChD6gCX3f
7qIcJGX+R6HAqGeK7k97cNhX1WEoobFPe/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 5 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。
