Cisco APIC リリース 4.0(1) セキュリティ設定ガイド

初版：2018 年 10 月 24 日

シスコシステムズ合同会社
〒107-6227 東京都港区赤坂9-7-1 ミッドタウン・タワー
http://www.cisco.com/jp
お問い合わせ先：シスコ コンタクトセンター
0120-092-255 （フリーコール、携帯・PHS含む）
電話受付時間：平日 10:00～12:00、13:00～17:00
http://www.cisco.com/jp/go/contactcenter/

目 次

はじめに

この前書きは、次の項で構成されています。

対象読者

このガイドは、次の 1 つ以上に責任を持つ、専門知識を備えたデータセンター管理者を主な対象にしています。

- 仮想マシンのインストールと管理
- サーバ管理
- スイッチおよびネットワークの管理

表記法

コマンドの説明には、次のような表記法が使用されます。

表記法	説明
bold	太字の文字は、表示どおりにユーザが入力するコマンドおよびキーワードです。
italic	イタリック体の文字は、ユーザが値を入力する引数です。
[x]	省略可能な要素（キーワードまたは引数）は、角カッコで囲んで示しています。

表記法	説明
[x | y]	いずれか1つを選択できる省略可能なキーワードや引数は、角カッコで囲み、縦棒で区切って示しています。
{x | y}	必ずいずれか1つを選択しなければならない必須キーワードや引数は、波カッコで囲み、縦棒で区切って示しています。
[x {y | z}]	角カッコまたは波カッコが入れ子になっている箇所は、任意または必須の要素内の任意または必須の選択肢であることを表します。角カッコ内の波カッコと縦棒は、省略可能な要素内で選択すべき必須の要素を示しています。
`variable`	ユーザが値を入力する変数であることを表します。イタリック体が使用できない場合に使用されます。
string	引用符を付けない一組の文字。string の前後には引用符を使用しません。引用符を使用すると、その引用符も含めて string とみなされます。

例では、次の表記法を使用しています。

表記法	説明
`screen フォント`	スイッチが表示する端末セッションおよび情報は、screen フォントで示しています。
`太字の screen フォント`	ユーザが入力しなければならない情報は、太字の screen フォントで示しています。
イタリック体の *screen* フォント	ユーザが値を指定する引数は、イタリック体の screen フォントで示しています。
< >	パスワードのように出力されない文字は、山カッコ（< >）で囲んで示しています。
[ ]	システム プロンプトに対するデフォルトの応答は、角カッコで囲んで示しています。
!、#	コードの先頭に感嘆符（!）またはポンド記号（#）がある場合には、コメント行であることを示します。

このマニュアルでは、次の表記法を使用しています。

（注） 「注釈」です。役立つ情報やこのマニュアルに記載されていない参照資料を紹介しています。

注意 「要注意」の意味です。機器の損傷またはデータ損失を予防するための注意事項が記述されています。

警告 安全上の重要な注意事項

「危険」の意味です。人身事故を予防するための注意事項が記述されています。機器の取り扱い作業を行うときは、電気回路の危険性に注意し、一般的な事故防止対策に留意してください。各警告の最後に記載されているステートメント番号を基に、装置に付属の安全についての警告を参照してください。

これらの注意事項を保存しておいてください。

関連資料

Application Policy Infrastructure Controller（APIC）のマニュアル

次のガイドでは、APIC のドキュメントを提供します。

- 『*Cisco APIC Getting Started Guide*』
- 『*Cisco APIC Basic Configuration Guide*』
- 『*Cisco ACI Fundamentals*』
- 『*Cisco APIC Layer 2 Networking Configuration Guide*』
- 『*Cisco APIC Layer 3 Networking Configuration Guide*』
- 『*Cisco APIC NX-OS Style Command-Line Interface Configuration Guide*』
- 『*Cisco APIC REST API Configuration Guide*』
- 『*Cisco APIC Layer 4 to Layer 7 Services Deployment Guide*』
- 『*Cisco ACI Virtualization Guide*』
- 『*Cisco Application Centric Infrastructure Best Practices Guide*』

これらすべてのドキュメントは、次の URL で入手できます。 http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html

シスコ アプリケーション セントリック インフラストラクチャ（ACI）のマニュアル

ACI の各種マニュアルは、次の URL から入手できます。http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html

シスコ アプリケーション セントリック インフラストラクチャ（ACI）シミュレータのマニュアル

Cisco ACI Simulator のマニュアルは、次の URL から入手できます：http://www.cisco.com/c/en/us/support/cloud-systems-management/application-centric-infrastructure-simulator/tsd-products-support-series-home.html

Cisco Nexus 9000 シリーズ スイッチのマニュアル

Cisco Nexus 9000 シリーズ スイッチのマニュアルは、次の URL で入手できます。
http://www.cisco.com/c/en/us/support/switches/nexus-9000-series-switches/tsd-products-support-series-home.html

Cisco Application Virtual Switch のマニュアル

Cisco Application Virtual Switch（AVS）のマニュアルは、次の URL で入手できます。
http://www.cisco.com/c/en/us/support/switches/application-virtual-switch/tsd-products-support-series-home.html

シスコ アプリケーション セントリック インフラストラクチャ（ACI）と OpenStack の統合に関するマニュアル

Cisco ACI と OpenStack の統合に関するマニュアルは、次の URL から入手できます。
http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html

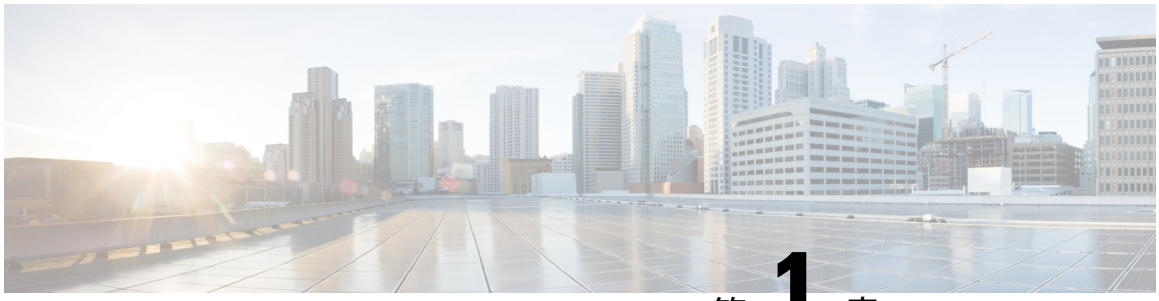
マニュアルに関するフィードバック

このマニュアルに関する技術的なフィードバック、または誤りや記載もれなどお気づきの点がございましたら、apic-docfeedback@cisco.com までご連絡ください。ご協力をよろしくお願いいたします。

マニュアルの入手方法およびテクニカル サポート

マニュアルの入手方法、Cisco バグ検索ツール（BST）の使用法、テクニカル サポートの依頼方法、および追加情報の収集方法については、『*What's New in Cisco Product Documentation*』（http://www.cisco.com/c/en/us/td/docs/general/whatsnew/whatsnew.html）を参照してください。

『*What's New in Cisco Product Documentation*』では、シスコの新規および改訂版の技術マニュアルの一覧を、RSS フィードとして購読できます。また、リーダー アプリケーションを使用して、コンテンツをデスクトップに直接配信することもできます。RSS フィードは無料のサービスです。



第 1 章

新機能および変更された機能に関する情報

この章の内容は、次のとおりです。

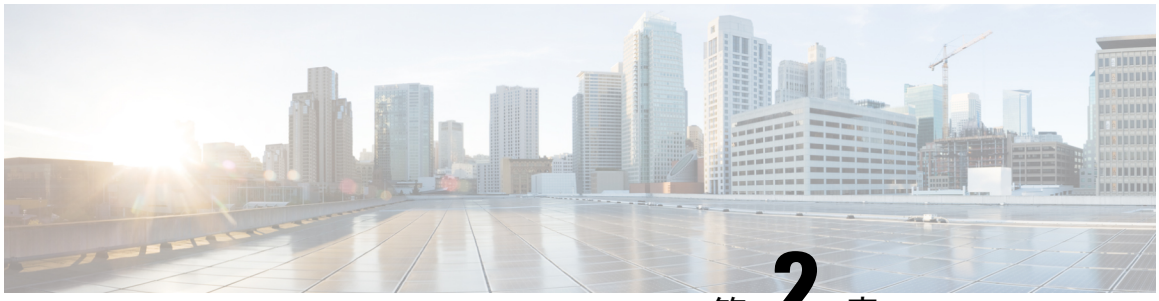
新機能および変更された機能に関する情報

次の表は、現行リリースに至るまでにガイドの編成と特徴に加えられた主な変更点の概要を示しています。ただし、今リリースまでのガイドにおける変更点や新機能の一部は表に記載されていません。

表 *1 : Cisco APIC* リリース *4.0(1)* の新機能と変更された動作

機能または変更	説明	参照先
証明書ベースの認証	証明書ベースの認証のサポート。	「アクセス、認証、およびアカウンティング」、「*HTTP*アクセス」の各章

新機能および変更された機能に関する情報



第 2 章

概要

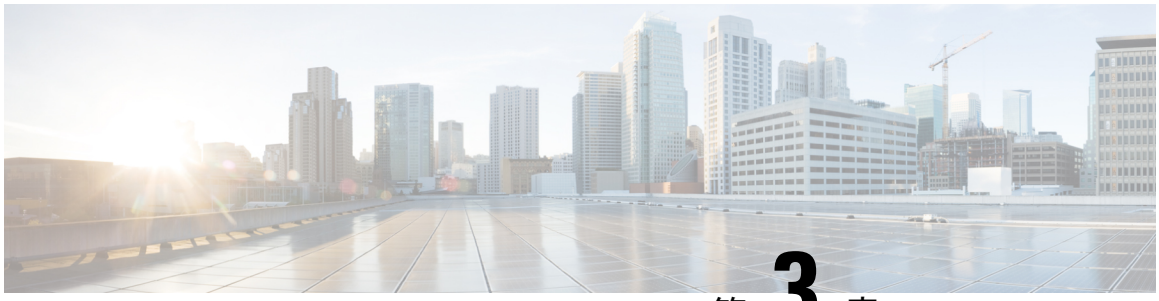
この章の内容は、次のとおりです。

概要

Cisco ACI がサポートするセキュリティ機能を利用すると、ネットワークをパフォーマンスの劣化や障害から保護するだけでなく、故意に行われる攻撃や、善意のネットワーク ユーザーの意図しない危険な間違いにより生ずるデータの紛失または毀損に対しても保護できます。

コア ファブリック サービスに関する詳細は、http://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/2-x/basic_config/b_APIC_Basic_Config_Guide_2_x/b_APIC_Basic_Config_Guide_2_x_chapter_011.html を参照してください。

概要



第 3 章

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。

概要

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） 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	特権	説明
aaa	aaa	ポリシーの認証、許可、アカウンティング、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	
特権	説明
access-connectivity-l1	インフラでレイヤ 1 の設定に使用します。例：セレクタとポート 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：セレ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	
特権	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：セレクタおよびポート 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	
特権	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-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	特権	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	
特権	説明
ops	設定されているポリシーの表示に使用されます（ポリシーのトラブルシューティングなど）。 （注） **Ops** ロールは、新しいモニタリングポリシーおよびトラブルシューティングポリシーの作成には使用できません。これらのポリシーは、APIC の他のすべての設定と同様に、**admin** 権限を使用して作成する必要があります。

Role: read-all	
特権	説明
access-connectivity-l1	インフラでレイヤ 1 の設定に使用します。例：セレクタとポート 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：セレ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: read-all	
特権	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：セレクタおよびポート 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-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	設定されているポリシーの表示に使用されます（ポリシーのトラブルシューティングなど）。 （注） **Ops** ロールは、新しいモニタリング ポリシーおよびトラブルシューティング ポリシーの作成には使用できません。これらのポリシーは、APIC の他のすべての設定と同様に、**admin** 権限を使用して作成する必要があります。
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。

Role: read-all	
特権	説明
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、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。

Role: read-all	
特権	**説明**
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: tenant-admin	
特権	**説明**
aaa	ポリシーの認証、許可、アカウンティング、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：セレクタとポート 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：セレ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。

Role: tenant-admin	
特権	説明
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：セレクタおよびポート 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。
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。

Role: tenant-admin	
特権	説明
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	設定されているポリシーの表示に使用されます（ポリシーのトラブルシューティングなど）。 （注） **Ops** ロールは、新しいモニタリング ポリシーおよびトラブルシューティング ポリシーの作成には使用できません。これらのポリシーは、APIC の他のすべての設定と同様に、**admin** 権限を使用して作成する必要があります。
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、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。

Role: tenant-admin	
特権	説明
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。
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。

Role: tenant-admin	
特権	説明
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-ext-admin	
特権	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。
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、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。

Role: tenant-ext-admin	
特権	説明
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。
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	
特権	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。

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「○」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	○		○		○	○	○		
ファブリック	○	○	○	○	○	○	○		
External	○	○	○		○	○			○
テナント	○	○	○	EPG、NP	○	○			○
Infra	○	○	○	○	○	○			○
操作					○	○			

	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
ストレージ	○	○	○	○	○	○			
ネットワークサービス	○	○	○	○	○	○		○	

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。

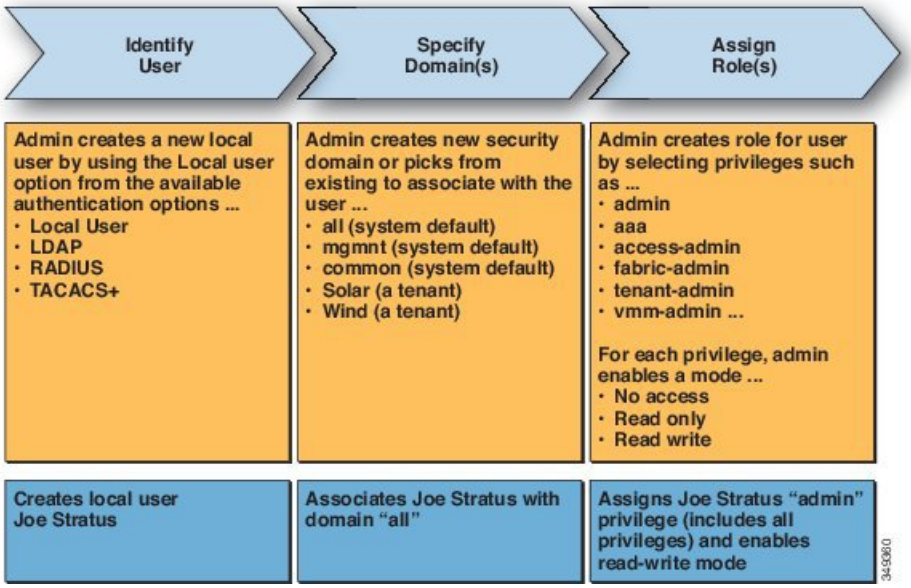
- 連続して繰り返される文字は 3 文字未満です。
- 小文字、大文字、数字、記号の文字種のうち少なくとも 3 種類の文字が含まれている必要があります。
- 簡単に推測できるパスワードは使用しません。
- ユーザ名やユーザ名を逆にしたものは使用できません。
- cisco、isco、またはこれらの文字列の並べ替えを変化させたものや、それらの文字の大文字化の変更により取得される変形語であってはなりません。

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 16 2 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。

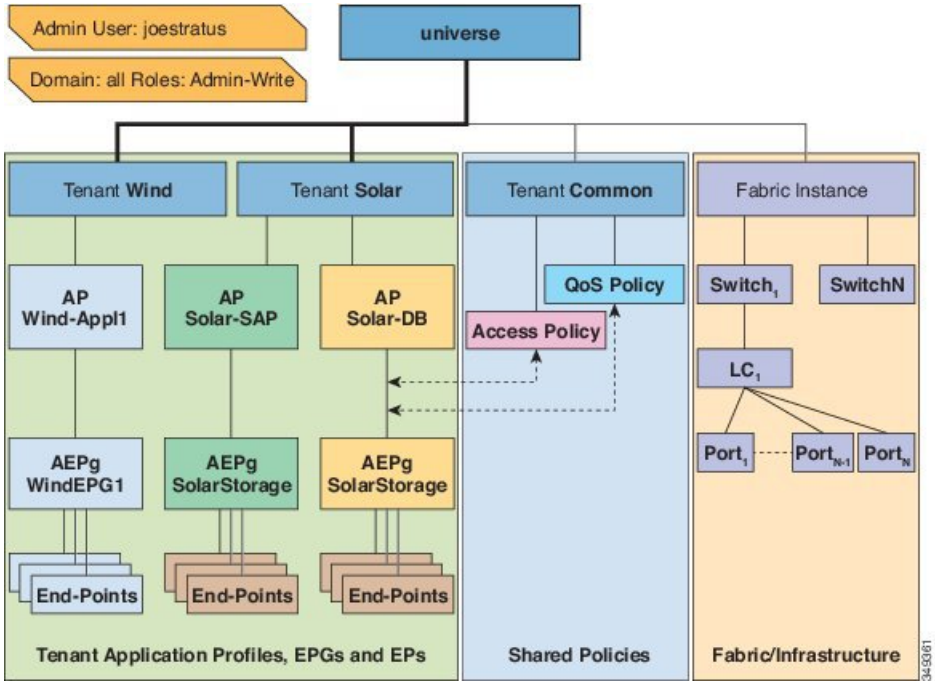
図 *1 : APIC* ローカル ユーザの設定プロセス



（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 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

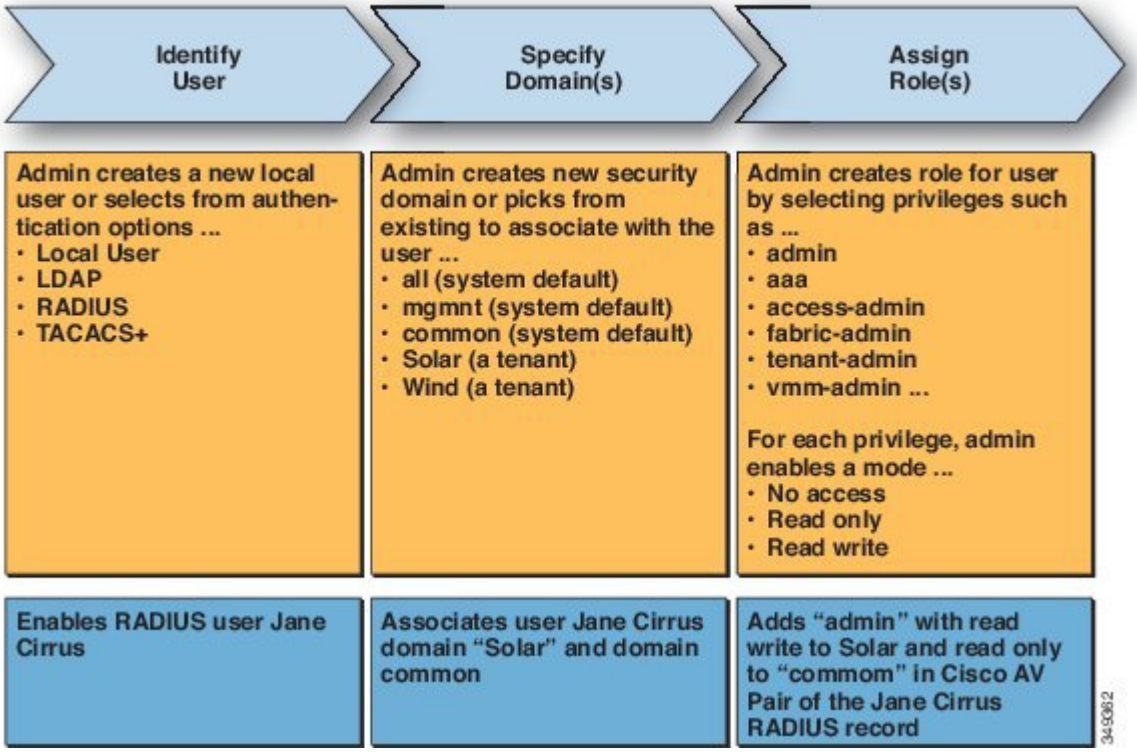


読み取り/書き込み「管理者」権限を持つユーザ Joe Stratus は、ドメイン「all」に割り当てられ、システム全体へのフル アクセス権が与えられます。

外部管理されている認証サーバのユーザ

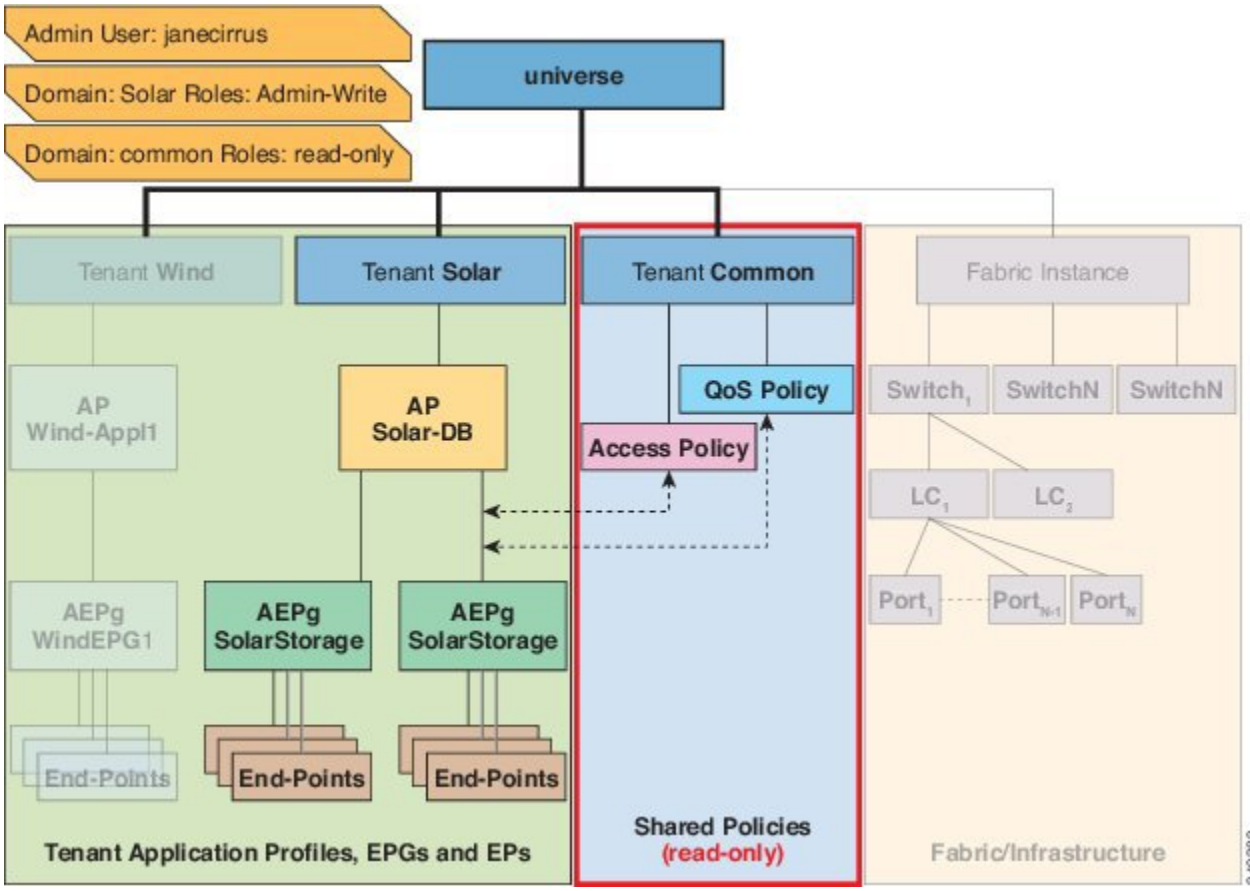
次の図は、テナント Solar へのフル アクセス権がある外部 RADIUS サーバ内の管理ユーザを設定するプロセスがどのように動作するかを示します。

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プリ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 Cisco ACS サーバ経由でのみサポートされており、明示的な「要求」に基づくロールの割り当てによって実行できます。

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、TACACS+プロトコルのリモート認証、認証、およびアカウンティングの機能を提供します。

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

GUI を使用したリモート ユーザ ロールの変更

始める前に

ロールは、最初に AVPair と一致するように Cisco ASC サーバで設定し、その一致に基づいてシェル認証プロファイルとして選択する必要があります。

手順

ステップ 1 ASC 認証ポリシーを作成します。**[Access Policies]** > **[Access Services]** > **[Default Device Admin Identity]** に移動し、次の手順を実行します。

（注） シェル プロファイルが CiscoAVPair を使用して設定され、ユーザの認証に使用されます。

a) [TACACS+:AVPair equals cisco-av-pair*] に条件を追加し、[OK] をクリックします。

（注） デフォルトでは、ユーザは **cisco-av-pair** ロールを使用して認証されます。

b) [TACACS+:AVPair equals cisco-av-pair*readall] に条件を追加し、[OK] をクリックします。

（注） APIC でキーワード **readall** を使用して、ロールを **default** ロールから **readall** ロールに変更します（シェル プロファイルで read-all が設定されます）。

ステップ 2 APIC GUI にログインし、[welcome, <ログイン名>] ドロップダウン リストをクリックして、[Change Remote User Role] を選択します。

ステップ 3 [Change Remote User Role] ダイアログボックスで、[User Name]、[Password]、[New Role] の各フィールドに情報を入力し、[Submit] をクリックします。

GUI が更新され、新しいロールが適用されます。

（注） 親ロールに戻るには、もう一度 [Change Remote User Role] ダイアログボックスを開き、[User Name] と [Password] に情報を入力しますが、[New Role] フィールドは空欄のままにしておきます。

REST API を使用したリモート ユーザ ロールの変更

始める前に

ロールは、最初に AVPair と一致するように Cisco ASC サーバで設定し、その一致に基づいてシェル認証プロファイルとして選択する必要があります。

ユーザは、ユーザ名 **apicadmin** とパスワードでログインします。

手順

ステップ 1 新しいロールに変更します。

例：

```
<!--   api/requestNewRole/json   -->
<aaaChangeRole>
<attributes userName="apic#tacacs” apicadmin="pwd Ins3965!” role=“newrole"/>
```

ステップ 2 元のロールに戻ります。

例：

```
<!--   api/requestNewRole/json   -->
<aaaChangeRole>
<attributes userName="apic#tacacs” apicadmin="pwd Ins3965!” role=“”/>
```

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。

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。

設定

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。

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] ペインで、[Local Users] タブを表示していることを確認します。

デフォルトでは admin ユーザが存在します。

ステップ 4 [Work] ペインで、タスク アイコンのドロップダウン リストをクリックし、[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、認証証明書からのユーザ 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。

手順

ステップ 1 メニュー バーで、**[ADMIN]** > **[Users]** を選択し、[Local Users] タブを表示し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） 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] ペインで、[Local Users] タブを表示していることを確認します。

デフォルトでは admin ユーザが存在します。

ステップ 4 [Work] ペインで、タスク アイコンのドロップダウン リストをクリックし、[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、認証証明書からのユーザ 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

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-----\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

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 payload.txt 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.txt 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。

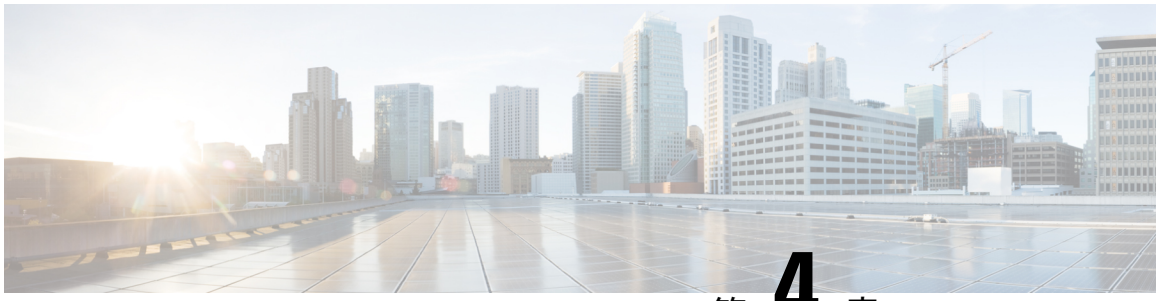
次のスクリプトは、ACI Python SDK の CertSession クラスを使用して、署名を使用して APIC に要求する方法の例です。

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。



第 4 章

TACACs +、RADIUS、LDAP、RSA、SAML

この章の内容は、次のとおりです。

概要

この記事は、RADIUS、TACACS＋、LDAP ユーザーが APIC にアクセスできるようにする手順を説明します。読者が Cisco プリケーションセントリック インフラストラクチャの基礎マニュアル、特にユーザーアクセス権、認証、アカウンティングの章を十分に利害していると仮定しています。

（注） セキュリティ上の理由により、AAA 認証に shell:domains=all/read-all/ を使用するリモート ユーザは、ファブリック内のリーフ スイッチおよびスパイン スイッチにアクセスすることはできません。このことは、4.0(1h) までのすべてのバージョンに当てはまります。

RADIUS

RADIUS サーバでユーザを設定するには、APIC 管理者は cisco-av-pair 属性を使用して必要な属性（shell:domains）を設定する必要があります。デフォルトのユーザ ロールは、network-operator です。

SNMPv3 認証プロトコルに指定できるオプションは、SHA と MD5 です。プライバシー プロトコルに指定できるオプションは、AES-128 と DES です。これらのオプションが cisco-av-pair 属性で指定されていない場合は、MD5 および DES がデフォルトの認証プロトコルとなります。

たとえば、SNMPv3 認証とプライバシー プロトコルの属性は次のように指定できます。

```
snmpv3:auth=SHA priv=AES-128
```

同様に、ドメインのリストは次のとおりです。

```
shell:domains="domainA domainB …"
```

TACACS+ 認証

Terminal Access Controller Access Control device Plus（TACACS+）は、シスコ デバイスでサポートされる別のリモート AAA プロトコルです。TACACS+ には、RADIUS 認証にはない次の利点があります。

- 独立した AAA ファシリティを提供する。たとえば、APICは、認証を行わずにアクセスを許可できます。
- AAA クライアントとサーバ間のデータ送信に TCP を使用しているため、コネクション型プロトコルによる確実な転送が可能になります。
- スイッチと AAA サーバ間でプロトコル ペイロード全体を暗号化して、高度なデータ機密性を実現します。RADIUS はパスワードのみを暗号化します。
- 構文と設定が RADIUS と異なる av-pairs を使用しますが、APIC は shell:domains をサポートします。

（注） TACACS サーバおよび TACAC ポートは、ping で到達可能である必要があります。

次に示す XML の例では、IP アドレス 10.193.208.9 の TACACS+ プロバイダーを ACI ファブリックに使用させるよう設定が行われています。

（注） この例では IPv4 アドレスを使用していますが、IPv6 アドレスも使用できます。

```
<aaaTacacsPlusProvider name="10.193.208.9"
           key="test123"
           authProtocol=”pap”/>
```

APIC Bash シェルのユーザ ID

APIC での Linux シェル用のユーザ ID は、ローカル ユーザ用に APIC 内で生成されます。認証クレデンシャルが外部サーバで管理されているユーザは、Linux シェル用のユーザ ID を cisco-av-pair で指定できます。上記の cisco-av-pair の（16001）を省略することは、リモート ユーザがデフォルトの Linux ユーザ ID 23999 を取得すれば可能です。Linux ユーザ ID がバッシュ セッション中に使用され、標準の Linux 権限が適用されます。また、ユーザが作成するすべての管理対象オブジェクトは、そのユーザの Linux ユーザ ID によって作成されたとマークされます。

次に、APIC Bash シェルに表示されるユーザ ID の例を示します。

```
admin@ifav17-ifc1:~> touch myfile
admin@ifav17-ifc1:~> ls -l myfile
-rw-rw-r-- 1 admin admin 0 Apr 13 21:43 myfile
admin@ifav17-ifc1:~> ls -ln myfile
-rw-rw-r-- 1 15374 15374 0 Apr 13 21:43 myfile
admin@ifav17-ifc1:~> id
uid=15374(admin) gid=15374(admin) groups=15374(admin)
```

ログイン ドメイン

ログイン ドメインは、ユーザの認証ドメインを定義します。ログイン ドメインは、ローカル、LDAP、RADIUS、または TACACS+ 認証メカニズムを設定できます。REST、CLI、または GUI からシステムにアクセスすると、APIC によりユーザは正しい認証ドメインを選択できます。

たとえば、REST シナリオでは、完全なログイン ユーザ名が次のように表示されるようにユーザ名の頭に文字列が付きます。

```
apic:<domain>\<username>
```

システムに GUI からアクセスする場合は、APIC により選択するユーザのドメインのドロップダウン リストが提供されます。`apic: domain` が指定されない場合は、デフォルトの認証ドメイン サーバがユーザ名の検索に使用されます。

ACI バージョン 1.0(2x) 以降、APIC のログイン ドメイン フォールバックのデフォルトはローカルになっています。デフォルト認証とコンソール認証方法がどちらも非ローカルの方法に設定されており、両方の非ローカル方法がローカル認証に自動的にフォールバックしない場合でも、APIC にはローカル認証を使用してアクセスすることができます。

APIC フォールバック ローカル認証にアクセスするには、次の文字列を使用します。

- GUI からは、*apic:fallback\\username* を使用します。
- REST API からは、*apic#fallback\\username* を使用します。

（注） フォールバック ログイン ドメインは変更しないでください。変更すると、システムからロックアウトされる可能性があります。

LDAP/Active Directory の認証

RADIUS および TACACS+ と同様、LDAP により、ネットワーク要素はユーザを認証し、特定のアクションの実行を許可するために使用できる AAA クレデンシャルを取得できます。追加された認証局の設定は管理者によって実行でき、LDAPS（SSL 経由の LDAP）の信頼性をイネーブルにし、中間者攻撃を防ぐことができます。

次に示す XML の例では、ACI ファブリックが IP アドレス 10.30.12.128 の LDAP プロバイダーを使用するように設定しています。

（注） この例では IPv4 アドレスを使用していますが、IPv6 アドレスも使用できます。

```
<aaaLdapProvider name="10.30.12.128"
          rootdn="CN=Manager,DC=ifc,DC=com"
          basedn="DC=ifc,DC=com"
          SSLValidationLevel="strict"
          attribute="CiscoAVPair"
          enableSSL="yes"
          filter="cn=$userid"
          port="636" />
```

（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。

Cisco AVPair を設定する代わりに、APIC で LDAP グループ マップを作成するオプションがあります。

RSA Secure ID 認証

RSA認証は、使用できる組み合わせで固定キーを使用して、パスワードを作成するさまざまな方法でトークンを提供します。これは、ハードウェア トークンとソフトウェア トークンの両方をサポートします。

GUI を使用して、RSA 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。

- RSA 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。

手順

ステップ 1 APIC で、RSA プロバイダを作成します。

a) メニュー バーで、**[Admin]** > **[AAA]** の順に選択します。
b) [Navigation] ペインで、**[RSA Managment]** > **[RSA Providers]** の順に選択します。
c) [Work] ペインで、**[Actions]** > **[Create RSA Provider]** の順に選択します。
d) RSA 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。

ステップ 2 RSA プロバイダー グループを作成します。

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

ステップ 3 RSA 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 RSA 設定手順は完了です。次に、RSA サーバを設定します。

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

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。

手順

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 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。

手順

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。

手順

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

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。

LDAP の設定

LDAP 設定には 2 つのオプションがあります。Cisco AVPair を設定したり、APIC 内で LDAP グループマップを設定したりできます。このセクションには、両方の設定オプションの手順が含まれています。

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。

手順

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。

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

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）ファブリックがインストールされていて、Application Policy Infrastructure 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い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。

手順

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

Cisco APIC での LDAP グループ マップ ルールの設定

Cisco APIC での LDAP グループ マップの設定には、作成の最初の LDAP グループ マップ ルールが必要です。このセクションでは、LDAP グループ マップ ルールを作成する方法について説明します。

始める前に

LDAPサーバが設定されているグループのマッピングを実行しています。

手順

ステップ 1 Cisco APIC GUI のメニュー バーで、**[Admin]** > **[AAA]** を選択します。

ステップ 2 [Navigation] ペインで、 [LDAP Managment] を展開し、[LDAP Group Map Rules] を右クリックして、 [Create LDAP Group Map Rule] をクリックします。[Create LDAP Group Map Rule: Security] ダイアログが表示されます。

ステップ 3 該当するフィールドにマップ ルールの名前、説明 (オプション)、グループの DN、およびセキュリティ ドメインを指定し、[Next] をクリックします。セキュリティ ドメイン オプションが表示された [Create LDAP Group Map Rule: Roles] ダイアログが表示されます。

ステップ 4 [+] をクリックして、[Role Name] および [Role Privilege Type] フィールドにアクセスします。

ステップ 5 [Role Name] ドロップダウン矢印をクリックして、ロール名を選択します。

ステップ 6 [Role Privilege Type] ドロップダウン矢印をクリックして、ロール権限のタイプを選択します ([Read] または [Write])。

ステップ 4 ～ 6 を繰り返して、LDAP グループ マップに他のロールを追加します。

ステップ 7 完了したら、[Finished] をクリックします。

次のタスク

LDAP グループ マップ ルールを指定した後に、LDAP グループ マップを作成します。

Cisco APIC での LDAP グループ マップの設定

Cisco APIC での LDAP グループ マップの設定には、作成の最初の LDAP グループ マップ ルールが必要です。このセクションでは、LDAP グループ マップを作成する方法について説明します。

始める前に

- 実行中の LDAPサーバは、グループ マッピングで設定されます。
- LDAP グループ マップ ルールが設定されています。

手順

ステップ 1 Cisco APIC GUI のメニュー バーで、**[Admin]** > **[AAA]** を選択します。

ステップ 2 [Navigation] ペインで、 [LDAP Managment] を展開し、[LDAP Group Maps] を右クリックして、[Create LDAP Group Map] をクリックします。[Create LDAP Group Map] ダイアログが表示されます。

ステップ 3 マップの名前と説明（オプション）を指定します。

ステップ 4 [Rules] フィールドで、[+] をクリックしてから、 [Name] ドロップダウン矢印をクリックして、指定した LDAP グループ マップ ルールを選択し、[Update] をクリックします。

ステップ 4 を繰り返して、LDAP グループ マップに他のルールを追加します。

ステップ 5 完了したら、[送信（Submit）] をクリックします。

NX-OS スタイル CLI を使用したリモート 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。

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。

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

手順

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。

手順

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
```

SAML について

SAML は XML ベースのオープン規格のデータ形式であり、いずれかのアプリケーションにサインインした後に、管理者は定義された一連のシスコのコラボレーション アプリケーションにシームレスにアクセスできます。SAML では、信頼できるビジネス パートナー間で、セキュリティに関連した情報交換を記述します。これは、サービス プロバイダーによってユーザの認

証に使用される認証プロトコルです。SAML により、ID プロバイダー（IdP）とサービス プロバイダーの間で、セキュリティ認証情報を交換できます。

SAML SSO は SAML 2.0 プロトコルを使用して、シスコのコラボレーション ソリューションのドメイン間と製品間で、シングル サインオンを実現しています。SAML 2.0 は、Cisco アプリケーション全体で SSO を有効にし、Cisco アプリケーションと IdP 間でフェデレーションを有効にします。SAML 2.0 では、高度なセキュリティ レベルを維持しながら、シスコの管理ユーザが安全なウェブ ドメインにアクセスして、IdP とサービス プロバイダーの間でユーザ認証と承認データを交換できます。この機能が安全なメカニズムを提供していることで、さまざまなアプリケーションにわたり、共通の資格情報や関連情報を使用します。

SAML SSO の管理者権限は、シスコのコラボレーション アプリケーションでローカルに設定されたロールベース アクセス コントロール（RBAC）に基づき認証されます。

SAML SSO は、IdP とサービス プロバイダーの間のプロビジョニング プロセスの一部として、メタデータと証明書を交換することで信頼の輪（CoT）を確立します。サービス プロバイダーは IdP のユーザ情報を信頼しており、さまざまなサービスやアプリケーションにアクセスできるようにします。

（注） サービス プロバイダーが認証にかかわることはありません。SAML 2.0 では、サービス プロバイダーではなく、IdP に認証を委任します。

クライアントは IdP に対する認証を行い、IdP はクライアントにアサーションを与えます。クライアントはサービス プロバイダーにアサーションを示します。CoT が確立されているため、サービス プロバイダーはアサーションを信頼し、クライアントにアクセス権を与えます。

SAML SSO を有効にすると、次のような利点が得られます。

- 別のユーザ名とパスワードの組み合わせを入力する必要がなくなるため、パスワードの劣化が軽減されます。
- アプリケーションをホストしているお使いのシステムからサードパーティのシステムに、認証を転送します。SAML SSO を使用することで、IdP とサービス プロバイダーの間で信頼の輪を作成できます。サービス プロバイダーは IdP 信頼して、ユーザを認証します。
- 認証情報を保護し、安全に保ちます。暗号化機能により、IdP、サービス プロバイダー、ユーザの間で認証情報を保護します。SAML SSO では、IdP とサービス プロバイダー間で転送される認証メッセージを外部ユーザから保護することもできます。
- 同じ ID に資格情報を再入力する時間が省けるため、生産性が向上します。
- パスワードをリセットするためのヘルプデスクへの問い合わせが減るため、コスト削減につながります。

SAML の基本要素

- クライアント（ユーザのクライアント）：これは、認証用にブラウザインスタンスを活用できる、ブラウザベースのクライアントまたはクライアントです。システム管理者のブラウザはその一例です。
- サービスプロバイダー：これは、クライアントがアクセスを試みるアプリケーションまたはサービスです。
- ID プロバイダー（IdP）サーバ：これは、ユーザ資格情報を認証し、SAML アサーションを発行するエンティティです。
- Lightweight Directory Access Protocol（LDAP）ユーザ：これらのユーザは、Microsoft Active Directory や OpenLDAP などの LDAP ディレクトリと統合されます。非 LDAP ユーザは、Unified Communications サーバ上にローカルに存在します。
- SAML アサーション：これは、ユーザ認証のために、IdP からサービス プロバイダーに転送されるセキュリティ情報で構成されます。アサーションは、ユーザ名や権限などのサブジェクトに関する信頼されたステートメントを含む、XML ドキュメントです。通常では、信頼性を確保するために、SAML アサーションはデジタル署名されます。
- SAML 要求：これは、Unified Communications アプリケーションにより生成される認証要求です。LDAP ユーザを認証するために、Unified Communications アプリケーションは認証要求を IdP に委任します。
- 信頼の輪（CoT）：これは、共同で 1 つの IdP に対して共有と認証を行うさまざまなサービス プロバイダーで構成されます。
- メタデータ：これは、IdP と同様に ACI アプリケーションによって生成された、XML ファイルです。SAML メタデータの交換により、IdP とサービス プロバイダーの間に信頼関係が確立します。
- Assertion Consumer Service（ACS）URL：この URL は、アサーションをポストする場所を IdP に指示します。ACS URL は、最終的な SAML 応答を特定の URL にポストすることを IdP に指示します。

（注） 認証が必要なすべてのインスコープ サービスでは、SSO のメカニズムとして SAML 2.0 を使用します。

サポートされている IdPs および SAML コンポーネント

サポートされる IdP

ID プロバイダー（IdP）は、ユーザ、システム、サービスの ID 情報を作成、維持、管理する認証モジュールです。また、分散ネットワーク内のその他のアプリケーションやサービスプロバイダーに対して認証も行います。

SAML SSO で、IdPs はユーザーのロールまたは各 Cisco コラボレーション アプリケーションのログイン オプションに基づいて、認証オプションを提供します。IdP は、ユーザ資格情報を保管、検証し、ユーザがサービス プロバイダーの保護リソースにアクセスできる SAML 応答を生成します。

（注） IdP サービスを十分理解している必要があります。現在インストールされていて、操作可能であることを確認してください。

APIC の SAML SSO 機能は、次の IdP でテストされています。

- https://technet.microsoft.com/en-us/library/cc772128(WS.10).aspx
- Okta シングル サインオン： https://www.okta.com/products/single-sign-on/
- PingFederate： https://documentation.pingidentity.com/pingfederate/pf90/index.shtml#gettingStartedGuide/concept/gettingStarted.html

SAML のコンポーネント

SAML SSO ソリューションは、特定のアサーション、プロトコル、バインディング、プロファイルの組み合わせに基づきます。さまざまなアサーションは、プロトコルやバインディングを使用しているアプリケーション間やサイト間で交換され、これらのアサーションによりサイト間でユーザを認証します。SAML のコンポーネントは次のとおりです。

- SAML アサーション：これは、IdP からサービス プロバイダーに転送される情報の構造と内容を定義します。セキュリティ情報のパケットで構成され、さまざまなレベルのアクセス コントロール決定にサービス プロバイダの用途があることを示す文書が含まれます。SAML SSO は次の種類の文書を提供します。
  - 認証ステートメント：これらのステートメントは、IdP とブラウザの間で特定の時間に行う認証の方法について、サービス プロバイダーにアサートします。
  - 属性ステートメント：これらのステートメントは、ユーザに関連付ける特定の属性（名前と値のペア）についてアサートします。属性アサーションには、ユーザに関する具体的な情報が含まれます。サービスプロバイダーは、属性を使用してアクセス制御の決定を行います。
- SAML プロトコル：SAML プロトコルは、SAML がアサーションをどのように要求し、取得するかを定義します。このプロトコルは、特定の SAML エレメントまたはアサーションで構成されている、SAML 要求と応答エレメントに対応します。SAML 2.0 には次のプロトコルがあります。
  - アサーション クエリと要求のプロトコル
  - 認証要求のプロトコル
- SAML バインディング：SAML バインディングは、SOAP 交換のような、標準メッセージング形式または通信プロトコルとの SAML アサーションまたはプロトコル メッセージ（ま

たはその両方）の交換のマッピングを指定します。ACI は次の SAML 2.0 バインディングをサポートしています。

- HTTP Redirect（GET）バインディング
- HTTP POST バインディング

- SAMLプロファイル：SAMLプロファイルでは、明確に定義された使用事例をサポートするために、SAMLアサーション、プロトコル、およびバインディングの組み合わせについて詳細に説明しています。

NTP の設定

SAML SSO で、Network Time Protocol（NTP）では APIC および IdP 間のクロック同期が可能です。SAML は時間的な制約のあるプロトコルであり、IdP は SAML アサーションが有効であることを時間ベースで判断します。IdP および APIC クロックが同期されていない場合、アサーションが無効になり SAML SSO 機能が停止します。IdP および APIC の間で許可される最大時差は 3 秒です。

（注） SAML SSO を動作させるには、NTP 設定を正しくインストールする必要があり、IdP と APIC アプリケーション間の時間差が 3 秒を超えていないことを確認する必要があります。IdP および APIC クロックが同期されていない場合、ユーザーは IdP で認証に成功した後でも APIC のログイン ページにリダイレクトされます。

DNS の設定

Domain Name System（DNS）により、ホスト名とネットワーク サービスをネットワーク（複数可）内の IP アドレスにマッピングできるようになります。ネットワーク内に配置された DNS サーバは、ネットワーク サービスをホスト名にマッピングし、次にホスト名を IP アドレスにマッピングするデータベースを備えています。ネットワーク上のデバイスは、DNS サーバに照会して、ネットワークにある他のデバイスの IP アドレスを受信できます。そのため、ネットワーク デバイス間の通信が容易になります。

まとめると、APIC および Idp は互いの完全修飾ドメイン名を IP アドレスに対して解消でき、クライアントによって解消される必要があります。

Certificate Authority：認証局

シスコは、次のいずれかの種類の認証局（CA）により署名されるサーバ証明書を使用することを推奨します。

- **パブリック CA**：サードパーティ企業が、サーバの識別情報を確認し、信頼できる証明書を発行します。
- **プライベート CA**：自身でローカルの CA を作成および管理し、信頼できる証明書を発行します。

署名プロセスは製品ごとに異なり、サーバのバージョン間でも異なる場合があります。各サーバのすべてのバージョンに関する詳細な手順については、このマニュアルの範囲外になります。CA により署名された証明書を取得する方法の詳細な手順については、該当するサーバのマニュアルを参照してください。

パブリック CA により署名されたサーバ証明書を取得する場合、パブリック CA は、クライアントコンピュータの信頼ストアで、ルート証明書をあらかじめ提示しておくようにします。この場合、クライアントコンピュータでルート証明書をインポートする必要はありません。プライベート CA など、CA により署名される証明書が信頼ストアにまだ存在しない場合は、ルート証明書をインポートしてください。SAML SSO では、CN または SAN での正しいドメインが記載された CA 署名付き証明書が、IdP およびサービス プロバイダーに必要になります。正しい CA 証明書が検証されない場合、ブラウザはポップ アップ警告を出します。

APIC の信頼ストアに IdP のルート証明書が含まれていない場合は、新しい証明機関を作成する必要があります。APIC で SAML プロバイダを設定する際は、この認証機関を後で使用する必要があります。

SAML アクセス用の APIC の設定

（注） SAML ベースの認証と CLI/REST の APIC GUI でのみです。また、リーフ スイッチと背表紙には適用されません。APIC CLI では、SAML 設定を行うことはできません。

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。
- SAML サーバ ホスト名または IP アドレスと、IdP メタデータの URL を使用できます。
- APIC 管理エンドポイント グループを使用できること。
- 次の設定を行います。
  - 時刻同期と NTP：https://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/3-x/basic_config/b_APIC_Basic_Config_Guide_3_x/b_APIC_Basic_Config_Guide_3_x_chapter_011.html#concept_9CE11B84AD78486AA7D83A7DE1CE2A77。
  - 拡張 GUI を使用した DNS プロバイダーと接続するための DNS サービス ポリシーの設定：https://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/3-x/basic_config/b_APIC_Basic_Config_Guide_3_x/b_APIC_Basic_Config_Guide_3_x_chapter_011.html#task_750E077676704BFBB5B0FE74628D821E。
  - GUI を使用した Cisco ACI HTTPS アクセス用カスタム証明書の設定：https://www.cisco.com/c/en/us/td/docs/switches/datacenter/aci/apic/sw/3-x/basic_config/b_APIC_Basic_Config_Guide_3_x/b_APIC_Basic_Config_Guide_3_x_chapter_011.html#task_F037F1B75FF74ED1BCA4F3C75A16C0FA。

手順

ステップ 1 APIC で、**SAML 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、**[SAML Management]** > **[SAML Providers]** を選択します。
c) [Work] ペインで、**[Actions]** > **[Create SAML Provider]** を選択します。
d) SAML ホスト名 (または IP アドレス) と IdP メタデータ URL を指定します。

- AD FS の場合、IdP メタデータ URL は *https://<ADFS の FQDN>/FederationMetadata/2007-06/FederationMetadata.xml* という形式になります。
- Okta の場合、IdP メタデータの URL を取得するには、Okta サーバから該当 SAML アプリケーションの [Sign On] セクションに、**アイデンティティ プロバイダー メタデータ**のリンクをコピーします。

e) SAML ベースのサービスの**エンティティ ID** を指定します。
f) IdP メタデータの URL にアクセスする必要がある場合は、Https プロキシを設定します。
g) IdP はプライベート CA によって署名された場合は、認証局を選択します。
h) ドロップダウン リストから、ユーザの要求の署名アルゴリズムの認証タイプを選択します。

ステップ 2 **SAML プロバイダー グループ**を作成します。

a) [Navigation] ペインで、**[SAML Management]** > **[SAML Providers Groups]** を選択します。
b) [Work] ペインで、**[Actions]** > **[Create SAML Provider Group]** を選択します。
c) 必要に応じて、SAML 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 SAML 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。

Okta で SAML アプリケーションの設定

Okta で SAML を設定するには、管理者特権を持つユーザーとして Okta 組織にログインします。

（注） Okta 組織をお持ちでない場合、空の Okta を作成できます。

https://www.okta.com/start-with-okta/

手順

ステップ 1 Okta で、青色の **[管理者]** ボタンをクリックします。

ステップ 2 **[アプリケーションの追加]** ショートカットをクリックします。

ステップ 3 緑色の **[新しいアプリケーションの作成]** ボタンをクリックし、次の操作を行います。

a) **[新しいアプリケーションの作成]** ダイアログ ボックスで、**[SAML 2.0]** オプションを選択し、緑色の **[作成]** ボタンをクリックします。

b) **[全般設定]** ボックスで、**[例 SAML アプリケーション]** を、**[アプリケーション名]** フィールドに入力し、緑色の **[次へ]** ボタンをクリックします。

c) **[SAML の設定]** セクション A **[SAML 設定]** フィールドで、**[シングル サインオン URL]**、**[受信者 URL]**、**[対象者の制限]** フィールドに SAML URL を貼り付けます。

このフィールドは次の形式にする必要があります。

- https://<APIC_hostname>/api/aaaLoginSSO.json?name=<Login_domain_name> </Login_domain_name> </APIC_hostname>
- 要求可能な SSO URL を使用して APIC のクラスタを設定します。
  - https://<APIC1_hostname>/api/aaaLoginSSO.json?name=<Login_domain_name> </Login_domain_name> </APIC1_hostname>
  - https://<APIC2_hostname>/api/aaaLoginSSO.json?name=<Login_domain_name> </Login_domain_name> </APIC2_hostname>
  - https://<APIC3_hostname>/api/aaaLoginSSO.json?name=<Login_domain_name> </Login_domain_name> </APIC3_hostname>
- 名前 ID 形式：Transient
- 応答：署名済み
- アサーション署名: 署名
- アサーション暗号化: 暗号化されていません。
- SAML シングル ログアウト: Disabled
- authnContextClassRef: PasswordProtectedTransport
- SAML 発行者 ID: http://www.okta.com/$ {org.externalKey}

d) **[Attribute Statements]** セクションで、**[FirstName]**、**[LastName]**、**[Email]**、**[CiscoAvpair]** フィールドに情報を追加して、**[次へ]** をクリックします。

（注） **CiscoAvpair** と呼ばれるカスタム属性は **[プロファイル エディタ]** で Okta ユーザーを作成する必要があります。CiscoAvpair の詳細は、外部認証サーバの AV ペア（48 ページ） を参照してください。

e) **[フィードバック]** ボックスで、**[私は内部アプリケーションを追加する Okta 顧客です]** および **[これは私が作成した内部アプリケーションです]** を選択して、**[終了]** をクリックします。

ステップ 4 新しく作成した **[例 SAML アプリケーション]** アプリケーションの **[サイン オン]** が表示されます。このページを保存し、別のタブまたはブラウザウィンドウで開きます。SAML 設定の **[ID プロバイダ メタデータ]** をコピーするには、後でこのページに戻ります。

（注） メタデータのリンクをコピーするには、**[ID プロバイダ メタデータ]** リンクを右クリックして **[コピー]** を選択します。

AD FS で Relying Party Trust の設定

AD FS 管理コンソールで信頼当事者証明を追加します。

手順

ステップ 1 証明書利用者信頼を追加します。

a) AD FS サーバの AD FS 管理コンソールにログインし、**ADFS** > **Trust Relationships** > **Relying Party Trusts** の順に移動して、 [Add Relying Party Trust] を右クリックしてから [Start] をクリックします。

b) APIC 内で、対応するログイン ドメイン設定で利用できる [Download SAML Metadata] オプションを使用して生成されたメタデータ ファイルをインポートすることによって、[Enter data about the relying party manually] または [Import data about relying party from a file (skip the steps d, e, f and g)] を選択します。

c) [Display Name] に信頼当事者証明の任意の表示名を入力し、[Next] をクリックします。

d) AD FS プロファイルを選択し、[Next] をクリックします。

e) もう一度 [Next] をクリックします。

f) [Enable support for the SAML 2.0 Web SSO Protocol] を選択し、**信頼当事者 SAML2.0 SSO サービス の URL** として *https://<APIC_hostname>/api/aaaLoginSSO.json?name=<Login_domain_name>* と入力し、[Next] をクリックします。

g) **信頼当事者証明の識別子**として *https://<APIC_hostname>/api/aaaLoginSSO.json* 入力します。

h) [I do not want to configure multi-factor authentication settings for this relying party trust at this time] を選択し、[Next] をクリックします。

i) [Permit all users to access this relying party] を選択し、[Next] をクリックします。

j) [Open the Edit Claim Rules dialog for this relying party trust when the wizard closes] を選択し、[Close] をクリックします。

ステップ 2 次の**クレーム** ルールを追加します。

a) LDAP 属性をクレームとして送信します。

- [Edit Claim Rules] ウィンドウで、[Add Rule] をクリックします。
- [Claim Rule Template] で [Send LDAP attributes as Claims] を選択し、[Next] をクリックします。
- [Rule_Name] を入力し、[Attribute Store] として [Active Directory] を選択します。
- CiscoAvpair を格納するための予約済みユーザ属性を選択します（たとえば、[LDAP attribute type] として [Department] を選択し、それを [Outgoing Claim Manually Type] の [CiscoAvpair] にマッピングします）。
- [LDAP Attribute] で [E-Mail-Addresses] を選択し、それを [Outgoing Claim Type] の [E-mail Address] にマッピングして、[Finish] をクリックします。

b) 着信要求を変換します。

- [Edit Claim Rules] ウィンドウで再度 [Add Rule] をクリックし、[Transform an Incoming Claim as Claim Rule Template] を選択して、[Next] をクリックします。
- [Incoming claim type] として [E-Mail Address] を選択 します。
- [Outgoing claim type] として [Name ID] を選択します。
- [Outgoing name ID format] として [Transient Identifier] を選択します。

ステップ 3 APIC のクラスタを追加するには、複数の**信頼当事者証明**をセットアップするか、または 1 つの**信頼当事者証明**をセットアップしてから複数の**信頼当事者識別子** および **SAML アサーション コンシューマ エンドポイント** をそれに追加することができます。

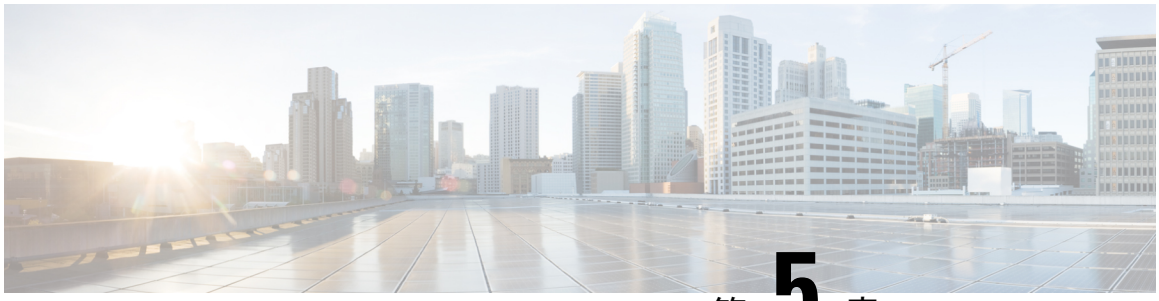
a) 上記で作成した同じ信頼当事者証明を持つクラスタ内に、他の APIC を追加する。

1. **ADFS Management Console** > **ADFS** > **Trust Relationships** > **Relying Party Trusts** と移動して、**CiscoAPIC** > **Properties** の順に右クリックします。
2. [Identifiers] タブをクリックし、クラスタ内に他の APIC を次のとおりに追加します：*https://<APIC2_hostname>/api/aaaLoginSSO.json*、*https://<APIC3_hostname>/api/aaaLoginSSO.json*
3. [Endpoints] タブをクリックし、[Add SAML] をクリックすることによって他の 2 つの APIC を追加します。[Add SAML Post Binding] 、[Index] を 1 として、信頼されている URL に *https://<APIC2_hostname>/api/aaaLoginSSO.json?name=<Login_domain_name>* のように入力します。そして、[Add SAML Post Binding] に *https://<APIC3_hostname>/api/aaaLoginSSO.json?name=<Login_domain_name* のように入力します。

ステップ 4 メッセージとアサーションは、ADFS サーバ内の powershell から ADFS で署名する必要があります。ADFS サーバでメッセージおよびアサーションを署名するには：

a) Windows Powershell を開き（管理者として実行する必要があります）、次のコマンドを実行します。

b) Set AdfsRelyingPartyTrust TargetName **RelyingpartytrustnameOfCiscoAPIC** -SamlResponseSignature **MessageAndAssertion** 。



第 5 章

802.1X

この章は、次の項で構成されています。

802.1X の概要

802.1X では、クライアント サーバ ベースのアクセス コントロールと認証プロトコルを定義し、許可されていないクライアントが公にアクセス可能なポートを経由して LAN に接続するのを規制します。認証サーバは、Cisco NX-OS デバイスのポートに接続されるクライアントを個々に認証します。

802.1X アクセス コントロールでは、クライアントが認証されるまで、そのクライアントが接続しているポート経由では Extensible Authentication Protocol over LAN（EAPOL）トラフィックしか許可されません。認証に成功すると、通常のトラフィックはポートを通過できるようになります。

RADIUS 分散クライアント/ サーバ システムを使用すると、不正アクセスからネットワークを保護できます。Cisco ACI 実装では、RADIUS クライアントは ToR で稼働し、すべてのユーザー認証情報およびネットワーク サービス アクセス情報が格納された中央の RADIUS サーバに認証要求およびアカウンティング要求を送信します。

ホスト サポート

802.1 X 機能は、次のモードでポート上のトラフィックを制限できます。

- **単一ホスト モード**：802.1Xポートで1台のエンドポイントデバイスのみからのトラフィックが許可されます。エンドポイント デバイスが認証されると、APIC はポートを許可状態にします。エンドポイント デバイスがログオフすると、OS はポートを無許可状態に戻します。802.1X のセキュリティ違反とは、認証に成功して許可された単一の MAC アドレスとは異なる MAC アドレスをソースとするフレームが検出された場合をいいます。このような場合、このセキュリティ アソシエーション（SA）違反（他の MAC アドレスからの EAPOL フレーム）が検出されたインターフェイスはディセーブルにされます。シングル ホスト モードは、ホストツースイッチ型トポロジで 1 台のホストが APIC のレイヤ 2 ポート（イーサネット アクセス ポート）またはレイヤ 3 ポート（ルーテッド ポート）に接続されている場合にだけ適用できます。
- **複数のホスト モード**：ポートごとに複数のホストを使用できますが、最初の 1 つだけが認証されます。最初のホストの許可に成功すると、ポートは許可ステートに移行します。ポートが許可ステートになると、後続のホストがネットワーク アクセスの許可を受ける必要はありません。再認証に失敗したり、または EAPOL ログオフ メッセージを受信して、ポートが無許可ステートになった場合には、接続しているすべてのクライアントはネットワーク アクセスを拒否されます。マルチ ホスト モードでは、SA 違反の発生時にインターフェイスをシャットダウンする機能がディセーブルになります。このモードは、スイッチツースイッチ型トポロジおよびホストツースイッチ型トポロジの両方に適用できます
- **マルチ認証モード**：複数のホストとすべてのホストを個別に認証を使用できます。

（注） 各ホストには、同じ EPG/VLAN 情報を必須です。

- **マルチ ドメイン モード**：別のデータおよび音声ドメイン。IP 電話で使用します。

認証モード

ACI 802.1 x は次の認証モードをサポートしています。

- **EAP**：オーセンティケータは EAP-Request/Identity フレームをサプリカントに送信して識別情報を要求します（通常、オーセンティケータは1つまたは複数の識別情報の要求のあとに、最初の Identity/Request フレームを送信します）。サプリカントはフレームを受信すると、EAP-Response/Identity フレームで応答します。
- **MAB**：フォールバック認証モードとして MAC 認証バイパス（MAB）がサポートされています。MAB により、エンドポイントの MAC アドレスを使用してポートベースのアクセス コントロールが有効になります。MAB が有効なポートは接続するデバイスの MAC アドレスに基づいて、動的に有効または無効にできます。MAB の前に、エンドポイントの ID が不明であり、すべてのトラフィックがブロックされます。スイッチでは、単一のパケットを検査して送信元 MAC アドレスを学習および認証します。MAB が成功するとエンドポイントの ID が判明し、エンドポイントからのすべてのトラフィックが許可されます。スイッチは送信元 MAC アドレス フィルタリングを実行し、MAB の認証されたエンドポイントのみがトラフィックの送信を許可されます。

注意事項と制約事項

802.1X ポートベースの認証には、次の設定に関する注意事項と制約事項があります。

- Cisco ACI が 802.1X 認証をサポートするのは、物理ポート上だけです。
- Cisco ACI は、ポート チャネルまたはサブインターフェイスでは 802.1X 認証をサポートしません。
- Cisco ACI は、ポート チャネルのメンバ ポートでは 802.1X 認証をサポートしますが、ポート チャネル自体ではサポートしません。
- 802.1X 設定を含むメンバ ポートと含まないメンバ ポートはポート チャネルで共存できます。ただし、チャネリングと 802.1X が連携して動作するためには、すべてのメンバ ポートで 802.1X 設定を同一にする必要があります
- 802.1X 認証を有効にした場合、サプリカントが認証されてから、イーサネット インターフェイス上のレイヤ 2 またはレイヤ 3 のすべての機能が有効になります。
- 802.1X は、EX または FX タイプのリーフ シャーシでのみサポートされています。
- 802.1X は、ファブリック アクセス ポートでのみサポートされています。802.1X は、ポート チャネルまたは仮想ポート チャネルではサポートされていません。
- IPv6 は、dot1x クライアント 3.2(1) リリースではサポートされていません。
- 特に特定のインターフェイス設定（ホストモードおよび認証タイプ）がそのリリースでサポートされていない場合に以前のリリースにダウングレードすると、dot1x 認証タイプはデフォルトでなしになります。ホストモードは希望に応じて単一のホストか複数のホストのどちらかに手動で再設定する必要があります。これで、ユーザーがそのリリースでのみサポートされているモード/認証タイプを設定し、サポートされていないシナリオで実行していないことを確認します。
- マルチ認証では、1 音声クライアントと複数のデータ クライアント（すべて同じデータ vlan/epg に属する）をサポートします。
- 802.1X ノード認証ポリシーでの障害 epg/vlan は必須設定です。
- 1 音声および 1 データ クライアント以上のマルチドメインは、ポートをセキュリティ無効の状態にします。
- 次のプラットフォームでは 802.1X はサポートされていません。
  - N9K-C9396PX
  - N9K-M12PQ
  - N9K-C93128TX
  - N9K-M12PQ

コンフィギュレーションの概要

APIC で有効になっている場合にのみ、802.1X および RADIUS プロセスが開始されます。内部的にこれは、radius エンティティの作成時に 802.1X Inst MO が作成され radius プロセスが作成されたときに、dot1x プロセスが開始されることを意味します。そのインターフェイスに接続しているユーザーを認証するため、Dot1x ベースの認証が各インターフェイスで有効になっている必要があります。そうでない場合、動作が変更されません。

RADIUS サーバの設定は、dot1x 設定とは別に行われます。RADIUS の設定は、RADIUS サーバのリストとそれらに到達する方法を定義します。Dot1x 設定には、認証に使用する RADIUS グループ（またはデフォルト グループ）への参照が含まれています。

正常に認証を行うには 802.1X と RADIUS の両方を設定する必要があります。設定の順序は重要ではありませんが、RADIUS 設定がない場合は、802.1X 認証は正常に行われません。

APIC GUI を使用した 802.1X ポート認証の設定

始める前に

RADIUS プロバイダのポリシーを設定します。

手順

ステップ 1 メニュー バーで、**[Fabric]** > **[External Access Policies]** > **[Policies]** > **[Interface]** > **[802.1X Port Authentication]** をクリックし、次の操作を行います。

a) [802.1X Port Authentication] を右クリックして、[Create 802.1X Port Authentication Policy] を開きます。

b) [Name] フィールドにポリシーの名前を入力します。

c) **[ホスト モード]** フィールドで、ポリシー モードを選択します。使用可能なモードを次に示します。

- **[マルチ認証]**：複数のホストおよびすべてのホストを個別に認証できます。

  （注） 各ホストには、同じ EPG/VLAN 情報が必須です。

- **[マルチ ドメイン]**：別のデータおよび音声ドメインです。IP 電話で使用します。
- **[マルチホスト]**：ポートごとに複数のホストを使用できますが、最初の 1 つだけが認証されます。
- **[単一ホスト]**：ポートごとに 1 個のホストのみ許可します。

d) デバイスが 802.1X をサポートしていない場合は、[MAC Auth] フィールドで [EAP_FALLBACK_MAB] を選択し、[Submit] をクリックします。

ステップ 2 **802.1X ポート認証ポリシー**をファブリック アクセス グループに関連付けるには、**[Fabric]** > **[External Access Policies]** > **[Interfaces]** > **[Leaf Interfaces]** > **[Policy Groups]** > **[Leaf Access Port]** に移動し、次の操作を行います。

a) **[リーフ アクセス ポート]** を右クリックして、**[リーフ アクセス ポート ポリシー グループの作成]** を開きます。

b) [Name] フィールドにポリシーの名前を入力します。

c) [802.1X Port Authentication Policy] フィールドで、以前に作成したポリシーを選択し、[Submit] をクリックします。

APIC GUI を使用した 802.1X ノード認証の設定

始める前に

RADIUS プロバイダのポリシーを設定します。

手順

ステップ 1 メニュー バーで、**[Fabric]** > **[External Access Policies]** > **[Policies]** > **[Switch]** > **[802.1X Node Authentication]** をクリックし、次の操作を行います。

a) [802.1X Node Authentication] を右クリックして、[Create 802.1X Node Authentication Policy] を開きます。

b) [Name] フィールドにポリシーの名前を入力します。

c) **[EPG 認証の失敗]** フィールドで、認証が失敗した場合に展開するテナント、アプリケーションプロファイル、EPG を選択します。

d) **[VLAN 認証の失敗]** で、認証が失敗した場合に展開する VLAN を選択します。

ステップ 2 **802.1X ノード認証ポリシー**をリーフ スイッチ ポリシー グループに関連付けるには、**[Fabric]** > **[External AccessPolicies]** > **[Switches]** > **[Leaf Switches]** > **[Policy Groups]** に移動し、次の操作を行います。

a) **[ポリシー グループ]** を右クリックして、**[アクセス スイッチ ポリシー グループの作成]** を開きます。

b) [Name] フィールドにポリシーの名前を入力します。

c) [802.1X Node Authentication Policy] フィールドで、以前に作成したポリシーを選択し、[Submit] をクリックします。

ステップ 3 **802.1X ノード認証ポリシー**をリーフ インターフェイス プロファイルに関連付けるには、**[Fabric]** > **[External AccessPolicies]** > **[Interfaces]** > **[Leaf Interfaces]** > **[Profiles]** に移動し、次の操作を行います。

a) **[プロファイル]** を右クリックして、**[リーフ インターフェイス プロファイルの作成]** を開きます。

b) [Name] フィールドにポリシーの名前を入力します。

c) **[インターフェイス セレクタ]** 表を展開し、**[アクセス ポート セレクタの作成]** ダイアログボックスを開き、**[名前]** および **[インターフェイス ID]** 情報を入力します。
d) **[インターフェイス ポリシー グループ]** フィールドで、以前に作成されたポリシーを選択し、**[OK]** および **[送信]** をクリックします。

NX-OS スタイル CLI を使用した 802.1X ポート認証の設定

手順

ステップ 1 ポリシー グループを設定します。

例：

```
apic1# configure
apic1(config)#
apic1(config)# template policy-group mypol
apic1(config-pol-grp-if)# switchport port-authentication mydot1x
apic1(config-port-authentication)# host-mode multi-host
apic1(config-port-authentication)# no shutdown
apic1(config-port-authentication)# exit
apic1(config-pol-grp-if)# exit
```

ステップ 2 リーフ インターフェイス ポリシーを設定します。

例：

```
apic1(config)#
apic1(config)#leaf-interface-profile myprofile
apic1(config-leaf-if-profile)#leaf-interface-group mygroup
apic1(config-leaf-if-group)# interface ethernet 1/10-12
apic1(config-leaf-if-group)# policy-group mypol
apic1(config-leaf-if-group)# exit
apic1(config-leaf-if-profile)# exit
```

ステップ 3 リーフ プロファイルを設定します。

例：

```
apic1(config)#
apic1(config)# leaf-profile myleafprofile
apic1(config-leaf-profile)# leaf-group myleafgrp
apic1(config-leaf-group)# leaf 101
apic1(config-leaf-group)# exit
```

ステップ 4 リーフ スイッチ プロファイルにインターフェイス ポリシーを適用します。

例：

```
apic1(config-leaf-profile)# leaf-interface-profile myprofile
apic1(config-leaf-group)# exit
```

NX-OS スタイル CLI を使用した 802.1X ノード認証の設定

手順

ステップ 1 Radius 認証グループを設定します。

例：

```
apic1# configure
apic1(config)#
apic1(config)# aaa group server radius myradiusgrp
apic1(config-radius)#server 192.168.0.100 priority 1
apic1(config-radius)#exit
```

ステップ 2 ノード レベル ポート認証ポリシーを設定します。

例：

```
apic1(config)# policy-map type port-authentication mydot1x
apic1(config-pmap-port-authentication)#radius-provider-group myradiusgrp
apic1(config-pmap-port-authentication)#fail-auth-vlan 2001
apic1(config-pmap-port-authentication)#fail-auth-epg tenant tn1 application ap1 epg
epg256
apic1(config)# exit
```

ステップ 3 ポリシー グループを設定し、グループ内でポート認証ポリシーを指定します。

例：

```
apic1(config)#template leaf-policy-group lpg2
apic1(config-leaf-policy-group)# port-authentication mydot1x
apic1(config-leaf-policy-group)#exit
```

ステップ 4 リーフ スイッチ プロファイルを設定します。

例：

```
apic1(config)# leaf-profile mylp2
apic1(config-leaf-profile)#leaf-group mylg2
apic1(config-leaf-group)#  leaf-policy-group lpg2
apic1(config-leaf-group)#exit
```

REST API を使用した 802.1X ポート認証の設定

手順

802.1X ポート認証ポリシーを作成します。

例：

```
<polUni>
<infraInfra>
  <l2PortAuthPol adminSt="enabled" annotation="" descr="" dn="uni/infra/portauthpol-test21"
 hostMode="multi-auth" name="test21" nameAlias="" ownerKey="" ownerTag="">
    <l2PortAuthCfgPol annotation="" macAuth="bypass" maxReauthReq="2" maxReq="2"
reAuthPeriod="3600" serverTimeout="30" suppTimeout="30" txPeriod="30"/>
  </l2PortAuthPol>
</infraInfra>
</polUni>

Modify:
<polUni>
<infraInfra>
  <l2PortAuthPol adminSt="enabled" annotation="" descr="" dn="uni/infra/portauthpol-test21"
 hostMode="multi-domain" name="test21" nameAlias="" ownerKey="" ownerTag="" >
    <l2PortAuthCfgPol annotation="" macAuth="eap" maxReauthReq="2" maxReq="2"
reAuthPeriod="3600" serverTimeout="30" suppTimeout="30" txPeriod="30"/>
  </l2PortAuthPol>
</infraInfra>
</polUni>

Delete:
<polUni>
<infraInfra>
  <l2PortAuthPol adminSt="enabled" annotation="" descr="" dn="uni/infra/portauthpol-test21"
 hostMode="multi-host" name="test21" nameAlias="" ownerKey="" ownerTag="" status="deleted">

    <l2PortAuthCfgPol annotation="" macAuth="bypass" maxReauthReq="2" maxReq="2"
reAuthPeriod="3600" serverTimeout="30" suppTimeout="30" txPeriod="30" status="deleted"/>

  </l2PortAuthPol>
</infraInfra>
</polUni>
```

REST API を使用した 802.1X ノード認証の設定

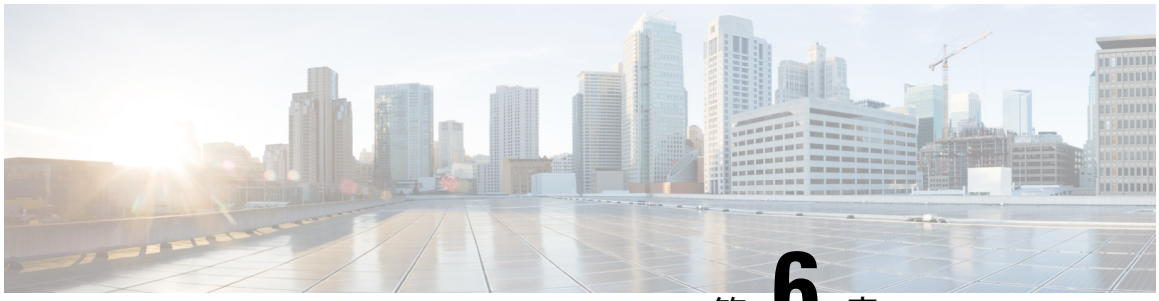
手順

802.1X ノード認証ポリシーを設定します。

例：

```
<polUni>
<infraInfra>
  <l2NodeAuthPol annotation="" descr="" dn="uni/infra/nodeauthpol-802-node-2"
failAuthEpg="tn-t2,ap-ap,epg-epg1" failAuthVlan="vlan-2078" name="802-node-2" nameAlias=""
 ownerKey="" ownerTag="">
<l2RsAaaRadiusProviderGroup annotation=""
tDn="uni/userext/radiusext/radiusprovidergroup-radius-grp"/>
</l2NodeAuthPol>
</infraInfra>
</polUni>

Modify:
<polUni>
<infraInfra>
```

```
  <l2NodeAuthPol annotation="" descr="" dn="uni/infra/nodeauthpol-802-node-2"
failAuthEpg="tn-t2,ap-ap,epg-epg1" failAuthVlan="vlan-2066" name="802-node-2" nameAlias=""
 ownerKey="" ownerTag="" status="deleted">
<l2RsAaaRadiusProviderGroup annotation=""
tDn="uni/userext/radiusext/radiusprovidergroup-radius-grp"/>
</l2NodeAuthPol>
</infraInfra>
</polUni>

Delete:
<polUni>
<infraInfra>
  <l2NodeAuthPol annotation="" descr="" dn="uni/infra/nodeauthpol-802-node-2"
failAuthEpg="tn-t2,ap-ap,epg-epg1" failAuthVlan="vlan-2078" name="802-node-2" nameAlias=""
 ownerKey="" ownerTag="" status="deleted">
<l2RsAaaRadiusProviderGroup annotation=""
tDn="uni/userext/radiusext/radiusprovidergroup-radius-grp" status="deleted"/>
</l2NodeAuthPol>
</infraInfra>
</polUni>
```



第 6 章

ポート セキュリティ

この章の内容は、次のとおりです。

ポート セキュリティと ACI について

ポート セキュリティ機能は、ポートごとに取得される MAC アドレスの数を制限することによって、不明な MAC アドレスでフラッディングしないように ACI ファブリックを保護します。ポート セキュリティ機能のサポートは、物理ポート、ポート チャネル、および仮想ポート チャネルで使用できます。

ポート セキュリティに関するガイドラインと制約事項

次のようなガイドラインと制約事項があります。

- ポート セキュリティは、ポートごとに使用できます。
- ポートセキュリティは、物理ポート、ポートチャネル、および仮想ポートチャネル（vPC）でサポートされています。
- スタティック MAC アドレスとダイナミック MAC アドレスがサポートされています。
- セキュアなポートからセキュアでないポートへと、セキュアでないポートからセキュアなポートへの MAC アドレスの移動がサポートされています。
- MAC アドレスの制限は、MAC アドレスにのみ適用され、MAC と IP によるアドレスには実行されません。

- ポート セキュリティは、ファブリック エクステンダ（FEX）ではサポートされていません。

ポート レベルでのポート セキュリティ

APICでは、ユーザがスイッチポートのポートセキュリティを設定できます。ポート上でMACが制限の最大設定値を超過すると、超過したMACアドレスからすべてのトラフィックが転送されます。次の属性がサポートされます。

- **ポート セキュリティのタイムアウト** ：現在サポートされているタイムアウト値値は、60 ~ 3600 秒の範囲でサポートされています。
- **違反行為**：違反行為は保護モードで使用できます。保護モードでは、MAC の取得が無効になるため、MAC アドレスは CAM テーブルに追加されません。Mac ラーニングが設定されているタイムアウト値の後に再度有効になります。
- **最大エンドポイント** ：現在のサポートされている最大のエンドポイント設定値は、0 ～ 12000 の範囲でサポートされています。最大エンドポイント値が 0 の場合、そのポートではポート セキュリティ ポリシーが無効になります。

APIC GUI を使用したポート セキュリティの設定

手順

ステップ 1 メニュー バーで、**[ファブリック]** > **[外部アクセス ポリシー]** をクリックし、**[ナビゲーション]** ペインで**[ポリシー]** > **[インターフェイス]** > **[ポート セキュリティ]** を展開します。

ステップ 2 **[ポート セキュリティ]** 右クリックして、**[ポート セキュリティ ポリシーの作成]** をクリックします。

ステップ 3 **[ポート セキュリティ ポリシーの作成]** ダイアログ ボックスで、次の操作を実行します。

a) [Name] フィールドにポリシーの名前を入力します。

b) **[ポート セキュリティのタイムアウト]** フィールドに、インターフェイスの MAC ラーニングを再度有効にする前に、タイムアウトの値を選択します。

c) **[最大エンドポイント]** フィールドに、インターフェイスで学習可能なエンドポイントの最大数の希望値を選択します。

d) **[違反アクション]** フィールドで、使用可能なオプションは **[保護]** です。[Submit] をクリックします。
ポート セキュリティ ポリシーが作成されます。

ステップ 4 （注） リーフ スイッチのインターフェイスを設定するときに、使用可能なポート セキュリティ ポリシーのリストからポートセキュリティポリシーを選択することができます。

[ナビゲーション] ペインで、**[ファブリック]** > **[インベントリ]** > **[トポロジ]** をクリックし、目的のリーフ スイッチに移動します。インターフェイスを設定する適切なポートを選択し、ポート セキュリティ ポリシー ドロップダウン リストから関連付けに必要なポート セキュリティ ポリシーを選択します。
これで、ポート上のポート セキュリティの設定を完了します。

REST API を使用して、ポート セキュリティの設定

手順

ポート セキュリティを設定します。

例：

```
<polUni>
 <infraInfra>

    <l2PortSecurityPol name="testL2PortSecurityPol" maximum="10" violation=”protect”
timeout=“300"/>

    <infraNodeP name="test">
        <infraLeafS name="test" type="range">
            <infraNodeBlk name="test" from_="101" to_="102"/>
        </infraLeafS>
        <infraRsAccPortP tDn="uni/infra/accportprof-test"/>
    </infraNodeP>

            <infraAccPortP name="test">
        <infraHPortS name="pselc" type="range">
           <infraPortBlk name="blk"
                                 fromCard="1" toCard="1" fromPort="20" toPort="22">
                 </infraPortBlk>
           <infraRsAccBaseGrp tDn="uni/infra/funcprof/accportgrp-testPortG" />
         </infraHPortS>
           </infraAccPortP>

           <infraFuncP>
           <infraAccPortGrp name="testPortG">
             <infraRsL2PortSecurityPol tnL2PortSecurityPolName="testL2PortSecurityPol"/>

                <infraRsAttEntP tDn="uni/infra/attentp-test" />
           </infraAccPortGrp>
           </infraFuncP>

    <infraAttEntityP name="test">
        <infraRsDomP tDn="uni/phys-mininet"/>
    </infraAttEntityP>
 </infraInfra>
</polUni>
```

CLI を使用したポート セキュリティの設定

手順

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `apic1# configure`	コンフィギュレーション モードに入ります。
ステップ 2	**leaf** *node-id* 例： `apic1(config)# leaf 101`	設定するリーフを指定します。
ステップ 3	**interface** *type-or-range* 例： `apic1(config-leaf)# interface eth 1/2-4`	設定するインターフェイスまたはインターフェイスの範囲を指定します。
ステップ 4	**[no] switchport port-security maximum** *number-of-addresses* 例： `apic1(config-leaf-if)# switchport port-security maximum 1`	インターフェイスのセキュア MAC アドレスの最大数を設定します。範囲は 0 ～ 12000 アドレスです。デフォルトは 1 アドレスです。
ステップ 5	**[no] switchport port-security violation protect** 例： `apic1(config-leaf-if)# switchport port-security violation protect`	セキュリティ違反が検出された場合に実行するアクションを設定します。**protect** アクションは、十分な数のセキュア MAC アドレスを削除して最大値を下回るまで、不明な送信元アドレスのパケットをドロップします。
ステップ 6	**[no] switchport port-security timeout** 例： `apic1(config-leaf-if)# switchport port-security timeout 300`	インターフェイスのタイムアウト値を設定します。範囲は 60 ～ 3600 です。デフォルトは 60 秒です。

例

次に、イーサネットインターフェイスでポートセキュリティを設定する方法を示します。

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface eth 1/2
apic1(config-leaf-if)# switchport port-security maximum 10
apic1(config-leaf-if)# switchport port-security violation protect
```

```
apic1(config-leaf-if)# switchport port-security timeout 300
```

次に、ポート チャネルでポート セキュリティを設定する例を示します。

```
apic1# configure
apic1(config)# leaf 101
apic1(config-leaf)# interface port-channel po2
apic1(config-leaf-if)# switchport port-security maximum 10
apic1(config-leaf-if)# switchport port-security violation protect
apic1(config-leaf-if)# switchport port-security timeout 300
```

次に、仮想ポートチャネル（VPC）でポートセキュリティを設定する例を示します。

```
apic1# configure
apic1(config)# vpc domain explicit 1 leaf 101 102
apic1(config-vpc)# exit
apic1(config)# template port-channel po4
apic1(config-if)# exit
apic1(config)# leaf 101-102
apic1(config-leaf)# interface eth 1/11-12
apic1(config-leaf-if)# channel-group po4 vpc
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
apic1(config)# vpc context leaf 101 102
apic1(config-vpc)# interface vpc po4
apic1(config-vpc-if)# switchport port-security maximum 10
apic1(config-vpc-if)# switchport port-security violation protect
apic1(config-leaf-if)# switchport port-security timeout 300
```

ポート セキュリティおよびラーニング動作

非 vPC ポートまたはポート チャネルでは、新しいエンドポイントに対して学習イベントが発生し、新しい学習が許可されているか確認する検証が行われます。対応するインターフェイスに設定されていない、または無効なポート セキュリティ ポリシーが存在する場合、エンドポイント ラーニング動作はサポートされているものから変更されません。ポリシーが有効になっており制限に到達している場合、現在のサポートされているアクションは次の通りです。

- エンドポイントを学習し、ドロップ アクションのハードウェアにインストールします。
- サイレントに学習を破棄します。

制限に到達していない場合、エンドポイントが学習され、この新しいエンドポイントが発生したため制限に達しているかどうか確認する検証が行われます。制限に到達しており、学習の無効化アクションが設定されている場合、インターフェイス上のハードウェアでラーニングが無効になります（物理インターフェイスまたはポートチャネルまたはvPC）。制限に到達しており、学習の無効化アクションが設定されていない場合、エンドポイントはドロップアクションでハードウェアにインストールされます。このようなエンドポイントは、他のエンドポイントのように通常期限切れです。

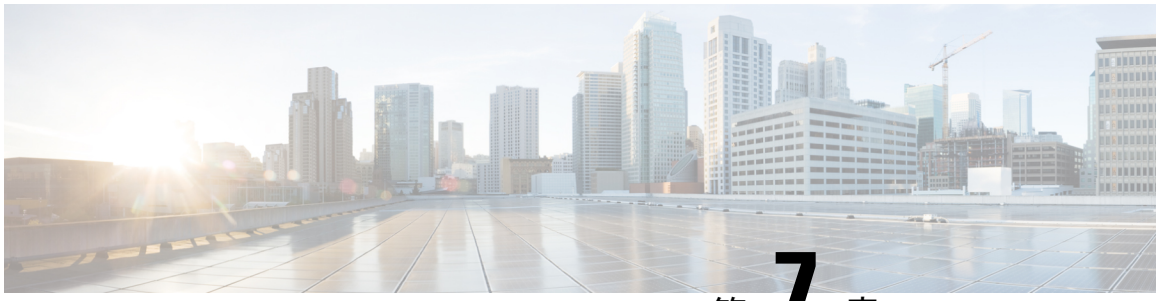
初めて制限に達したとき、ポート セキュリティ ポリシー オブジェクトの動作状態がそれを反映して更新されます。スタティック ルールは、ユーザーに警告ができるように、障害の発生と定義されます。制限に到達すると、Syslog も発生します。

vPCの場合、MAC 制限に到達するとピア リーフ スイッチにも通知されるため、ラーニングがピアで無効になる可能性があります。vPC ピアはいつでも再起動でき、vPC レッグが動作不能になるか再起動できるため、この状態はピアと調和して vPC ピアはこの状態に同期されません。同期しない場合は、1 個のレッグでラーニングが有効になり、他のレッグで無効になる状況が発生する可能性があります。

デフォルトでは、制限に到達してラーニングが無効になると、60 秒のデフォルト タイムアウト値の後、自動的に再度有効になります。

保護モード

保護モードはセキュリティ違反を発生している以上に増やさないようにします。MAC の制限がポートで設定されている最大値を超えると、超過したMACアドレスからすべてのトラフィックはドロップされ、さらにラーニングが無効になります。



第 7 章

ファーストホップ セキュリティ

この章の内容は、次のとおりです。

ファースト ホップ セキュリティについて

ファーストホップセキュリティ（FHS）機能では、レイヤ2リンク上でより優れたIPv4とIPv6のリンク セキュリティおよび管理が可能になります。サービス プロバイダ環境で、これらの機能は重複アドレス検出（DAD）とアドレス解像度（AR）などのアドレス割り当てや派生操作が、より緊密に制御可能です。

次のサポートされている FHS 機能はプロトコルをセキュアにして、ファブリック リーフ スイッチにセキュアなエンドポイント データベースを構築するのに役立ち、MIM 攻撃や IP の盗難などのセキュリティ盗難を軽減するために使用されます。

- ARP Inspection
- ND 検査
- DHCP 検査
- RA ガード
- IPv4 および Ipv6 ソース ガード
- トラスト制御

FHS 機能は、次のセキュリティ対策を提供します。

- **ロールの適用**：信頼できない主催者が、そのロールの有効範囲を超えるメッセージを送信することを防ぎます。
- **バインディングの適用**：アドレスの盗難を防止します。
- **DoS攻撃の軽減対策**：悪意あるエンドポイントを防ぎ、データベースが操作サービスを提供することを停止するポイントにエンドポイント データベースを成長させます。
- **プロキシ サービス**：アドレス解決の効率を高めるため一部のプロキシ サービスを提供します。

FHS機能は、テナントブリッジドメイン（BD）ごとに有効になっています。ブリッジドメインとして、単一または複数のリーフ スイッチで展開可能で、FHS 脅威の制御と軽減のメカニズムは単一のスイッチと複数のスイッチのシナリオにも対応できます。

ACI FHS の導入

ほとんどの FHS 機能はツーステップ傾向で設定されています。最初に機能の動作を説明するポリシーを定義し、次にこのポリシーを「ドメイン」に適用します（テナント ブリッジ ドメインまたはテナント エンドポイント グループになる）。異なる動作を定義する別のポリシーは、さまざまな交差ドメインに適用できます。特定のポリシーを使用する決定は、ポリシーを適用するもっとも明確なドメインで行われます。

ポリシーのオプションは、[Tenant_*name*]>[Networking]>[Protocol Policies]>[First Hop Security] タブの下にある Cisco APIC GUI から定義できます。

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。

- リリース 3.1 (1) より、仮想エンドポイント（AV のみ）で FHS はサポートされています。
- EPG が VXLAN カプセル化で展開されるとき、FHS 機能はサポートされていません。
- **[ダウン]** 状態の FHS バインディング表データベースでセキュリティ保護されたエンドポイント エントリは、タイムアウトから **18 時間** 後に消去されます。エントリが学習する前面パネル ポートがリンク ダウンする場合、エントリは **[ダウン]** 状態に移動します。この **18 時間** ウィンドウの中で、エンドポイントが別のロケーションに移動し別のポートで確認される場合、エンドポイントが他のポートから到達可能な限り移行され、エントリはグレースフルに **[ダウン]** 状態から **[REACHABLE/STALE]** に移行します。
- IP 発信元ガードが有効な時、IP 送信元アドレスとして Ipv6 リンク ローカル アドレスを使用して供給される Ipv6 トラフィックは、IP 送信元ガード施行を受けません（例：送信元 MAC の施行 <=> IP 調査機能によりセキュリティ保護された送信元 IP バインディング）。バインディング チェック障害に関係なく、デフォルトでこのトラフィックが許可されます。

- L3Out インターフェイスでは、FHS はサポートされていません。
- TOR に基づいて N9K-M12PQ では FHS はサポートされていません。
- ACI マルチサイトの FHS はサイトのローカル機能であるため、APIC クラスタからサイトでのみ有効にできます。また、ACI マルチサイトの FHS は、BD や EPG がサイト ローカルであり、サイト上でストレッチしない場合にのみ動作します。ストレッチ BD または EPG の FHS セキュリティを有効にすることはできません。
- レイヤ 2 専用ブリッジ ドメインでは、FHS はサポートされていません。
- FHS の有効化機能ではトラフィックが 50 秒間中断することがあります。これは、BD 内の EP がフラッシュされ、BD 内の EP ラーニングが 50 秒間無効になるためです。

APIC GUI を使用して FHS の設定

始める前に

- テナントとブリッジ ドメインが設定されています。

手順

ステップ 1 メニューバーで、**[テナント]** > **[Tenant_name]** をクリックします。**[ナビゲーション]** ペインで、**[ポリシー]** > **[プロトコル]** > **[最初のホップ セキュリティ]** をクリックします。**[最初のホップ セキュリティ]** を右クリックして **[機能ポリシーの作成]** を開き、次の操作の実行します。

a) **[名前]** フィールドにホップ セキュリティ セキュリティ ポリシーの名前を入力します。
b) **[IP 検査]**、**[送信元ガード]**、**[ルータ アドバタイズメント]** フィールドが有効になっていることを確認し、**[提出]** をクリックします。

ステップ 2 **[ナビゲーション]** ペインで、**[最初のホップ セキュリティ]** を展開し、**[制御ポリシーの信頼]** を右クリックして**[信頼制御ポリシーの作成]** を開いて次のアクションを実行します。

a) **[名前]** フィールドに信頼制御ポリシーの名前を入力します。
b) ポリシーで許可する機能を選択し、**[提出]** をクリックします。

ステップ 3 （オプション）EPG に信頼制御ポリシーを適用するには、[Navigation] ペインで、**[Application Profiles]** > **[ApplicationProfile_*name]*** > **[Application EPGs]** を展開し、**[Application EPG_*name*]** をクリックして、次の操作を行います。

a) **[作業]** ペインで、**[全般]** タブをクリックします。
b) **[FHS 信頼制御ポリシー]** の下矢印をクリックして、以前作成したポリシーを選択し、**[提出]** をクリックします。

ステップ 4 **[ナビゲーション]** ペインで、**[ブリッジ ドメイン]** > **[ブリッジ ドメイン名]** を展開して、**[アドバンスト/トラブルシューティング]** タブをクリックして、次のアクションを実行します。

a) **[ホップの最初のセキュリティ ポリシー]** フィールドで、作成したポリシーを選択し、**[提出]** をクリックします。これで FHS 設定を完了します。

NX-OS CLI を使用した FHS の設定

始める前に

- テナントとブリッジ ドメインが設定されています。

手順

ステップ 1 **configure**

コンフィギュレーション モードに入ります。

例：

```
apic1# configure
```

ステップ 2 FHS ポリシーを設定します。

例：

```
apic1(config)# tenant coke
apic1(config-tenant)# first-hop-security
apic1(config-tenant-fhs)# security-policy pol1
apic1(config-tenant-fhs-secpol)#
apic1(config-tenant-fhs-secpol)# ip-inspection-admin-status enabled-both
apic1(config-tenant-fhs-secpol)# source-guard-admin-status enabled-both
apic1(config-tenant-fhs-secpol)# router-advertisement-guard-admin-status enabled
apic1(config-tenant-fhs-secpol)# router-advertisement-guard
apic1(config-tenant-fhs-raguard)#
apic1(config-tenant-fhs-raguard)# managed-config-check
apic1(config-tenant-fhs-raguard)# managed-config-flag
apic1(config-tenant-fhs-raguard)# other-config-check
apic1(config-tenant-fhs-raguard)# other-config-flag
apic1(config-tenant-fhs-raguard)# maximum-router-preference low
apic1(config-tenant-fhs-raguard)# minimum-hop-limit 10
apic1(config-tenant-fhs-raguard)# maximum-hop-limit 100
apic1(config-tenant-fhs-raguard)# exit
apic1(config-tenant-fhs-secpol)# exit
apic1(config-tenant-fhs)# trust-control tcpol1
pic1(config-tenant-fhs-trustctrl)# arp
apic1(config-tenant-fhs-trustctrl)# dhcpv4-server
apic1(config-tenant-fhs-trustctrl)# dhcpv6-server
apic1(config-tenant-fhs-trustctrl)# ipv6-router
apic1(config-tenant-fhs-trustctrl)#  router-advertisement
apic1(config-tenant-fhs-trustctrl)# neighbor-discovery
apic1(config-tenant-fhs-trustctrl)# exit
apic1(config-tenant-fhs)# exit
apic1(config-tenant)# bridge-domain bd1
apic1(config-tenant-bd)# first-hop-security security-policy pol1
apic1(config-tenant-bd)# exit
```

```
apic1(config-tenant)# application ap1
apic1(config-tenant-app)# epg epg1
apic1(config-tenant-app-epg)# first-hop-security trust-control tcpol1
```

ステップ 3 FHS の設定例を示します。

例：

```
leaf4# show fhs bt all

Legend:
    TR       : trusted-access                       UNRES : unresolved              Age
   : Age since creation
    UNTR     : untrusted-access                    UNDTR : undetermined-trust      CRTNG
 : creating
    UNKNW    : unknown                              TENTV : tentative               INV
   : invalid
    NDP      : Neighbor Discovery Protocol         STA   : static-authenticated    REACH
 : reachable
    INCMP    : incomplete                           VERFY : verify                  INTF
  : Interface
    TimeLeft : Remaining time since last refresh    LM    : lla-mac-match           DHCP
  : dhcp-assigned

EPG-Mode:
    U : unknown    M : mac    V : vlan    I : ip

BD-VNID             BD-Vlan             BD-Name
15630220            3                   t0:bd200

------------------------------------------------------------------------------------------------------
| Origin | IP              | MAC               | INTF   | EPG(sclass)(mode) | Trust-lvl
 | State | Age      | TimeLeft |
------------------------------------------------------------------------------------------------------
| ARP    | 192.0.200.12    | D0:72:DC:A0:3D:4F | eth1/1 | epg300(49154)(V)  | LM,TR
 | STALE | 00:04:49 | 18:08:13 |
| ARP    | 172.29.205.232  | D0:72:DC:A0:3D:4F | eth1/1 | epg300(49154)(V)  | LM,TR
 | STALE | 00:03:55 | 18:08:21 |
| ARP    | 192.0.200.21    | D0:72:DC:A0:3D:4F | eth1/1 | epg300(49154)(V)  | LM,TR
 | REACH | 00:03:36 | 00:00:02 |
| LOCAL  | 192.0.200.1     | 00:22:BD:F8:19:FF | vlan3  | LOCAL(16387)(I)   | STA
 | REACH | 04:49:41 | N/A      |
| LOCAL  | fe80::200       | 00:22:BD:F8:19:FF | vlan3  | LOCAL(16387)(I)   | STA
 | REACH | 04:49:40 | N/A      |
| LOCAL  | 2001:0:0:200::1 | 00:22:BD:F8:19:FF | vlan3  | LOCAL(16387)(I)   | STA
 | REACH | 04:49:39 | N/A      |
------------------------------------------------------------------------------------------------------
```

ステップ 4 さまざまなタイプと理由の例とともに違反を表示します。

例：

```
leaf4# show fhs violations all

Violation-Type:
    POL  : policy      THR : address-theft-remote
    ROLE : role        TH  : address-theft
    INT  : internal

Violation-Reason:
    IP-MAC-TH   : ip-mac-theft                    OCFG_CHK  : ra-other-cfg-check-fail
   ANC-COL     : anchor-collision
    PRF-LVL-CHK : ra-rtr-pref-level-check-fail    INT-ERR   : internal-error
   TRUST-CHK   : trust-check-fail
    SRV-ROL-CHK : srv-role-check-fail             ST-EP-COL : static-ep-collision
```

```
   LCL-EP-COL  : local-ep-collision
    MAC-TH      : mac-theft                        EP-LIM    : ep-limit-reached
   MCFG-CHK    : ra-managed-cfg-check-fail
    HOP-LMT-CHK : ra-hoplimit-check-fail           MOV-COL   : competing-move-collision
   RTR-ROL-CHK : rtr-role-check-fail
    IP-TH       : ip-theft

EPG-Mode:
    U : unknown    M : mac    V : vlan    I : ip

BD-VNID             BD-Vlan             BD-Name
15630220            3                   t0:bd200
-------------------------------------------------------------------------------------------------------
| Type | Last-Reason | Proto | IP           | MAC               | Port    |
EPG(sclass)(mode) | Count |
-------------------------------------------------------------------------------------------------------
| THR  | IP-TH       | ARP   | 192.0.200.21 | D0:72:DC:A0:3D:4F | tunnel5 |
epg300(49154)(V)  | 21    |
-------------------------------------------------------------------------------------------------------
Table Count: 1
```

ステップ 5 FHS 設定の表示:

例:

```
swtb23-ifc1# show tenant t0 bridge-domain bd200 first-hop-security binding-table

Pod/Node  Type    Family  IP Address            MAC Address        Interface     Level
          State
--------  ------  ------  --------------------  -----------------  ------------
---------------  -----
1/102     local   ipv4    192.0.200.1           00:22:BD:F8:19:FF  vlan3         static-
          reach

authenticated    able
1/102     local   ipv6    fe80::200             00:22:BD:F8:19:FF  vlan3         static-
          reach

authenticated    able
1/102     local   ipv6    2001:0:0:200::1       00:22:BD:F8:19:FF  vlan3         static-
          reach

authenticated    able
1/101     arp     ipv4    192.0.200.23          D0:72:DC:A0:02:61  eth1/2
lla-mac-match    stale

,untrusted-
                                                                                  access
1/101     local   ipv4    192.0.200.1           00:22:BD:F8:19:FF  vlan3         static-
          reach

authenticated    able
1/101     nd      ipv6    fe80::d272:dcff:fea0  D0:72:DC:A0:02:61  eth1/2
lla-mac-match    reach
                          :261
,untrusted-      able
                                                                                  access
1/101     nd      ipv6    2001:0:0:200::20      D0:72:DC:A0:02:61  eth1/2
lla-mac-match    stale

,untrusted-
                                                                                  access
1/101     nd      ipv6    2001::200:d272:dcff:  D0:72:DC:A0:02:61  eth1/2
lla-mac-match    stale
```

```
                            fea0:261
,untrusted-
                                                                                access
1/101     local   ipv6    fe80::200             00:22:BD:F8:19:FF  vlan3         static-
          reach

authenticated    able
1/101     local   ipv6    2001:0:0:200::1       00:22:BD:F8:19:FF  vlan3         static-
          reach

authenticated    able
1/103     local   ipv4    192.0.200.1           00:22:BD:F8:19:FF  vlan4         static-
          reach

authenticated    able
1/103     local   ipv6    fe80::200             00:22:BD:F8:19:FF  vlan4         static-
          reach

authenticated    able
1/103     local   ipv6    2001:0:0:200::1       00:22:BD:F8:19:FF  vlan4         static-
          reach

authenticated    able
1/104     arp     ipv4    192.0.200.10          F8:72:EA:AD:C4:7C  eth1/1
lla-mac-match    stale

,trusted-access
1/104     arp     ipv4    172.29.207.222        D0:72:DC:A0:3D:4C  eth1/1
lla-mac-match    stale

,trusted-access
1/104     local   ipv4    192.0.200.1           00:22:BD:F8:19:FF  vlan4         static-
          reach

authenticated    able
1/104     nd      ipv6    fe80::fa72:eaff:fead  F8:72:EA:AD:C4:7C  eth1/1
lla-mac-match    stale
                           :c47c
,trusted-access
1/104     nd      ipv6    2001:0:0:200::10      F8:72:EA:AD:C4:7C  eth1/1
lla-mac-match    stale

,trusted-access
1/104     local   ipv6    fe80::200             00:22:BD:F8:19:FF  vlan4         static-
          reach

authenticated    able
1/104     local   ipv6    2001:0:0:200::1       00:22:BD:F8:19:FF  vlan4         static-
          reach

authenticated    able


Pod/Node  Type    IP Address            Creation TS                    Last Refresh TS
               Lease Period
--------  ------  --------------------  -----------------------------
-----------------------------  ------------
1/102     local   192.0.200.1           2017-07-20T04:22:38.000+00:00
2017-07-20T04:22:38.000+00:00
1/102     local   fe80::200             2017-07-20T04:22:56.000+00:00
2017-07-20T04:22:56.000+00:00
1/102     local   2001:0:0:200::1       2017-07-20T04:22:57.000+00:00
2017-07-20T04:22:57.000+00:00
1/101     arp     192.0.200.23          2017-07-27T10:55:20.000+00:00
```

```
2017-07-27T16:07:24.000+00:00
1/101     local   192.0.200.1           2017-07-27T10:48:09.000+00:00
2017-07-27T10:48:09.000+00:00
1/101     nd      fe80::d272:dcff:fea0  2017-07-27T10:52:16.000+00:00
2017-07-27T16:04:29.000+00:00
                  :261
1/101     nd      2001:0:0:200::20      2017-07-27T10:57:32.000+00:00
2017-07-27T16:07:24.000+00:00
1/101     nd      2001::200:d272:dcff:  2017-07-27T11:21:45.000+00:00
2017-07-27T16:07:24.000+00:00
                  fea0:261
1/101     local   fe80::200             2017-07-27T10:48:10.000+00:00
2017-07-27T10:48:10.000+00:00
1/101     local   2001:0:0:200::1       2017-07-27T10:48:11.000+00:00
2017-07-27T10:48:11.000+00:00
1/103     local   192.0.200.1           2017-07-26T22:03:56.000+00:00
2017-07-26T22:03:56.000+00:00
1/103     local   fe80::200             2017-07-26T22:03:57.000+00:00
2017-07-26T22:03:57.000+00:00
1/103     local   2001:0:0:200::1       2017-07-26T22:03:58.000+00:00
2017-07-26T22:03:58.000+00:00
1/104     arp     192.0.200.10          2017-07-27T11:21:13.000+00:00
2017-07-27T16:05:48.000+00:00
1/104     arp     172.29.207.222        2017-07-27T11:54:48.000+00:00
2017-07-27T16:06:38.000+00:00
1/104     local   192.0.200.1           2017-07-27T10:49:13.000+00:00
2017-07-27T10:49:13.000+00:00
1/104     nd      fe80::fa72:eaff:fead  2017-07-27T11:21:13.000+00:00
2017-07-27T16:06:43.000+00:00
                  :c47c
1/104     nd      2001:0:0:200::10      2017-07-27T11:21:13.000+00:00
2017-07-27T16:06:19.000+00:00
1/104     local   fe80::200             2017-07-27T10:49:14.000+00:00
2017-07-27T10:49:14.000+00:00
1/104     local   2001:0:0:200::1       2017-07-27T10:49:15.000+00:00
2017-07-27T10:49:15.000+00:00

swtb23-ifc1#

swtb23-ifc1# show tenant t0 bridge-domain bd200 first-hop-security statistics arp
Pod/Node         : 1/101
Request Received : 4
Request Switched : 2
Request Dropped  : 2
Reply Received   : 257
Reply Switched   : 257
Reply Dropped    : 0

Pod/Node         : 1/104
Request Received : 6
Request Switched : 6
Request Dropped  : 0
Reply Received   : 954
Reply Switched   : 954
Reply Dropped    : 0


swtb23-ifc1# show tenant t0 bridge-domain bd200 first-hop-security statistics dhcpv4
Pod/Node                   : 1/102
Discovery Received         : 5
Discovery Switched         : 5
Discovery Dropped          : 0
Offer Received             : 0
Offer Switched             : 0
```

```
Offer Dropped               : 0
Request Received            : 0
Request Switched            : 0
Request Dropped             : 0
Ack Received                : 0
Ack Switched                : 0
Ack Dropped                 : 0
Nack Received               : 0
Nack Switched               : 0
Nack Dropped                : 0
Decline Received            : 0
Decline Switched            : 0
Decline Dropped             : 0
Release Received            : 0
Release Switched            : 0
Release Dropped             : 0
Information Received        : 0
Information Switched        : 0
Information Dropped         : 0
Lease Query Received        : 0
Lease Query Switched        : 0
Lease Query Dropped         : 0
Lease Active Received       : 0
Lease Active Switched       : 0
Lease Active Dropped        : 0
Lease Unassignment Received : 0
Lease Unassignment Switched : 0
Lease Unassignment Dropped  : 0
Lease Unknown Received      : 0
Lease Unknown Switched      : 0
Lease Unknown Dropped       : 0

swtb23-ifc1# show tenant t0 bridge-domain bd200 first-hop-security statistics
neighbor-discovery
Pod/Node                          : 1/101
Neighbor Solicitation Received  : 125
Neighbor Solicitation Switched  : 121
Neighbor Solicitation Dropped   : 4
Neighbor Advertisement Received : 519
Neighbor Advertisement Switched : 519
Neighbor Advertisement Drop     : 0
Router Solicitation Received    : 4
Router Solicitation Switched    : 4
Router Solicitation Dropped     : 0
Router Adv Received             : 0
Router Adv Switched             : 0
Router Adv Dropped              : 0
Redirect Received               : 0
Redirect Switched               : 0
Redirect Dropped                : 0

Pod/Node                          : 1/104
Neighbor Solicitation Received  : 123
Neighbor Solicitation Switched  : 47
Neighbor Solicitation Dropped   : 76
Neighbor Advertisement Received : 252
Neighbor Advertisement Switched : 228
Neighbor Advertisement Drop     : 24
Router Solicitation Received    : 0
Router Solicitation Switched    : 0
Router Solicitation Dropped     : 0
Router Adv Received             : 53
Router Adv Switched             : 6
Router Adv Dropped              : 47
```

```
Redirect Received               : 0
Redirect Switched               : 0
Redirect Dropped                : 0
```

FHS スイッチ iBASH コマンド

手順

ステップ 1 BD の FHS 機能設定と、EPG の信頼コントロール ポリシー設定を表示する show コマンド：

例：

```
leaf4# show fhs features all

BD-VNID             BD-Vlan             BD-Name
15630220            4                   t0:bd200
    Feature Policy:
        Feature      Family    Protocol    Operational-State    Options
        ipinspect    IPV4      ARP         UP                   stalelifetime: 180s
        ipinspect    IPV4      DHCP        UP                   -
        ipinspect    IPV4      LOCAL       UP                   -
        ipinspect    IPV4      STATIC      UP                   -
        ipinspect    IPV6      ND          UP                   stalelifetime: 180s
        ipinspect    IPV6      DHCP        UP                   -
        ipinspect    IPV6      LOCAL       UP                   -
        ipinspect    IPV6      STATIC      UP                   -
        raguard      IPV6      -           UP                   ManagedCfgFlag: on
                                                                  OtherCfgFlag: on
                                                                  maxHopLimit: 15
                                                                  minHopLimit: 3
                                                                  routerPref: medium
    ------------------------------------------------------------------------------

    Trust Policy:
    Epg-id              Epg-type            Epg-name
    49154               Ckt-Vlan            epg300
        Trust-Attribute    Operational-State
        PROTO-ARP          UP
        PROTO-ND           UP
        DHCPV4-SERVER      UP
        DHCPV6-SERVER      UP
        ROUTER             UP
```

ステップ 2 FHS のセキュリティ保護されたエンドポイントのデータベースを表示する show コマンド：

例：

```
leaf1# show fhs bt
all       data      dhcpv4    local     static
arp       detailed  dhcpv6    nd        summary

leaf1# show fhs bt all

Legend:
    DHCP     : dhcp-assigned                      TR    : trusted-access          UNRES
 : unresolved
```

```
    Age      : Age since creation                  CRTNG : creating                TENTV
 : tentative
    VERFY    : verify                               UNDTR : undetermined-trust       INV
   : invalid
    NDP      : Neighbor Discovery Protocol         STA   : static-authenticated    REACH
 : reachable
    LM       : lla-mac-match                        UNKNW : unknown                 INTF
  : Interface
    TimeLeft : Remaining time since last refresh    INCMP : incomplete              UNTR
  : untrusted-access

EPG-Mode:
    U : unknown    M : mac    V : vlan    I : ip

BD-VNID             BD-Vlan             BD-Name
15630220            3                   t0:bd200

_____________________________________________________________________________________
| Origin | IP                       | MAC               | INTF   | EPG(sclass)(mode) |
Trust-lvl | State | Age      | TimeLeft |
_____________________________________________________________________________________
| ARP    | 192.0.200.23             | D0:72:DC:A0:02:61 | eth1/2 | epg200(32770)(V)  |
LM,UNTR   | STALE | 00:07:47 | 00:01:33 |
| LOCAL  | 192.0.200.1              | 00:22:BD:F8:19:FF | vlan3  | LOCAL(16387)(I)   |
STA       | REACH | 00:14:58 | N/A      |
| NDP    | fe80::d272:dcff:fea0:261 | D0:72:DC:A0:02:61 | eth1/2 | epg200(32770)(V)  |
LM,UNTR   | STALE | 00:10:51 | 00:00:47 |
| NDP    | 2001:0:0:200::20         | D0:72:DC:A0:02:61 | eth1/2 | epg200(32770)(V)  |
LM,UNTR   | STALE | 00:05:35 | 00:00:42 |
| LOCAL  | fe80::200                | 00:22:BD:F8:19:FF | vlan3  | LOCAL(16387)(I)   |
STA       | REACH | 00:14:58 | N/A      |
| LOCAL  | 2001:0:0:200::1          | 00:22:BD:F8:19:FF | vlan3  | LOCAL(16387)(I)   |
STA       | REACH | 00:14:57 | N/A      |
_____________________________________________________________________________________


leaf1# show fhs bt summary all

-------------------------------------------------------------------------------
                        FHS Binding Table Summary
-------------------------------------------------------------------------------
BD-Vlan: 3        BD-Name: t0:bd200
    Total number of ARP entries       : 1
    Total number of DHCPv4 entries    : 0
    Total number of ND entries        : 2
    Total number of DHCPv6 entries    : 0
    Total number of Data entries      : 0
    Total number of Static entries    : 0
    Total number of Local entries     : 3
    Total number of entries           : 6
-------------------------------------------------------------------------------
Total entries across all BDs matching given filters
    Total number of ARP entries       : 1
    Total number of DHCPv4 entries    : 0
    Total number of ND entries        : 2
    Total number of DHCPv6 entries    : 0
    Total number of Data entries      : 0
    Total number of Static entries    : 0
    Total number of Local entries     : 3
    Total number of entries           : 6
-------------------------------------------------------------------------------
```

ステップ 3 FHS エンドポイントの違反を表示する show コマンド：

例：

```
leaf1# show fhs violations all

Violation-Type:
    POL  : policy      THR : address-theft-remote
    ROLE : role        TH  : address-theft
    INT  : internal

Violation-Reason:
    IP-MAC-TH   : ip-mac-theft                    OCFG_CHK  : ra-other-cfg-check-fail
   ANC-COL     : anchor-collision
    PRF-LVL-CHK : ra-rtr-pref-level-check-fail    INT-ERR   : internal-error
   TRUST-CHK   : trust-check-fail
    SRV-ROL-CHK : srv-role-check-fail             ST-EP-COL : static-ep-collision
   LCL-EP-COL  : local-ep-collision
    MAC-TH      : mac-theft                       EP-LIM    : ep-limit-reached
   MCFG-CHK    : ra-managed-cfg-check-fail
    HOP-LMT-CHK : ra-hoplimit-check-fail          MOV-COL   : competing-move-collision
   RTR-ROL-CHK : rtr-role-check-fail
    IP-TH       : ip-theft

Trust-Level:
    TR   : trusted-access    UNTR : untrusted-access        UNDTR : undetermined-trust
    INV  : invalid           STA  : static-authenticated    LM    : lla-mac-match
    DHCP : dhcp-assigned

EPG-Mode:
    U : unknown    M : mac    V : vlan    I : ip

BD-VNID             BD-Vlan             BD-Name
15630220            4                   t0:bd200
------------------------------------------------------------------------------------------------
| Type | Last-Reason | Proto | IP                         | MAC               | Port   |
 EPG(sclass)(mode) | Trust-lvl | Count |
------------------------------------------------------------------------------------------------
| TH   | IP-TH       | ND    | 2001:0:0:200::20           | D0:72:DC:A0:3D:4F | eth1/1 |
 epg300(49154)(V)  | LM,UNTR   | 2     |
| POL  | HOP-LMT-CHK | RD    | fe80::fa72:eaff:fead:c47c | F8:72:EA:AD:C4:7C | eth1/1 |
 epg300(49154)(V)  | LM,TR     | 2     |
------------------------------------------------------------------------------------------------
Table Count: 2
```

ステップ 4 FHS コントロール パケット転送カウンタを表示する show コマンド：

例：

```
leaf1# show fhs counters
all     arp     dhcpv4  dhcpv6  nd
leaf4# show fhs counters all

BD-VNID             BD-Vlan             BD-Name
15630220            4                   t0:bd200
 ------------------------------------------------------------------------------------------
| Counter Type           |           Received |           Switched |             Dropped
 |
|------------------------------------------------------------------------------------------|
| Arp Request            |                  6 |                  6 |
  0 |
| Arp Reply              |                 94 |                 94 |
  0 |
------------------------------------------------------------------------------------------
| Dhcpv4 Ack             |                  0 |                  0 |
  0 |
| Dhcpv4 Decline         |                  0 |                  0 |
```

```
  0 |
| Dhcpv4 Discover        |                  0 |                  0 |
  0 |
| Dhcpv4 Inform          |                  0 |                  0 |
  0 |
| Dhcpv4 Leaseactive     |                  0 |                  0 |
  0 |
| Dhcpv4 Leasequery      |                  0 |                  0 |
  0 |
| Dhcpv4 Leaseunassigned |                  0 |                  0 |
  0 |
| Dhcpv4 Leaseunknown    |                  0 |                  0 |
  0 |
| Dhcpv4 Nack            |                  0 |                  0 |
  0 |
| Dhcpv4 Offer           |                  0 |                  0 |
  0 |
| Dhcpv4 Release         |                  0 |                  0 |
  0 |
| Dhcpv4 Request         |                  0 |                  0 |
  0 |
-------------------------------------------------------------------------------------------
| Dhcpv6 Advertise       |                  0 |                  0 |
  0 |
| Dhcpv6 Confirm         |                  0 |                  0 |
  0 |
| Dhcpv6 Decline         |                  0 |                  0 |
  0 |
| Dhcpv6 Informationreq  |                  0 |                  0 |
  0 |
| Dhcpv6 Rebind          |                  0 |                  0 |
  0 |
| Dhcpv6 Reconfigure     |                  0 |                  0 |
  0 |
| Dhcpv6 Relayforw       |                  0 |                  0 |
  0 |
| Dhcpv6 Relayreply      |                  0 |                  0 |
  0 |
| Dhcpv6 Release         |                  0 |                  0 |
  0 |
| Dhcpv6 Renew           |                  0 |                  0 |
  0 |
| Dhcpv6 Reply           |                  0 |                  0 |
  0 |
| Dhcpv6 Request         |                  0 |                  0 |
  0 |
| Dhcpv6 Solicit         |                  0 |                  0 |
  0 |
-------------------------------------------------------------------------------------------
| Nd Na                  |                 18 |                 18 |
  0 |
| Nd Ns                  |                 26 |                 22 |
  4 |
| Nd Ra                  |                 11 |                  6 |
  5 |
| Nd Redirect            |                  0 |                  0 |
  0 |
| Nd Rs                  |                  0 |                  0 |
  0 |
-------------------------------------------------------------------------------------------
```

ステップ 5 NxOS メモリから FHS のセキュリティ保護されたエンドポイントのデータベースを表示します。

例：

```
leaf1# vsh -c 'show system internal fhs bt'

Binding Table has 7 entries, 4 dynamic

Codes:
L - Local         S - Static        ND - Neighbor Discovery   ARP - Address Resolution
Protocol
DH4 - IPv4 DHCP   DH6 - IPv6 DHCP   PKT - Other Packet        API - API created


Preflevel flags (prlvl):
0001: MAC and LLA match     0002: Orig trunk            0004: Orig access
0008: Orig trusted trunk    0010: Orig trusted access   0020: DHCP assigned
0040: Cga authenticated     0080: Cert authenticated    0100: Statically assigned

EPG types:
V - Vlan Based EPG      M - MAC Based EPG       I - IP Based EPG

------------------------------------------------------------------------------------------------------
| Code | Network Layer Address                    | Link Layer Address | Interface
 | Vlan | Epg               | prlvl | Age         | State      | Time left       |
------------------------------------------------------------------------------------------------------
| ARP  | 172.29.207.222                           | d0:72:dc:a0:3d:4c  | Eth1/1
 |    4 | 0x40000c002 (V)   | 0011  | 29 s        | STALE      | 157 s           |
| L    | 192.0.200.1                              | 00:22:bd:f8:19:ff  | Vlan4
 |    4 | 0x400004003 (I)   | 0100  | 55 mn       | REACHABLE  |                 |
| ARP  | 192.0.200.10                             | f8:72:ea:ad:c4:7c  | Eth1/1
 |    4 | 0x40000c002 (V)   | 0011  | 156 s       | STALE      | 30 s            |
| L    | 2001:0:0:200::1                          | 00:22:bd:f8:19:ff  | Vlan4
 |    4 | 0x400004003 (I)   | 0100  | 55 mn       | REACHABLE  |                 |
| ND   | 2001:0:0:200::10                         | f8:72:ea:ad:c4:7c  | Eth1/1
 |    4 | 0x40000c002 (V)   | 0011  | 143 s       | STALE      | 47 s            |
| L    | fe80::200                                | 00:22:bd:f8:19:ff  | Vlan4
 |    4 | 0x400004003 (I)   | 0100  | 55 mn       | REACHABLE  |                 |
| ND   | fe80::fa72:eaff:fead:c47c                | f8:72:ea:ad:c4:7c  | Eth1/1
 |    4 | 0x40000c002 (V)   | 0011  | 176 s       | STALE      | 11 s            |
------------------------------------------------------------------------------------------------------
```

ステップ 6 NX-OS FHS プロセス内蔵メモリから FHS 機能の設定を表示します。

例：

```
leaf4# vsh -c 'show system internal fhs pol'

Target           Type  Policy           Feature       Target-Range Sub-Feature
epg 0x40000c002  EPG   epg 0x40000c002  Trustctrl      vlan 4      Device-Roles:
DHCPv4-Server, DHCPv6-Server, Router
                                                                   Protocols: ARP ND
vlan 4          VLAN  vlan 4          IP inspect     vlan all    Protocols: ARP, DHCPv4,
 ND, DHCPv6,
vlan 4           VLAN  vlan 4           RA guard       vlan all    Min-HL:3, Max-HL:15,
 M-Config-flag:Enable,On
                                                                  O-Config-flag:Enable,On,
 Router-Pref:medium
```

ステップ 7 NX-OS 共有データベースから FHS のセキュリティ保護されたエンドポイントのデータベースを表示します。

例：

```
leaf1# vsh -c 'show system internal fhs sdb bt'

Preflevel flags (preflvl):
0001: MAC and LLA match     0002: Orig trunk            0004: Orig access
0008: Orig trusted trunk    0010: Orig trusted access   0020: DHCP assigned
0040: Cga authenticated     0080: Cert authenticated    0100: Statically assigned

Origin      Zone ID     L3 Address                                MAC Address
VLAN ID  EPG ID              If-name         Preflvl  State
----------  ----------  ------------------------------------  ------------------
-------  ------------------  --------------  -------  ---------
ARP         0x4         172.29.207.222                            d0:72:dc:a0:3d:4c   4
       0x40000c002         Eth1/1          0011     STALE
L           0x4         192.0.200.1                               00:22:bd:f8:19:ff   4
       0x400004003         Vlan4           0100     REACHABLE
ARP         0x4         192.0.200.10                              f8:72:ea:ad:c4:7c   4
       0x40000c002         Eth1/1          0011     REACHABLE
L           0x4         2001:0:0:200::1                           00:22:bd:f8:19:ff   4
       0x400004003         Vlan4           0100     REACHABLE
ND          0x4         2001:0:0:200::10                          f8:72:ea:ad:c4:7c   4
       0x40000c002         Eth1/1          0011     STALE
L           0x80000004  fe80::200                                 00:22:bd:f8:19:ff   4
       0x400004003         Vlan4           0100     REACHABLE
ND          0x80000004  fe80::fa72:eaff:fead:c47c                 f8:72:ea:ad:c4:7c   4
       0x40000c002         Eth1/1          0011     STALE
```

ステップ 8 NxOS 共有データベースから FHS 機能の設定を表示します。

例：

```
leaf1# vsh -c 'show system internal fhs sdb pol'
Policies:

IP inspect      Vlan 4                    Protocols:ARP DHCPv4 ND DHCPv6
RA guard        Vlan 4                    Min-HL:3 Max-HL:15 M-Config-Flag:enable,on
O-Config-Flag:enable,on Router-Pref:medium
Trustctrl       Epg 0x40000c002           Vlan:4
                                        Device-Roles:DHCPv4-Server DHCPv6-Server Router

                                          Protocols:ARP ND
```

ステップ 9 セキュリティ保護されたデータベース エンドポイント エントリを消去する show コマンド：

例：

```
leaf1# vsh -c 'clear system internal fhs bt ipv4 172.29.207.222'
```

REST API を使用して apic 内で FHS の設定

始める前に

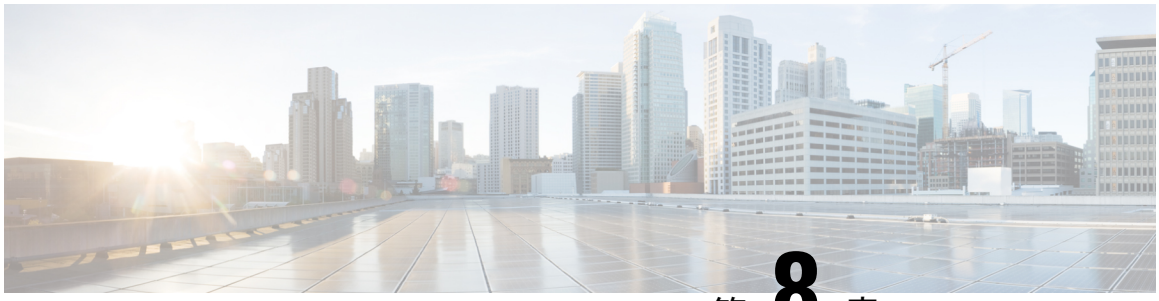
- テナントおよびブリッジ ドメインは設定しておく必要があります。

手順

FHS と信頼制御ポリシーを設定します。

例：

```
<polUni>
    <fvTenant name="Coke">
        <fhsBDPol name="bdpol5" ipInspectAdminSt="enabled-ipv6"
srcGuardAdminSt="enabled-both" raGuardAdminSt="enabled" status="">
            <fhsRaGuardPol name="raguard5" managedConfigCheck="true"
managedConfigFlag="true" otherConfigCheck="true" otherConfigFlag="true"
maxRouterPref="medium" minHopLimit="3" maxHopLimit="15" status=""/>
        </fhsBDPol>
        <fvBD name="bd3">
            <fvRsBDToFhs tnFhsBDPolName="bdpol5" status=""/>
        </fvBD>
    </fvTenant>
</polUni>

<polUni>
<fvTenant name="Coke">
       <fhsTrustCtrlPol name="trustctrl5" hasDhcpv4Server="true" hasDhcpv6Server="true"
 hasIpv6Router="true" trustRa="true" trustArp="true" trustNd="true" />
        <fvAp name="wwwCokecom3">
           <fvAEPg name="test966">
           <fvRsTrustCtrl tnFhsTrustCtrlPolName="trustctrl5" status=""/>
        </fvAEPg>
        </fvAp>
</fvTenant>
</polUni>
```



第 8 章

プロトコル認証

この章は、次の項で構成されています。

COOP

概要

マッピング情報（ロケーションおよび ID）をスパイン プロキシに伝達するために、Council of Oracles Protocol（COOP）を使用します。リーフ スイッチは、Zero Message Queue（ZMQ）を使用して、エンドポイント アドレス情報をスパイン スイッチ「Oracle」に転送します。スパイン ノードで実行している COOP によって、すべてのスパイン ノードが一貫性のあるエンドポイント アドレスとロケーション情報のコピーを維持することができ、さらに、ロケーション マッピング データベースに対するエンドポイント ID の分散ハッシュ テーブル（DHT）レポジトリを維持することができます。

COOP データ パス通信は、セキュアな接続を介した転送を優先します。COOP は悪意のあるトラフィック インジェクションから COOP メッセージを保護するために MD5 オプションの活用が強化されます。APIC コントローラおよびスイッチは、COOP プロトコル認証をサポートします。

COOP プロトコルは 2 つの ZMQ 認証モードをサポートするために強化されています：ストリクトおよび互換性。

- ストリクト モード：COOP では MD5 認証済みの ZMQ 接続のみを許可します。
- 互換性モード；COOP ではメッセージの転送に MD5 認証接続と非認証 ZMQ 接続の両方を許可します。

Cisco APIC で COOP を使用する

Cisco Application Centric Infrastructure（ACI）ファブリック上で COOP ゼロ メッセージ キュー（ZMQ）認証サポートを行うため、Application Policy Infrastructure Controller（APIC）では MD5 パスワードおよび COOP セキュア モードをサポートします。

COOP ZMQ 認証タイプの設定：新しい管理対象オブジェクトの `coop: AuthP` は、データ管理エンジン（DME）データベース（DME）/COOP に追加されます。属性タイプのデフォルト値は「互換性」ですが、ユーザーには「厳密」タイプ設定を行うオプションがあります。

COOP ZMQ 認証 MD5 パスワード：APIC では管理対象オブジェクト（`fabric:SecurityToken`）提供し、MD5 パスワードに使用する属性が含まれます。「トークン」と呼ばれるこの管理対象オブジェクト内の属性は、1 時間ごとに変更される文字列です。COOP は、ZMQ 認証用のパスワードを更新するため DME から通知を取得します。属性トークン値は表示されません。

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。

- ACI ファブリックのアップグレード中は、すべてのスイッチがアップグレードされるまで、COOP 厳格モードが許可されません。この保護は、早期に厳格なモードを有効にすることでトリガされる可能性がある、予期しない COOP 接続の拒否を防ぎます。

APIC GUI を使用した COOP 認証の設定

手順

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

ステップ 2 **[ナビゲーション]** ペインで **[COOP グループ]** をクリックします。

ステップ 3 **[作業]** ペインの **[タイプ]** フィールドにある **[ポリシー プロパティ]** 領域で、**[互換性のあるタイプ]** および **[ストリクト タイプ]** オプションから希望のタイプを選択します。

ステップ 4 [Submit] をクリックします。
これにより、COOP 認証ポリシー設定を完了します。

Cisco NX OS スタイル CLI を使用した COOP 認証の設定

手順

ストリクト モード オプションを使用して、COOP 認証ポリシーを設定します。

例：

```
apic1# configure
apic1(config)# coop-fabric
apic1(config-coop-fabric)# authentication type ?
compatible  Compatible type
strict      Strict type
apic101-apic1(config-coop-fabric)# authentication type strict
```

REST API を使用した COOP 認証の設定

手順

COOP 認証ポリシーを設定します。

例では、ストリクト モードが選択されます。

例：

```
https://172.23.53.xx/api/node/mo/uni/fabric/pol-default.xml

<coopPol type="strict">
</coopPol>
```

EIGRP

概要

EIGRP は、リンクステート プロトコルの機能にディスタンス ベクトル プロトコルの利点を組み合わせたプロトコルです。EIGRP は、定期的に Hello メッセージを送信してネイバーを探索します。EIGRP は、新規ネイバーを検出すると、すべてのローカル EIGRP ルートおよびルート メトリックに対する 1 回限りの更新を送信します。受信側の EIGRP ルータは、受信したメトリックと、その新規ネイバーにローカルで割り当てられたリンクのコストに基づいて、ルート ディスタンスを計算します。この最初の全面的なルート テーブルの更新後は、ルート変更の影響を受けるネイバーにのみ、差分更新が EIGRP により送信されます。この処理により、コンバージェンスにかかる時間が短縮され、EIGRP が使用する帯域幅が最小限になります。

Cisco APIC では、EIGRP 認証でルートマップのキーチェーンのインフラストラクチャが MD5 認証に使用されます。2 つの EIGRP ピア間で認証を設定するには 2 つのパラメータが必要になります。パラメータは次のとおりです。

- モード
- Keychain

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。

- MD5 認証のみサポートされます。キーチェーンは、RPM で設定されているキーチェーン名です。
- 2 つの EIGRP ピア間で認証の不一致がある場合は、ネイバーシップのフラッピングが発生します。フラッピングの理由は **show eigrp internal event-history syslog** で確認できます。

APIC GUI を使用した EIGRP 認証の設定

手順

ステップ 1 メニュー バーで、**[Tenant]*/tenant-name]***を選択します。

ステップ 2 [Navigation] ペインで、**[Policies]** > **[Protocol]** > **[EIGRP]** を展開します。

ステップ 3 [EIGRP] を展開し、[EIGRP KeyChains] を右クリックして [Create Keychain Policy] を開き、次の操作を行います。

a) [Name] フィールドにポリシーの名前を入力します。
b) [KeyID] フィールドに、キー ID 番号を入力します。
c) [Preshared key] フィールドに、事前共有キーの情報を入力します。
d) オプション。[Start Time] フィールドと [End Time] フィールドに、時間を入力します。

ステップ 4 [Navigation] ペインで、[EIGRP Interface] を右クリックし、次の操作を行います。

a) [Authentication] フィールドで、ボックスをクリックして有効にします。
b) [Key Chain Policy] フィールドで、ドロップダウン リストから作成したポリシーを選択し、[Submit] をクリックします。

NX-OS CLI を使用した EIGRP 認証の設定

手順

ステップ 1 テナントで、キーチェーン ポリシーとキーポリシーを設定します。

例：

```
tenant T1
keychain-policy KeyChainPol
key-policy 2
```

ステップ 2 オプション。開始時刻を設定します。

例：

```
startime 2018-11-01T08:39:27.000+00:00
exit
```

ステップ 3 APIC からリーフ設定を開始します。インターフェイスでの認証を有効にし、キーチェーン ポリシーを設定します。

例：

```
IFC1(config-leaf)# show run
# Command: show running-config leaf 104
# Time: Thu Nov 8 12:05:45 2018
leaf 104
interface ethernet 1/2.45
vrf member tenant T1 vrf V1 l3out L3Out
ip router eigrp authentication keychain-policy KeyChainPol
ip router eigrp authentication enable
!
ipv6 router eigrp authentication keychain-policy KeyChainPol
ipv6 router eigrp authentication enable
exit
```

ステップ 4 EIGRP の設定を確認するには、次の手順を実行します。

例：

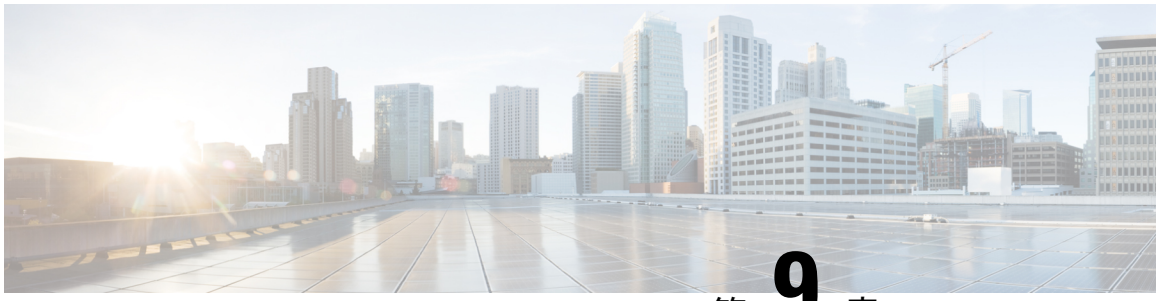
```
fav-blr4-ls-leaf4# show ip eigrp interfaces eth1/2.17
EIGRP interfaces for process 1 VRF T1:V1
Xmit Queue Mean Pacing Time Multicast Pending
Interface Peers Un/Reliable SRTT Un/Reliable Flow Timer Routes
eth1/2.17 0 0/0 0 0/0 50 0
Hello interval is 5 sec
Holdtime interval is 15 sec
Next xmit serial: 0
Un/reliable mcasts: 0/3 Un/reliable ucasts: 6/4
Mcast exceptions: 0 CR packets: 0 ACKs suppressed: 1
Retransmissions sent: 0 Out-of-sequence rcvd: 0
Classic/wide metric peers: 0/0
Authentication mode is md5, key-chain is T1:KeyChainPol
ifav-blr4-ls-leaf4#
```

ステップ 5 スイッチでトラブルシューティングを行う場合は、次の CLI を使用できます。EIGRP 認証は、IPv4 と IPv6 の両方のアドレス ファミリでサポートされています。

例：

```
(none)# show ip eigrp interface vrf all
EIGRP interfaces for process 100 VRF pepsi
Xmit Queue Mean Pacing Time Multicast Pending
Interface Peers Un/Reliable SRTT Un/Reliable Flow Timer Routes
eth1/1 1 0/0 207 0/0 828 0
Hello interval is 10 sec
Holdtime interval is 15 sec
Next xmit serial: 0
Un/reliable mcasts: 0/7 Un/reliable ucasts: 21/18
Mcast exceptions: 0 CR packets: 0 ACKs suppressed: 0
Retransmissions sent: 4 Out-of-sequence rcvd: 2
Classic/wide metric peers: 0/1
Authentication mode is md5, key-chain is eigrp-auth

(none)# show ipv6 eigrp interface vrf pepsi
IPv6-EIGRP interfaces for process 100 VRF pepsi
Xmit Queue Mean Pacing Time Multicast Pending
```

```
Interface Peers Un/Reliable SRTT Un/Reliable Flow Timer Routes
eth1/1 0 0/0 0 0/0 0 0
Hello interval is 10 sec
Holdtime interval is 15 sec
Next xmit serial: 0
Un/reliable mcasts: 0/0 Un/reliable ucasts: 0/0
Mcast exceptions: 0 CR packets: 0 ACKs suppressed: 0
Retransmissions sent: 0 Out-of-sequence rcvd: 0
Classic/wide metric peers: 0/0
Authentication mode is md5, key-chain is eigrp-auth
```



第 9 章

コントロール プレーンのトラフィック

CoPP の概要

コントロール プレーン ポリシング（CoPP）はコントロール プレーンを保護し、それをデータ プレーンから分離することによって、ネットワークの安定性、到達可能性、およびパケット配信を保証します。

この機能により、コントロールプロセッサに到達可能な各プロトコルに対して、パラメータの仕様でポリサーを使用したレート制限が可能になります。ポリシングは、ルータまたはレイヤ 3 スイッチの IP アドレスのいずれかを宛先とするすべてのトラフィックに適用されます。ネットワーク デバイスへの一般的な攻撃ベクトルは、過剰なトラフィックがデバイスインターフェイスに転送されるサービス妨害（DoS）攻撃です。

Cisco ACI リーフ/スパイン NX-OSは、DoS 攻撃がパフォーマンスに影響しないようにするために CoPP を提供します。このような攻撃は誤って、または悪意を持って実行される場合があり、通常は、ACI リーフ/スパイン CPU のスーパーバイザ モジュールまたは CPU 自体に宛てられた大量のトラフィックが含まれます。

ACI リーフ/スパイン スイッチのスーパーバイザ モジュールは、管理対象のトラフィックを次の 2 つの機能コンポーネント（プレーン）に分類します。

- **データ プレーン**：すべてのデータ トラフィックを処理します。NX-OS デバイスの基本的な機能は、インターフェイス間でパケットを転送することです。スイッチ自身に向けられたものでないパケットは、中継パケットと呼ばれます。データ プレーンで処理されるのはこれらのパケットです。
- **コントロール プレーン**：ルーティング プロトコルのすべての制御トラフィックを処理します。ボーダー ゲートウェイ プロトコル（BGP）や Open Shortest Path First（OSPF）プロトコルなどのルーティング プロトコルは、デバイス間で制御パケットを送信します。これらのパケットはルータのアドレスを宛先とし、コントロール プレーン パケットと呼ばれます。

ACI リーフ/スパイン スーパーバイザ モジュールはコントロール プレーンを所有しており、ネットワークの動作に重要です。スーパーバイザモジュールの動作が途絶したり、スーパーバイザモジュールが攻撃されたりすると、重大なネットワークの停止につながります。たとえばスーパーバイザに過剰なトラフィックが加わると、スーパーバイザ モジュールが過負荷になり、Cisco ACI ファブリック全体のパフォーマンスが低下する可能性があります。またたとえば、ACI リーフ/スパイン スーパーバイザ モジュールに対する DoS 攻撃は、コントロール プレーンに対して非常に高速に IP トラフィック ストリームを生成することがあります。これにより、コントロールプレーンは、これらのパケットを処理するために大量の時間を費やしてしまい、本来のトラフィックを処理できなくなります。

次に、DoS 攻撃の例を示します。

- インターネット制御メッセージ プロトコル（ICMP）エコー要求
- IP フラグメント
- TCP SYN フラッディング

これらの攻撃によりデバイスのパフォーマンスが影響を受け、次のようなマイナスの結果をもたらします。

- サービス品質の低下（音声、ビデオ、または重要なアプリケーション トラフィックの低下など）
- ルート プロセッサまたはスイッチ プロセッサの高い CPU 使用率
- ルーティング プロトコルのアップデートまたはキープアライブの消失によるルート フラップ
- メモリやバッファなどのプロセッサ リソースの枯渇
- 着信パケットの無差別のドロップ

（注） ACI リーフ/スパインは、デフォルト設定で CoPP によって保護されているデフォルトです。この機能では、顧客のニーズに基づいてノードのグループにパラメータを調整できます。

コントロール プレーン保護

コントロール プレーンを保護するために、ACI リーフ/スパインで実行している Cisco NX-OS でコントロールプレーンに向かうさまざまなパケットを異なるクラスに分離します。クラスの識別が終わると、Cisco NX-OS デバイスはパケットをポリシングします。これにより、スーパーバイザ モジュールに過剰な負担がかからないようになります。

コントロール プレーンのパケット タイプ:

コントロール プレーンには、次のような異なるタイプのパケットが到達します。

- **受信パケット**：ルータの宛先アドレスを持つパケット。宛先アドレスには、レイヤ2アドレス（ルータ MAC アドレスなど）やレイヤ 3 アドレス（ルータ インターフェイスの IP

アドレスなど）があります。これらのパケットには、ルータアップデートとキープアライブ メッセージも含まれます。ルータが使用するマルチキャスト アドレス宛てに送信されるマルチキャスト パケットも、このカテゴリに入ります。

- **例外パケット**：スーパーバイザモジュールによる特殊な処理を必要とするパケット。たとえば、宛先アドレスが Forwarding Information Base（FIB; 転送情報ベース）に存在せず、結果としてミスとなった場合は、スーパーバイザモジュールが送信側に到達不能パケットを返します。他には、IP オプションがセットされたパケットもあります。
- **リダイレクト パケット**：スーパーバイザ モジュールにリダイレクトされるパケット。ダイナミック ホスト コンフィギュレーション プロトコル（DHCP）スヌーピングやダイナミック アドレス解決プロトコル（ARP）インスペクションなどの機能は、パケットをスーパーバイザ モジュールにリダイレクトします。
- **収集パケット**：宛先 IP アドレスのレイヤ 2 MAC アドレスが FIB に存在していない場合は、スーパーバイザ モジュールがパケットを受信し、ARP 要求をそのホストに送信します。

これらのさまざまなパケットはすべて、コントロールプレーンへの悪意ある攻撃に利用され、Cisco ACI ファブリックに過剰な負荷をかける可能性があります。CoPP は、これらのパケットを異なるクラスに分類し、これらのパケットを ACI リーフ/スパイン スーパーバイザが受信する速度を個別に制御するメカニズムを提供します。

CoPP の分類：

効果的に保護するために、ACI リーフ/スパイン NX-OS はスーパーバイザ モジュールに到達するパケットを分類して、パケット タイプに基づいた異なるレート制御ポリシーを適用できるようにします。たとえば、Hello メッセージなどのプロトコル パケットには厳格さを緩め、IP オプションがセットされているためにスーパーバイザ モジュールに送信されるパケットには厳格さを強めることが考えられます。

レート制御メカニズム：

パケットの分類が終わると、ACI リーフ/スパイン NX-OS にはスーパーバイザ モジュールに到達するパケットのレートを制御するメカニズムがあります。

ポリシングには、次のパラメータを設定できます。

- **認定情報速度（CIR）**：リンク レートをビット レートまたはパーセンテージで指定した、目的の帯域幅。
- **認定バースト（BC）**：指定した時間枠内に CIR を超過する可能性があるが、スケジューリングには影響を与えないトラフィック バーストのサイズ。

デフォルトのポリシング ポリシー：

Cisco ACI リーフ/スパイン ブートアップがある場合は、プラットフォーム設定事前定義された CoPP パラメータのさまざまなプロトコルが Cisco によって実行テストに基づいています。

CoPP の注意事項と制約事項

CoPP に関する注意事項と制約事項は次のとおりです。

- 最初にデフォルト CoPP ポリシーを使用し、後で、データセンターおよびアプリケーションの要件に基づいて CoPP ポリシーを変更することをお勧めします。
- CoPP のカスタマイズは継続的なプロセスです。CoPP を設定するときには、特定の環境で使用されるプロトコルや機能に加えて、サーバ環境に必要なスーパーバイザ機能を考慮する必要があります。これらのプロトコルや機能が変更されたら、CoPP を変更する必要があります。
- CoPP を継続的にモニタすることを推奨します。ドロップが発生した場合は、CoPP がトラフィックを誤ってドロップしたのか、または誤動作や攻撃に応答してドロップしたのかを判定してください。いずれの場合も、状況を分析し、CoPP ポリシーを変更する必要を評価します。
- CoPP ポリシーによって、ルーティング プロトコルなどのクリティカルなトラフィック、またはデバイスへのインタラクティブなアクセスがフィルタリングされないように注意してください。このトラフィックをフィルタリングすると、Cisco ACI リーフ/スパインへのリモート アクセスが禁止され、コンソール接続が必要となる場合があります。
- APIC UI を使用して、CoPP パラメータを調整することができます。
- プロトコルごとの各インターフェイスはリーフ スイッチでのみサポートされています。
- プロトコルごとの各インターフェイスで FEX ポートはサポートされていません。
- プロトコルごとの各インターフェイスでサポートされているプロトコルは、ARP、ICMP、CDP、LLDP、LACP、BGP、STP、BFD、および OSPF です。
- プロトコルごとの各インターフェイスの最大の TCAM エントリは 256 です。しきい値を超過すると、障害が発生します。

APIC GUI を使用した CoPP の設定

手順

ステップ 1 メニュー バーで、**[ファブリック]** > **[外部アクセス ポリシー]** をクリックします。

ステップ 2 **[ナビゲーション]** ペインで、**[ポリシー]** > **[スイッチ]** > **[CoPP リーフ]** を展開して、**[リーフ レベルで適用される CoPP のプロファイルの作成]** ダイアログ ボックスを右クリックし、**[リーフ レベルで適用される CoPP のプロファイルの作成]** ダイアログ ボックスの次のアクションを実行します。

a) **[名前]** フィールドでポリシー名を追加します。
b) **[プロファイルのタイプ]** フィールドで、プロファイル タイプを選択します。

（注） 各プロトコルを個別に設定する場合、**[CoPPにカスタム値がある]** を選択します。プロファイル タイプを選択しない場合、デフォルト値が適用されます。

c) **[送信]** をクリックしてポリシーを作成します。

ステップ 3 **[ナビゲーション]** ペインで、**[スイッチ]** > **[リーフ スイッチ]** > **[ポリシー グループ]** を展開し、**[アクセス スイッチ ポリシー グループの作成]** ダイアログ ボックスを右クリックして、**[アクセス スイッチ ポリシー グループの作成]** ダイアログ ボックスの次のアクションを実行します。

a) **[名前]** フィールドでポリシー名を追加します。

b) **[COPP リーフ ポリシー]** フィールドで、以前に作成されたポリシーを選択します。

c) [Submit] をクリックします。

ステップ 4 **[ナビゲーション]** ペインで、**[スイッチ]**>**[リーフ スイッチ]**>**[プロファイル]** を展開して、**[リーフ プロファイルの作成]** ダイアログ ボックスを右クリックして、**[リーフ プロファイルの作成]** ダイアログ ボックスの次のアクションを実行します。

a) **[名前]** フィールドで、プロファイル名を追加します。

b) **[リーフ セレクタ]** 表を展開して、**[名前]** と **[ブロック]** フィールドにリーフ情報を追加して、以前作成した **[ポリシー グループ]** を選択します。

c) **[次へ]** および **[終了]** をクリックして、CoPP 設定を実行します。

Cisco NX-OS CLI を使用した CoPP の設定

手順

ステップ 1 CoPP リーフ プロファイルを設定します。

例：

```
# configure copp Leaf Profile
apic1(config)# policy-map type control-plane-leaf leafProfile
apic1(config-pmap-copp-leaf)# profile-type custom
apic1(config-pmap-copp-leaf)# set arpRate 786
# create a policy group to be applied on leaves
apic1(config)# template leaf-policy-group coppForLeaves
apic1(config-leaf-policy-group)# copp-aggr leafProfile
apic1(config-leaf-policy-group)# exit
# apply the leaves policy group on leaves
apic1(config)# leaf-profile applyCopp
apic1(config-leaf-profile)# leaf-group applyCopp
apic1(config-leaf-group)# leaf 101-102
apic1(config-leaf-group)# leaf-policy-group coppForLeaves
```

ステップ 2 CoPP スパイン プロファイルを設定します。

例：

```
# configure copp Spine Profile
apic1(config)# policy-map type control-plane-spine spineProfile
apic1(config-pmap-copp-spine)# profile-type custom
apic1(config-pmap-copp-spine)# set arpRate 786
```

```
# create a policy group to be applied on spines
apic1(config)# template leaf-policy-group coppForSpines
apic1(config-spine-policy-group)# copp-aggr spineProfile
apic1(config-spine-policy-group)# exit
# apply the spine policy group on spines
apic1(config)# spine-profile applyCopp
apic1(config-spine-profile)# spine-group applyCopp
apic1(config-spine-group)# spine 201-202
apic1(config-spine-group)# spine-policy-group coppForSpines
```

REST API を使用した CoPP の設定

手順

ステップ 1 CoPP リーフ プロファイルを設定します。

例：

```
<!-- api/node/mo/uni/.xml -->
<infraInfra>
  <coppLeafProfile type="custom" name="mycustom">                <!-- define copp leaf
 profile -->
    <coppLeafGen1CustomValues bgpBurst="150" bgpRate="300"/>
  </coppLeafProfile>
  <infraNodeP name="leafCopp">
    <infraLeafS name="leafs" type="range">
      <infraNodeBlk name="leaf1" from_="101" to_="101"/>
      <infraNodeBlk name="leaf3" from_="103" to_="103"/>
      <infraRsAccNodePGrp tDn="uni/infra/funcprof/accnodepgrp-myLeafCopp"/>
    </infraLeafS>
  </infraNodeP>
  <infraFuncP>
    <infraAccNodePGrp name="myLeafCopp">
      <infraRsLeafCoppProfile tnCoppLeafProfileName="mycustom"/>   <!-- bind copp leaf
policy to leaf </infraAccNodePGrp>
profile -->
  </infraFuncP>
</infraInfra>
```

ステップ 2 CoPP スパイン プロファイルを設定します。

例：

```
<!-- api/node/mo/uni/.xml -->
<infraInfra>
  <coppSpineProfile type="custom" name="mycustomSpine">             <!-- define copp
leaf profile -->
    <coppSpineGen1CustomValues bgpBurst="150" bgpRate="300"/>
  </coppSpineProfile>
  <infraSpineP name="spineCopp">
    <infraSpineS name="spines" type="range">
      <infraNodeBlk name="spine1" from_="104" to_="104"/>
      <infraRsSpineAccNodePGrp tDn="uni/infra/funcprof/spaccnodepgrp-mySpineCopp"/>
    </infraSpineS>
  </infraSpineP>
  <infraFuncP>
    <infraSpineAccNodePGrp name="mySpineCopp">
```

```
      <infraRsSpineCoppProfile tnCoppSpineProfileName="mycustomSpine"/> <!-- bind copp
spine policy to
    </infraSpineAccNodePGrp>                                                    spine profile
 -->
  </infraFuncP>
</infraInfra>
```

GUI を使用した CoPP 統計情報の表示

CoPP の調整を適切に行うには、指定のモードの指定のプロトコルでドロップ/許可されたパケット数を知る必要があります。次の手順を使用して、GUI で情報を表示できます。

手順

メニュー バーで、**[ファブリック]** > **[インベントリ]** > **[ポッド]/番号/** > **[ノード]/名前/** > **[コントロール プレーンの統計情報]** > **[デフォルト]** の順にクリックして、クラスのリストから選択し、統計情報の表示形式を設定します。

CoPP によって許可またはドロップされたパケット数に関する統計情報を収集することができます。

APIC GUI を使用したプロトコル CoPP ポリシーごとの各インターフェイスの設定

手順

ステップ 1 メニュー バーで、**[ファブリック]** > **[外部アクセス ポリシー]** をクリックします。

ステップ 2 **[ナビゲーション]** ペインで、**[ポリシー]** > **[インターフェイス]** > **[CoPP インターフェイス]** を展開して、**[プロトコル CoPP ポリシーごとの各インターフェイスの作成]** ダイアログ ボックスを右クリックして、**[プロトコル CoPP ポリシーごとの各インターフェイスの作成]** ダイアログ ボックスの次のアクションを実行します。

a) **[名前]** フィールドでポリシー名を追加します。

b) **[CoPP ポリシー プロトコル]** 表を展開し、プロトコル名、タイプ、レート、バースト情報を入力します。**[更新]** と **[送信]** をクリックします。

ステップ 3 **[ナビゲーション]** ペインで、**[インターフェイス]** > **[リーフ インターフェイス]** > **[ポリシー グループ]** > **[リーフ アクセス ポート ポリシー グループの作成]** を展開して、**[リーフ アクセス ポート ポリシー グループの作成]** ダイアログ ボックスを右クリックして、**[リーフ アクセス ポート ポリシー グループの作成]** ダイアログ ボックスの次のアクションを実行します。

a) **[名前]** フィールドでポリシー名を追加します。

b) **[COPP リーフ ポリシー]** フィールドで、以前に作成されたポリシーを選択します。
c) [Submit] をクリックします。

ステップ 4 **[ナビゲーション]** ペインで、**[インターフェイス]** > **[リーフ インターフェイス]** > **[プロファイル]** > **[リーフ プロファイル]** を展開して、**[リーフ インターフェイス プロファイルの作成]** ダイアログ ボックスを右クリックして、**[リーフ インターフェイス プロファイルの作成]** ダイアログ ボックスの次のアクションを実行します。

a) **[名前]** フィールドで、プロファイル名を追加します。
b) **[インターフェイス セレクタ]** 表を展開し、**[名前]** および **[インターフェイス ID]** フィールドにインターフェイス情報を追加して、以前作成した **[インターフェイス ポリシー グループ]** を選択します。
c) **[Ok]** および **[送信]** をクリックして、プロトコル CoPP ごとの各インターフェイス設定を完了します。

NX-OS スタイル CLI を使用するプロトコル CoPP ポリシーごとのインターフェイスごとの設定

手順

ステップ 1 CoPP クラス マップおよびポリシー マップを定義します。

例：

```
(config)# policy-map type control-plane-if <name>
        (config-pmap-copp)# protocol bgp bps <value>
        (config-pmap-copp)# protocol ospf bps <value>
```

ステップ 2 リーフのインターフェイスに設定を適用します。

例：

```
(config)# leaf 101
        (config-leaf)# int eth 1/10
        (config-leaf-if)# service-policy type control-plane-if output<name>
```

RESTAPIを使用するプロトコルごとのインターフェイスあたりのCoPPの設定

手順

プロトコルごとにインターフェイスあたりの CoPP を設定します。

例：

```
<polUni>
    <infraInfra>
    <infraNodeP name="default">
        <infraLeafS name="default" type="range">
            <infraNodeBlk name="default" to_="101" from_="101"/>
        </infraLeafS>
        <infraRsAccPortP tDn="uni/infra/accportprof-default"/>
    </infraNodeP>
    <infraAccPortP name="default">
        <infraHPortS name="regularPorts" type="range">
            <infraPortBlk name="blk1" toPort="7" fromPort="1" toCard="1" fromCard="1"/>
                <infraRsAccBaseGrp tDn="uni/infra/funcprof/accportgrp-copp"/>
        </infraHPortS>
    </infraAccPortP>

    <infraFuncP>
        <infraAccPortGrp name="copp">
            <infraRsCoppIfPol tnCoppIfPolName="pc"/>
        </infraAccPortGrp>
    </infraFuncP>

    <coppIfPol name = "pc" >
        <coppProtoClassP name = "test" matchProto="lldp,arp" rate="505" burst = "201"/>
        <coppProtoClassP name = "test1" matchProto="bgp" rate="500" burst = "200" />
    </coppIfPol>
</infraInfra>
</polUni>
```

CoPP プレフィルタについて

DDoS 攻撃に対する保護のため、CoPP プレフィルタ プロファイルはスパインとリーフ スイッチで使用され、指定されたソースと TCP ポートに基づく認証サービスへのアクセスをフィルタします。CoPP プレフィルタ プロファイルがスイッチに展開されるとき、デフォルトでコントロール プレーン トラフィックは拒否されます。CoPP プレフィルタ プロファイルで指定されたトラフィックのみが許可されます。

サポートされるプラットフォーム

このセクションでは、CoPP プレフィルター機能のサポートされているプラットフォームを示します。

リーフ スイッチがサポートされています。

- N9K-C93108TC-EX
- N9K-C93108TC-FX
- N9K-C93108YC-FX
- N9K-C93180LC-EX
- N9K-C93180YC-EX

- N9K-C9348GC-FXP

スパイン スイッチがサポートされています。

- N9K-C92300YC
- N9K-C92304QC
- N9K-C9232C
- N9K-C9236C
- N9K-C9272Q
- N9K-C9364C
- N9K C9508 FM 2
- N9K-C9516-FM-E2

制限事項

- イーサネット タイプ IPv4 または IPv6 パケットだけは、出力 TCAM で一致することができます。ARP ND パケットが一致しません。
- 合計 128 (ワイドキー) エントリの許可リストに含めることができます。ただし、一部のエントリは、社外秘予約されています。

GUI を使用した CoPP プレフィルタ、ポリシー グループ、プロファイルの設定

Cisco APIC GUI を使用した CoPP プレフィルタの設定

このセクションでは、リーフ レベルとスパイン レベルは、Cisco APIC GUI を使用して、CoPP プレフィルタを設定する方法について説明します。

始める前に

APIC GUI へのアクセス

手順

ステップ 1 **[Fabric]** > **[External Access Policies]**をクリックします 。

ステップ 2 [Navigation] ペインで、**[Policies]** > **[Switch]** をクリックします。
[Navigation] ペインに [CoPP Pre-Filter for Leaf] および [CoPP Pre-Filter for Spine] ノードが表示されます。

ステップ 3 [Navigation] ペインで、次のオプションから選択します。

- [CoPP Pre-Filter for Leaf] – リーフ スイッチの CoPP プレフィルタを作成する場合は、[CoPP Pre-Filter for Leaf] を右クリックして、[Create Profiles for CoPP Pre-Filter To Be Applied At The Leaf Level] を選択します。
- [CoPP Pre-Filter for Spine] – スパイン スイッチの CoPP プレフィルタを作成する場合は、[CoPP Pre-Filter for Spine] を右クリックして、[Create Profiles for CoPP Pre-Filter To Be Applied At The Spine Level] を選択します。

それぞれの CoPP プレフィルターのダイアログが表示されます。

ステップ 4 ダイアログのフィールドに適切な値を入力します。

（注） ダイアログ ボックスのフィールドの詳細については、ヘルプ アイコンをクリックすると Cisco APIC のヘルプ ファイルが表示されます。

ステップ 5 完了したら、[Submit] をクリックします。

次のタスク

ポリシー グループを設定します。

GUI を使用したリーフ ポリシー グループの設定

このセクションでは、ポリシー グループを作成する方法について説明します。

始める前に

Cisco APIC GUI にアクセスします。

手順

ステップ 1 **[Fabric]** > **[External Access Policies]**をクリックします 。

ステップ 2 [**ナビゲーション**] ペインで、**[スイッチ]** > **[リーフ スイッチ]** をクリックします。
[ポリシー グループ] ノードが **[ナビゲーション]** ウィンドウに表示されます。

ステップ 3 [**ナビゲーション**] ペインの **[ポリシー グループ]** で、リーフ ポリシー グループを作成するには、**[ポリシー グループ]** を右クリックして、**[アクセス スイッチ ポリシー グループの作成]** をクリックします。

それぞれのポリシー グループ ダイアログが表示されます。

ステップ 4 ポリシー グループ ダイアログから、**[名前]** フィールドに名前を入力して、適用するポリシー タイプのドロップダウン矢印をクリックします。選択したポリシータイプに設定されているポリシーがドロップダウン リストに表示されます。

（注） ダイアログ ボックスのフィールドの詳細については、ヘルプ アイコンをクリックすると Cisco APIC ヘルプ ファイルが表示されます。

ステップ 5 完了したら、[送信（Submit）] をクリックします。

次のタスク

プロファイルを設定します。

GUI を使用したリーフ プロファイルの設定

このセクションでは、プロファイルを作成する方法について説明します。

始める前に

設定されているポリシー グループが必要です。

手順

ステップ 1 **[Fabric]** > **[External Access Policies]**をクリックします 。

ステップ 2 [**ナビゲーション**] ペインで、**[スイッチ]** > **[リーフ スイッチ]** > **[プロファイル]** をクリックします。
[リーフ プロファイル] ノードが **[ナビゲーション]** ウィンドウに表示されます。

ステップ 3 [**ナビゲーション**] ペインの **[プロファイル]** で、リーフ スイッチのプロファイルを作成するには、**[プロファイル]** を右クリックして **[リーフ プロファイルの作成]** を選択します。
個別にプロファイル ダイアログが表示されます。

ステップ 4 プロファイル ダイアログから **[名前]** フィールドに名前を入力し、**[+]** をクリックしてセレクタ情報を入力します。完了したら、[Update] をクリックします。
[更新] をクリックした後、プロファイル ダイアログに戻ります。

ステップ 5 **[次へ]** をクリックして、インターフェイス セレクタ プロファイル情報を入力します。

（注） ダイアログ ボックスのフィールドの詳細については、ヘルプ アイコンをクリックすると Cisco APIC ヘルプ ファイルが表示されます。

ステップ 6 完了したら、**[終了]** をクリックします。

CLI を使用した CoPP プレフィルタの設定

CLI を使用したリーフ スイッチの CoPP プレフィルタの設定

このセクションでは、CoPP プレフィルタ ポリシーとポリシー グループを設定し、CLI を使用してスイッチ ポリシー グループとスイッチ プロファイルを関連付ける方法を説明します。

手順

ステップ 1 Switch# **configure terminal**

グローバル コンフィギュレーション モードを開始します。

ステップ 2 Switch(config)# **template control-plane-policing-prefilter-leaf** <*name*>

リーフ スイッチの CoPP プレフィルタ プロファイルを作成します。

ステップ 3 Switch (config-control-plane-policing-prefilter-leaf)# **permit proto** { **tcp** | **udp** | **eigrp** | **unspecified** | **icmp** | **icmpv6** | **egp** | **igp** | **l2tp** | **ospf** | **pim** }

指定された IP プロトコルを許可します。

ステップ 4 Switch (config-control-plane-policing-prefilter-leaf)#**exit**

グローバル コンフィギュレーション モードを開始します。

ステップ 5 Switch(config)# **template leaf-policy-group** <*name*>

CoPP プレフィルタ ポリシー グループ リーフ スイッチを作成します。

ステップ 6 Switch(config-leaf-policy-group)# **control-plane-policing-prefilter** <*name*>

CoPP プレフィルタ ポリシーとリーフ ポリシー グループを関連付けます。

ステップ 7 Switch(config-leaf-policy-group)# **exit** <*name*>

グローバル コンフィギュレーション モードを開始します。

ステップ 8 Switch(config)# **leaf-profile** <*name*>

リーフ プロファイルを作成します。

ステップ 9 Switch(config-leaf-profile)# **leaf-group** <*name*>

リーフ プロファイルとリーフ グループを関連付けます。

ステップ 10 Switch(config-leaf-group)# **leaf-policy-group** <*name*>

リーフ グループとリーフ ポリシー グループを関連付けます。

CLI を使用したスパイン スイッチの CoPP プレフィルタの設定

このセクションでは、CoPP プレフィルタ ポリシーとポリシー グループを設定し、CLI を使用してスイッチ ポリシー グループとスイッチ プロファイルを関連付ける方法を説明します。

手順

ステップ 1 Switch# **configure terminal**

グローバル コンフィギュレーション モードを開始します。

ステップ 2 Switch(config)# **template control-plane-policing-prefilter-spine** <*name*>

スパイン スイッチの CoPP プレフィルタ プロファイルを作成します。

ステップ 3 Switch (config-control-plane-policing-prefilter-spine)# **permit proto** { **tcp** | **udp** | **eigrp** | **unspecified** | **icmp** | **icmpv6** | **egp** | **igp** | **l2tp** | **ospf** | **pim** }

指定された IP プロトコルを許可します。

ステップ 4 Switch (config-control-plane-policing-prefilter-spine)#**exit**

グローバル コンフィギュレーション モードを開始します。

ステップ 5 Switch(config)# **template spine-policy-group** <*name*>

CoPP プレフィルタ ポリシー グループ スパイン スイッチを作成します。

ステップ 6 Switch(config-spine-policy-group)# **control-plane-policing-prefilter** <*name*>

CoPP プレフィルタ ポリシーとスパイン ポリシー グループを関連付けます。

ステップ 7 Switch(config-spine-policy-group)# **exit** <*name*>

グローバル コンフィギュレーション モードを開始します。

ステップ 8 Switch(config)# **spine-profile** <*name*>

スパイン プロファイルを作成します。

ステップ 9 Switch(config-spine-profile)# **spine-group** <*name*>

スパイン プロファイルとスパイン グループを関連付けます。

ステップ 10 Switch(config-spine-group)# **spine-policy-group** <*name*>

スパイン グループとスパイン ポリシー グループを関連付けます。

REST API を使用した CoPP プレフィルタの設定

REST API を使用したリーフ スイッチの CoPP プレフィルタ ポリシーの設定

このセクションでは、REST API を使用してリーフ スイッチの CoPP プレフィルタ ポリシーを設定する方法について説明します。

手順

ステップ 1 許可リストのエントリとともに CoPP プレフィルタのスイッチ ポリシーを作成します。

```
<iaclLeafProfile descr="" dn="uni/infra/iaclspinep-spine_icmp"
name="COPP_PreFilter_BGP_Config " ownerKey="" ownerTag="">
<iaclEntry dstAddr="0.0.0.0/0" dstPortFrom="179" dstPortTo="179" ipProto="tcp" name="bgp"
 nameAlias="" srcAddr="0.0.0.0/0" srcPortFrom="179" srcPortTo="179"/>
</iaclLeafProfile>
```

ステップ 2 CoPP プレフィルタ ポリシーでスイッチ ポリシー グループを作成します。

```
<infraAccNodePGrp descr="" dn="uni/infra/funcprof/accnodepgrp-COPP_PreFilter_BGP_Config
 " name="COPP_PreFilter_BGP_Config" nameAlias="" ownerKey="" ownerTag="">
<infraRsIaclLeafProfile tnIaclLeafProfileName="COPP_PreFilter_BGP_Config"/>
</infraAccNodePGrp>
```

ステップ 3 スイッチ プロファイルにスイッチ ポリシー グループを関連付けます。

```
<infraNodeP descr="" dn="uni/infra/nprof-leafP-103" name="leafP-103" nameAlias=""
ownerKey="" ownerTag="">
<infraLeafS descr="" name="103_Sel" nameAlias="" ownerKey="" ownerTag="" type="range">
<infraRsAccNodePGrp tDn="uni/infra/funcprof/accnodepgrp-COPP_PreFilter_BGP_Config"/>
<infraNodeBlk descr="" from_="103" name="nblk1" nameAlias="" to_="103"/>
</infraLeafS>
</infraNodeP>
```

REST API を使用したスパインの CoPP プレフィルタ ポリシーの設定

このセクションでは、REST API を使用してスパイン スイッチの CoPP プレフィルタ ポリシーを設定する方法について説明します。

手順

ステップ 1 許可リストのエントリとともに CoPP プレフィルタのスイッチ ポリシーを作成します。

```
<iaclSpineProfile descr="" dn="uni/infra/iaclspinep-spine_icmp"
name="COPP_PreFilter_OSPF_Config" ownerKey="" ownerTag="">
<iaclEntry dstAddr="0.0.0.0/0" dstPortFrom="unspecified" dstPortTo="unspecified"
ipProto="ospfigp" name="" nameAlias="" srcAddr="0.0.0.0/0" srcPortFrom="unspecified"
srcPortTo="unspecified"/>
</iaclSpineProfile>
```

ステップ 2 CoPP プレフィルタ ポリシーでスイッチ ポリシー グループを作成します。

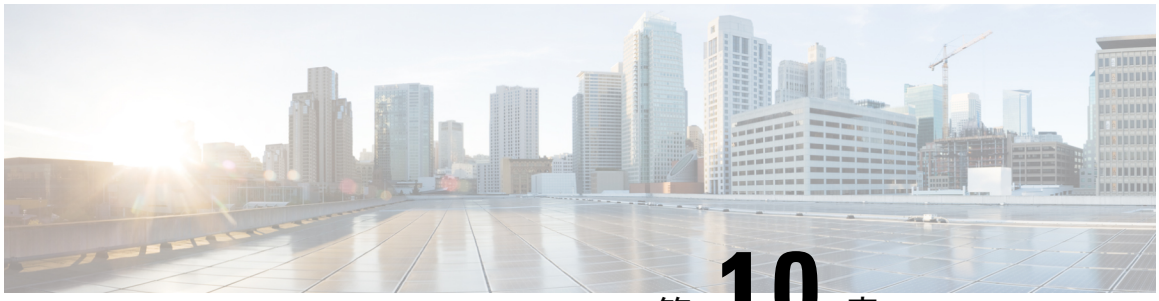
```
<infraSpineAccNodePGrp descr=""
dn="uni/infra/funcprof/spaccnodepgrp-COPP_PreFilter_OSPF_Config"
name="COPP_PreFilter_OSPF_Config" nameAlias="" ownerKey="" ownerTag="">
<infraRsIaclSpineProfile tnIaclSpineProfileName="COPP_PreFilter_OSPF_Config"/>
</infraSpineAccNodePGrp>
```

ステップ 3 スイッチ プロファイルにスイッチ ポリシー グループを関連付けます。

```
<infraSpineP descr="" dn="uni/infra/spprof-204" name="204" nameAlias="" ownerKey=""
ownerTag="">
<infraSpineS descr="" name="204" nameAlias="" ownerKey="" ownerTag="" type="range">
<infraRsSpineAccNodePGrp
```

```
tDn="uni/infra/funcprof/spaccnodepgrp-COPP_PreFilter_OSPF_Config"/>
<infraNodeBlk descr="" from_="204" name="nodeblock1" nameAlias="" to_="204"/>
</infraSpineS>
<infraRsSpAccPortP tDn="uni/infra/spaccportprof-204"/>
</infraSpineP>
```

次のタスク



第 10 章

ファブリック セキュリティ

この章の内容は、次のとおりです。

連邦情報処理標準（FIPS）について

連邦情報処理標準（FIPS）発行 140-2、暗号化モジュールのセキュリティ要件では、暗号化モジュールの米国政府要件が詳述されています。FIPS 140-2 では、暗号モジュールがハードウェア、ソフトウェア、ファームウェア、または何らかの組み合わせのセットで、暗号機能またはプロセスを実装し、暗号アルゴリズムおよび任意のキー生成機能を含み、明確に定義された暗号境界の内部に位置しなければならないと定義しています。

FIPS は特定の暗号アルゴリズムがセキュアであることを条件とするほか、ある暗号モジュールが FIPS 準拠であると称する場合は、どのアルゴリズムを使用すべきかも指定しています。

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。

- FIPS を有効にすると、Cisco APIC 全体に適用されます。
- Cisco APIC ソフトウェアのダウン グレードを実行しているときに FIPS を最初に無効にする必要があります。
- パスワードは最小限 8 文字の長さで作成してください。
- Telnet をディセーブルにします。ユーザのログインは SSH だけで行ってください。
- SSH サーバの RSA1 キー ペアすべてを削除してください。

- RADIUS/TACACS+によるリモート認証をディセーブルにしてください。ローカルとLDAPユーザのみを認証できます。
- セキュア シェル（SSH） および SNMP がサポートされます。
- SNMP v1 および v2 をディセーブルにしてください。SNMPv3 に対して設定された、スイッチ上の既存ユーザ- アカウントのいずれについても、認証およびプライバシー用 AES3 は SHA でのみ設定されていなければなりません。
- リリース 2.3(1x) で始まる、FIPS は、スイッチのレベルで設定できます。
- リリース 3.1(1x) から始まる FIPs を有効にすると、NTP が FIPSモードの動作は、HMAC SHA1 による認証と認証なしで FIPSモード NTP をサポートしています。

GUI を使用した Cisco APIC の FIPS の設定

FIPS を有効にすると、Cisco APIC 全体に適用されます。

手順

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 **[ナビゲーション]** ペインで、**[AAA]** > **[ファブリック セキュリティ]** を展開します。

ステップ 3 **[作業]** ペインの **[プロパティ]** 領域で、目的の FIPSモードを選択します。

FIPSモードのオプションは、**[無効化]** と **[有効化]** です。デフォルト値は [Disable] です。

（注） 設定を完了するには再起動する必要があります。モードを変更すると、設定を完了するため必ず再起動する必要があります。

NX-OS スタイル CLI を使用した Cisco APIC の FIPS 設定

FIPS を有効にすると、Cisco APIC 全体に適用されます。

手順

	コマンドまたはアクション	目的
ステップ 1	**configure** 例： `apic1# configure`	コンフィギュレーション モードに入ります。

	コマンドまたはアクション	目的
ステップ 2	**fips mode enable** 例： `apic1(config)# fips mode enable`	FIP を有効にします。**no fips mode enable** コマンドにより FIPS が無効になります。 設定を完了するため再起動する必要があります。モードを変更すると、設定を完了するため必ず再起動する必要があります。

REST API を使用した Cisco APIC の FIPS の設定

FIPS を有効にすると、Cisco APIC 全体に適用されます。

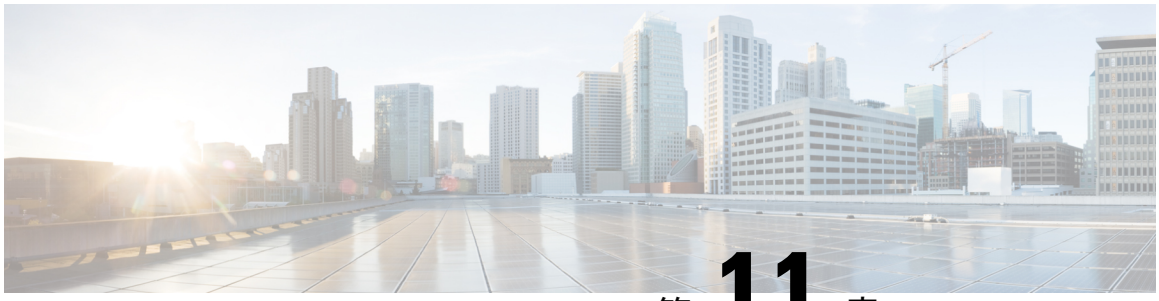
手順

すべてのテナントの FIPS を設定します。

例：

```
https://apic1.cisco.com/api/node/mo/uni/userext.xml
<aaaFabricSec fipsMode="enable" />
```

（注） 設定を完了するには再起動する必要があります。モードを変更すると、設定を完了するため必ず再起動する必要があります。



第 11 章

セキュリティ ポリシー

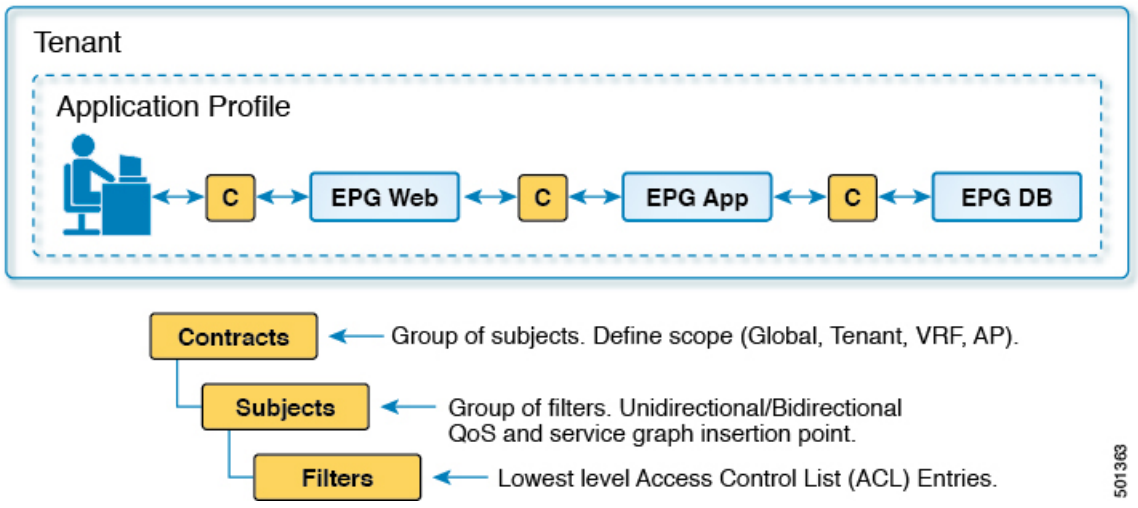
この章の内容は、次のとおりです。

ACI ファブリック ネットワーク アクセス セキュリティ ポリシー モデル（契約）

ACI のファブリック セキュリティ ポリシー モデルはコントラクトに基づいています。このアプローチにより、従来のアクセス コントロール リスト（ACL）の制限に対応できます。コントラクトには、エンドポイント グループ間のトラフィックで適用されるセキュリティ ポリシーの仕様が含まれます。

次の図は、 契約のコンポーネントを示しています。

図 *5:* 契約のコンポーネント



EPG 通信にはコントラクトが必要です。EPG/EPG 通信はコントラクトなしでは許可されません。APICは、コントラクトや関連するEPGなどのポリシーモデル全体を各スイッチの具象モデルにレンダリングします。入力時に、ファブリックに入るパケットはすべて、必要なポリシーの詳細でマークされます。EPGの間を通過できるトラフィックの種類を選択するためにコントラクトが必要とされるので、コントラクトはセキュリティポリシーを適用します。コントラクトは、従来のネットワーク設定でのアクセス コントロール リスト（ACL）によって扱われるセキュリティ要件を満たす一方で、柔軟性が高く、管理が容易な、包括的なセキュリティ ポリシー ソリューションです。

アクセス コントロール リストの制限

従来のアクセス コントロール リスト（ACL）には、ACI ファブリック セキュリティ モデルが対応する多数の制限があります。従来の ACL は、ネットワーク トポロジと非常に強固に結合されています。それらは通常、ルータまたはスイッチの入力および出力インターフェイスごとに設定され、そのインターフェイス、およびそれらのインターフェイスを流れることが予期されるトラフィックに合わせてカスタマイズされます。このカスタマイズにより、それらは多くの場合インターフェイス間で再利用できません。もちろんこれはルータまたはスイッチ間にも当てはまります。

従来の ACL は、非常に複雑で曖昧です。なぜなら、そのリストには、許可された特定の IP アドレス、サブネット、およびプロトコルのリストと、明確に許可されていない多くのものが含まれているためです。この複雑さは、問題が生じるのを管理者が懸念して ACL ルールを削除するのを躊躇するため、維持が困難で、多くの場合は増大するだけということを意味します。複雑さは、それらが通常 WAN と企業間または WAN とデータセンター間の境界などのネットワーク内の特定の境界ポイントでのみ配置されていることを意味します。この場合、ACLのセキュリティのメリットは、エンタープライズ内またはデータセンターに含まれるトラフィック向けには生かされません。

別の問題として、1 つの ACL 内のエントリ数の大幅増加が考えられます。ユーザは多くの場合、一連の送信元が一連のプロトコルを使用して一連の宛先と通信するのを許可する ACL を作成します。最悪の場合、*N*の送信元が*K*のプロトコルを使用して*M*の宛先と対話する場合、ACLに*N***M***K*の行が存在する場合があります。ACLは、プロトコルごとに各宛先と通信する各送信元を一覧表示する必要があります。また、ACLが非常に大きくなる前に多くのデバイスやプロトコルを取得することはありません。

ACI ファブリック セキュリティ モデルは、これらの ACL の問題に処理します。ACI ファブリックセキュリティモデルは、管理者の意図を直接表します。管理者は、連絡先、フィルタ、およびラベルの管理対象オブジェクトを使用してエンドポイントのグループがどのように通信するかを指定します。これらの管理対象オブジェクトは、ネットワークのトポロジに関連していません。なぜなら、それらは特定のインターフェイスに適用されないためです。それらは、エンドポイントのこれらのグループの接続場所に関係なく、ネットワークが強要しなければならない簡易なルールです。このトポロジの独立性は、これらの管理対象オブジェクトが特定の境界ポイントとしてだけではなくデータセンター全体にわたって容易に配置して再利用できることを意味します。

ACI ファブリック セキュリティ モデルは、エンドポイントのグループ化コンストラクトを直接使用するため、サーバのグループが相互に通信できるようにするための概念はシンプルで

す。1 つのルールにより、任意の数の送信元が同様に任意の数の宛先と通信することを可能にできます。このような簡略化により、そのスケールと保守性が大幅に向上します。つまり、データセンター全体でより簡単に使用できることにもつながります。

セキュリティ ポリシー仕様を含むコントラクト

ACI セキュリティ モデルでは、コントラクトに EPG 間の通信を管理するポリシーが含まれます。コントラクトは通信内容を指定し、EPG は通信の送信元と宛先を指定します。コントラクトは次のように EPG をリンクします。

EPG 1 ---------- コントラクト ---------- EPG 2

コントラクトで許可されていれば、EPG 1 のエンドポイントは EPG 2 のエンドポイントと通信でき、またその逆も可能です。このポリシーの構造には非常に柔軟性があります。たとえば、EPG 1 と EPG 2 間には多くのコントラクトが存在でき、1 つのコントラクトを使用する EPG が 3 つ以上存在でき、コントラクトは複数の EPG のセットで再利用できます。

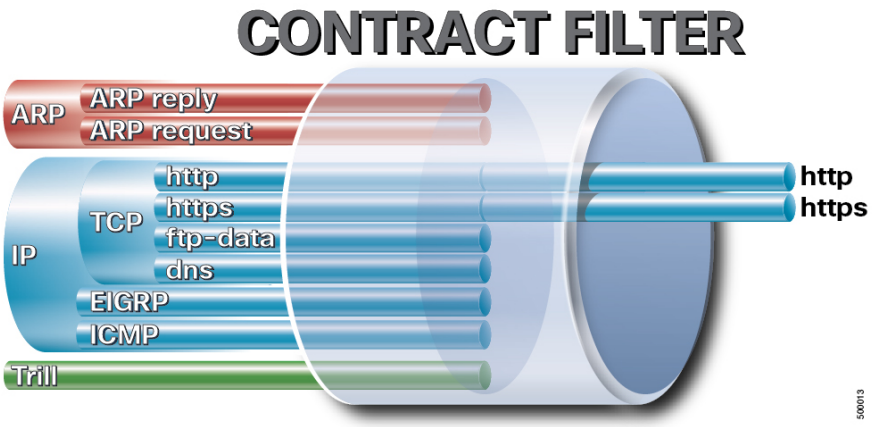
また EPG とコントラクトの関係には方向性があります。EPG はコントラクトを提供または消費できます。コントラクトを提供する EPG は通常、一連のクライアント デバイスにサービスを提供する一連のエンドポイントです。そのサービスによって使用されるプロトコルはコントラクトで定義されます。コントラクトを消費する EPG は通常、そのサービスのクライアントである一連のエンドポイントです。クライアント エンドポイント (コンシューマ) がサーバ エンドポイント (プロバイダー) に接続しようとすると、コントラクトはその接続が許可されるかどうかを確認します。特に指定のない限り、そのコントラクトは、サーバがクライアントへの接続を開始することを許可しません。ただし、EPG 間の別のコントラクトが、その方向の接続を簡単に許可する場合があります。

この提供/消費の関係は通常、EPG とコントラクト間を矢印を使って図で表されます。次に示す矢印の方向に注目してください。

EPG 1 <------- 消費 -------- コントラクト <------- 提供 -------- EPG 2

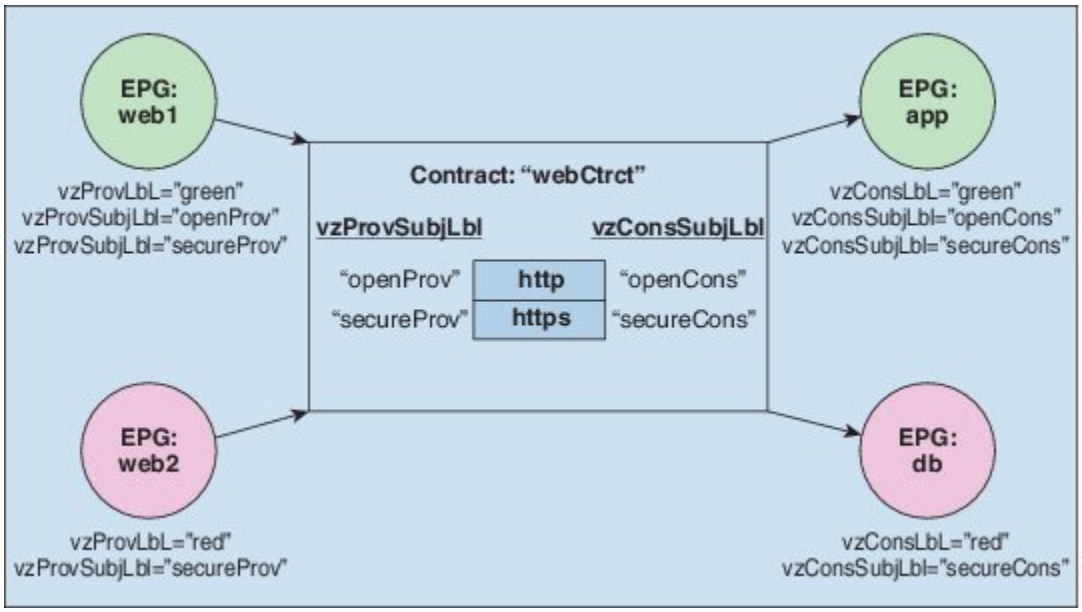
コントラクトは階層的に構築されます。1 つ以上のサブジェクトで構成され、各サブジェクトには 1 つ以上のフィルタが含まれ、各フィルタは 1 つ以上のプロトコルを定義できます。

図 *6:* コントラクト フィルタ



次の図は、コントラクトが EPG の通信をどのように管理するかを示します。

図 *7: EPG/EPG* 通信を決定するコントラクト



たとえば、TCP ポート 80 とポート 8080 を指定する HTTP と呼ばれるフィルタと、TCP ポート 443 を指定する HTTPS と呼ばれる別のフィルタを定義できます。その後、2 セットのサブジェクトを持つ webCtrct と呼ばれるコントラクトを作成できます。openProv と openCons are は HTTP フィルタが含まれるサブジェクトです。secureProv と secureCons は HTTPS フィルタが含まれるサブジェクトです。この webCtrct コントラクトは、Web サービスを提供する EPG とそのサービスを消費するエンドポイントを含む EPG 間のセキュアな Web トラフィックと非セキュアな Web トラフィックの両方を可能にするために使用できます。

これらの同じ構造は、仮想マシンのハイパーバイザを管理するポリシーにも適用されます。EPG が Virtual Machine Manager（VMM）のドメイン内に配置されると、APIC は EPG に関連付けられたすべてのポリシーを VMM ドメインに接続するインターフェイスを持つリーフ スイッチにダウンロードします。VMM ドメインの完全な説明については、『*Application Centric Infrastructure Fundamentals*』の「*Virtual Machine Manager Domains*」の章を参照してください。このポリシーが作成されると、APIC は EPG のエンドポイントへの接続を可能にするスイッチを指定する VMM ドメインにそれをプッシュ（あらかじめ入力）します。VMM ドメインは、EPG 内のエンドポイントが接続できるスイッチとポートのセットを定義します。エンドポイントがオンラインになると、適切な EPG に関連付けられます。パケットが送信されると、送信元 EPG および宛先 EPG がパケットから取得され、対応するコントラクトで定義されたポリシーでパケットが許可されたかどうかが確認されます。許可された場合は、パケットが転送されます。許可されない場合は、パケットはドロップされます。

コントラクトは 1 つ以上のサブジェクトで構成されます。各サブジェクトには 1 つ以上のフィルタが含まれます。各フィルタには 1 つ以上のエントリが含まれます。各エントリは、アクセス コントロール リスト (ACL) の 1 行に相当し、エンドポイント グループ内のエンドポイントが接続されているリーフ スイッチで適用されます。

詳細には、コントラクトは次の項目で構成されます。

- 名前：テナントによって消費されるすべてのコントラクト (common テナントまたはテナント自体で作成されたコントラクトを含む) にそれぞれ異なる名前が必要です。
- サブジェクト：特定のアプリケーションまたはサービス用のフィルタのグループ。
- フィルタ：レイヤ 2 ～ レイヤ 4 の属性 (イーサネット タイプ、プロトコル タイプ、TCP フラグ、ポートなど) に基づいてトラフィックを分類するために使用します。
- アクション：フィルタリングされたトラフィックで実行されるアクション。次のアクションがサポートされます。
  - トラフィックの許可 (通常のコントラクトのみ)
  - トラフィックのマーク (DSCP/CoS) (通常のコントラクトのみ)
  - トラフィックのリダイレクト (サービス グラフによる通常のコントラクトのみ)
  - トラフィックのコピー (サービス グラフまたは SPAN による通常のコントラクトのみ)
  - トラフィックのブロック（禁止コントラクトのみ）

    Cisco APIC リリース 3.2(x) および名前が EX または FX で終わるスイッチでは、標準コントラクトで代わりに件名 [拒否] アクションまたは [コントラクトまたは件名の除外] を使用して、指定のパターンを持つトラフィックをブロックできます。
  - トラフィックのログ（禁止コントラクトと通常のコントラクト）
- エイリアス：(任意) 変更可能なオブジェクト名。オブジェクト名は作成後に変更できませんが、エイリアスは変更できるプロパティです。

このように、コントラクトによって許可や拒否よりも複雑なアクションが可能になります。コントラクトは、所定のサブジェクトに一致するトラフィックをサービスにリダイレクトしたり、コピーしたり、その QoS レベルを変更したりできることを指定可能です。具象モデルでアクセス ポリシーをあらかじめ入力すると、APIC がオフラインまたはアクセスできない場合でも、エンドポイントは移動でき、新しいエンドポイントをオンラインにでき、通信を行うことができます。APIC は、ネットワークの単一の障害発生時点から除外されます。ACI ファブリックにパケットが入力されると同時に、セキュリティ ポリシーがスイッチで実行している具象モデルによって適用されます。

セキュリティ ポリシーの適用

トラフィックは前面パネルのインターフェイスからリーフ スイッチに入り、パケットは送信元 EPG の EPG でマーキングされます。リーフ スイッチはその後、テナント エリア内のパケットの宛先 IP アドレスでフォワーディング ルックアップを実行します。ヒットすると、次のシナリオのいずれかが発生する可能性があります。

1. ユニキャスト（/32）ヒットでは、宛先エンドポイントの EPG と宛先エンドポイントが存在するローカル インターフェイスまたはリモート リーフ スイッチの VTEP IP アドレスが提供されます。

2. サブネット プレフィクス（/32 以外）のユニキャスト ヒットでは、宛先サブネット プレフィクスの EPG と宛先サブネット プレフィクスが存在するローカル インターフェイスまたはリモート リーフ スイッチの VTEP IP アドレスが提供されます。
3. マルチキャスト ヒットでは、ファブリック全体の VXLAN カプセル化とマルチキャスト グループの EPG で使用するローカル レシーバのローカル インターフェイスと外側の宛先 IP アドレスが提供されます。

（注） マルチキャストと外部ルータのサブネットは、入力リーフ スイッチでのヒットを常にもたらします。セキュリティ ポリシーの適用は、宛先 EPG が入力リーフ スイッチによって認識されるとすぐに発生します。

転送テーブルの誤りにより、パケットがスパイン スイッチの転送プロキシに送信されます。転送プロキシはその後、転送テーブル検索を実行します。これが誤りである場合、パケットはドロップされます。これがヒットの場合、パケットは宛先エンドポイントを含む出力リーフ スイッチに送信されます。出力リーフ スイッチが宛先の EPG を認識するため、セキュリティ ポリシーの適用が実行されます。出力リーフ スイッチは、パケット送信元の EPG を認識する必要があります。ファブリック ヘッダーは、入力リーフ スイッチから出力リーフ スイッチに EPG を伝送するため、このプロセスをイネーブルにします。スパイン スイッチは、転送プロキシ機能を実行するときに、パケット内の元の EPG を保存します。

出力リーフ スイッチでは、送信元 IP アドレス、送信元 VTEP、および送信元 EPG 情報は、学習によってローカルの転送テーブルに保存されます。ほとんどのフローが双方向であるため、応答パケットがフローの両側で転送テーブルに入力し、トラフィックが両方向で入力フィルタリングされます。

マルチキャストおよび EPG セキュリティ

マルチキャスト トラフィックでは、興味深い問題が起こります。ユニキャスト トラフィックでは、宛先 EPG はパケットの宛先の検査からはっきり知られています。ただし、マルチキャスト トラフィックでは、宛先は抽象的なエンティティ、マルチキャスト グループです。パケットの送信元はマルチキャスト アドレスではないため、送信元 EPG は以前のユニキャストの例と同様に決定されます。宛先グループの起源はマルチキャストが異なる場所です。

マルチキャスト グループが、ネットワーク トポロジから若干独立しているので、グループ バインディングへの (S, G) および (*, G) の静的設定は受け入れ可能です。マルチキャスト グループが転送テーブルにある場合、マルチキャスト グループに対応する EPG は、転送テーブルにも配置されます。

（注） このマニュアルでは、マルチキャスト グループとしてマルチキャスト ストリームを参照します。

リーフ スイッチは、マルチキャスト ストリームに対応するグループを常に宛先 EPG と見なし、送信元 EPG と見なすことはありません。前述のアクセス コントロール マトリクスでは、マルチキャスト EPG が送信元の場合は行の内容は無効です。トラフィックは、マルチキャスト ストリームの送信元またはマルチキャスト ストリームに加わりたい宛先からマルチキャストストリームに送信されます。マルチキャストストリームが転送テーブルにある必要があり、ストリーム内に階層型アドレッシングがないため、マルチキャストトラフィックは、入力ファブリックの端でアクセスが制御されます。その結果、IPv4 マルチキャストは入力フィルタリングとして常に適用されます。

マルチキャストストリームの受信側は、トラフィックを受信する前にマルチキャストストリームに最初に加わる必要があります。IGMP Join 要求を送信すると、マルチキャスト レシーバは実際に IGMP パケットの送信元になります。宛先はマルチキャスト グループとして定義され、宛先 EPG は転送テーブルから取得されます。ルータが IGMP Join 要求を受信する入力点で、アクセス制御が適用されます。Join 要求が拒否された場合、レシーバはその特定のマルチキャスト ストリームからトラフィックを受信しません。

マルチキャスト EPG へのポリシーの適用は、前述のようにコントラクトのルールに従ってリーフ スイッチにより入力時に発生します。また、EPG バインディングに対するマルチキャストグループは、APIC によって特定のテナント（VRF）を含むすべてのリーフ スイッチにプッシュされます。

タブー

セキュリティを確保する通常のプロセスも適用されますが、ACI ポリシーモデルは、どのようなセキュリティ プラクティスが採用されても完全性を確保するのに役立ちます。ACI ポリシーモデルのアプローチでは、すべての通信がこれらの条件に準拠する必要があります。

- 通信は、モデルの管理対象オブジェクトであるコントラクトに基づいてのみ許可されます。コントラクトがなければ、EPG 間通信はデフォルトでディセーブルになります。
- ハードウェアへのダイレクト アクセスはなく、すべてのインタラクションはポリシー モデルを通じて管理されます。

禁止コントラクトは特定のトラフィックを拒否するために使用できます。そうしないと、コントラクトによって許可されます。ドロップされるトラフィックは、パターンと一致しています（すべての EPG、特定の EPG、フィルタに一致するトラフィックなど）。禁止ルールは単方向で、コントラクトを提供する EPG に対して一致するトラフィックを拒否します。

Cisco APIC リリース 3.2(x) および名前が EX または FX で終わるスイッチでは、標準コントラクトで代わりに件名 [拒否] アクションまたは [コントラクトまたは件名の除外] を使用して、指定のパターンを持つトラフィックをブロックできます。

ACL コントラクトおよび拒否ログの有効化および表示

ACL 契約の許可および拒否ログについて

契約ルールのトラフィック フローをログ記録および監視するには、契約の許可ルールのため送信されることが許可されたパケットまたはフローのログを有効化および表示するか、契約の許可ルールのためドロップされたフローのログを有効化および表示できます。

- 禁止契約拒否ルール
- 契約の件名でアクションを拒否する
- 契約または件名の例外
- ACL ファブリックの ACL 契約許可および拒否のログは、EX または FX で終わる名前の Nexus 9000 シリーズ スイッチと、それ以降のすべてのモデルでのみサポートされています。たとえば、N9K-C93180LC-EX や N9K-C9336C-FX のように指定してください。
- 管理契約のフィルタでログ directive を使用することはサポートされていません。ログ directive を設定すると、ゾーン分割ルールの展開エラーが発生します。

標準および禁止契約と件名についての詳細は、『*CiscoApplication Centric Infrastructure Fundamentals*』および『*Cisco APIC Basic Configuration Guide*』を参照してください。

ACL 許可および拒否ログ出力に含まれる EPG データ

Cisco APIC、リリース 3.2(1) まで、ACL 許可および拒否ログでは、記録されている契約に関連付けられた EPG を識別していませんでした。リリース 3.2(1) では、送信元 EPG と 送信先 EPG が ACI 許可および拒否ログの出力に追加されます。ACL 許可および拒否ログには、次の制限を持つ関連 EPG を含めます。

- ネットワーク内の EPG の配置によっては、ログの EPG データを使用できない場合があります。
- 設定の変更が発生するとき、ログデータが期限切れになっている可能性があります。安定した状態では、ログ データは正確です。

ログが次にフォーカスされているとき、許可および拒否ログの EPG データは最も正確になります。

- 入力 TOR で入力ポリシーがインストールされており、出力 TOR で出力ポリシーがインストールされている場合の EPG から EPG へのフロー。
- 境界リーフ TOR で 1 個のポリシーが適用され、非 BL TOR で他のポリシーが適用されている場合の EPG から L3Out へのフロー。

ログ出力の EPG は、共有サービス（共有 L3Outs を含む）で使用される uSeg Epg または Epg ではサポートされていません。

GUI を使用してACL 契約の許可とロギングの拒否を有効にする

次の手順では、GUI を使用してACL 契約の許可とロギングの拒否を有効にする方法を表示します。

（注） 許可ロギングを含むテナントは、EPG が関連する VRF を含むテナントです。これは必ずしも EPG と同じテナントや関連する契約である必要はありません。

手順

ステップ 1 メニュー バーで、**[Tenants]** > **[<tenant name>]** の順に選択します。

ステップ 2 [Navigation] ペインで、[Contracts] を展開し、 [Standard] を右クリックして [Create Contract] を選択します。

ステップ 3 [Create Contract] ダイアログボックスで、次の作業を実行します。

a) [Name] フィールドに、契約の名前を入力します。

b) [Scope] フィールドで、そのスコープ（[VRF]、[Tenant]、または [Global]）を選択します

c) オプション。契約に適用するターゲット DSCP または QoS クラスを設定します。

d) [+] アイコンをクリックして、[Subject] を展開します。

ステップ 4 [Create Contract Subject] ダイアログボックスで、次の操作を実行します。

ステップ 5 件名の名前と詳細な説明を入力します。

ステップ 6 オプション。ターゲット DSCP のドロップダウン リストから、件名に適用する DSCP を選択します。

ステップ 7 契約を両方向でなくコンシューマからプロバイダの方向にのみ適用するのでない限り、[Apply Both Directions] はオンにしたままにしておきます。

ステップ 8 [Apply Both Directions] をチェックしてない場合 [Reverse Filter Ports] をチェックしたままにして、ルールがプロバイダから消費者に適用されるようにレイヤ 4 ソースと宛先ポートを交換します。

ステップ 9 [+] アイコンをクリックして、[Filters] を展開します。

ステップ 10 [Name] ドロップダウン リストで、たとえば、**arp**、**default**、**est**、**icmp** などオプションを選択するか、以前設定したフィルタを選択します。

ステップ 11 [Directives] ドロップダウン リストで、[log] をクリックします。

ステップ 12 （任意） この件名で実行するアクションを [Deny] に変更します（またはアクションをデフォルトの [Permit] のままにします。

Directive：ログ有効化により、この件名のアクションが [Permit] になっている場合、ACL は件名と契約により制御されているフローとパケットを追跡します。この件名のアクションが [Deny] の場合、ACL の拒否ログはフローとパケットを追跡します。

ステップ 13 （任意） 件名の優先順位を設定します。

ステップ 14 [Update] をクリックします。

ステップ 15 [OK] をクリックします。

ステップ 16 [Submit] をクリックします。
ロギングがこの契約に対して有効になります。

NX-OS CLI を使用した ACL 契約許可ロギングの有効化

次の例は、NX-OS CLI を使用して契約許可ロギングを有効にする方法を示しています。

手順

ステップ 1 契約許可ルールにより送信できたパケットまたはフローのロギングを有効にするには、次のコマンドを使用します。

```
configure
tenant <tenantName>
contract <contractName> type <permit>
subject <subject Name>
access-group <access-list> <in/out/both> log
```

例：

次に例を示します。

```
apic1# configure
apic1(config)# tenant BDMode1
apic1(config-tenant)# contract Logicmp type permit
apic1(config-tenant-contract)# subject icmp
apic1(config-tenant-contract-subj)# access-group arp both log
```

ステップ 2 許可ロギングを無効にするには、**no** 形式の access-group コマンドを使用します。たとえば、`no access-group arp both log` コマンドを使用します。

REST API を使用した ACL 契約許可ロギングの有効化

次の例は、REST API を使用して許可および拒否ロギングを有効にする方法を示しています。この例では、ACL の許可を設定し、件名 Permit 設定し、設定されたアクションを拒否するには、契約のロギングを拒否します。

手順

この設定では、次の例のように XML で post を送信します。

例：

```
<vzBrCP dn="uni/tn-Tenant64/brc-C64" name="C64" scope="context">
    <vzSubj consMatchT="AtleastOne" name="HTTPSsbj" provMatchT="AtleastOne"
revFltPorts="yes" rn="subj-HTTPSsbj">
         <vzRsSubjFiltAtt action="permit" directives="log" forceResolve="yes"
```

```
priorityOverride="default"
rn="rssubjFiltAtt-PerHTTPS" tDn="uni/tn-Tenant64/flt-PerHTTPS" tRn="flt-PerHTTPS"
tnVzFilterName="PerHTTPS"/>
    </vzSubj>
    <vzSubj consMatchT="AtleastOne" name="httpSbj" provMatchT="AtleastOne"
revFltPorts="yes" rn="subj-httpSbj">
        <vzRsSubjFiltAtt action="deny" directives="log" forceResolve="yes"
priorityOverride="default"
rn="rssubjFiltAtt-httpFilter" tDn="uni/tn-Tenant64/flt-httpFilter" tRn="flt-httpFilter"
 tnVzFilterName="httpFilter"/>
    </vzSubj>
   <vzSubj consMatchT="AtleastOne" name="subj64" provMatchT="AtleastOne" revFltPorts="yes"
 rn="subj-subj64">
        <vzRsSubjFiltAtt action="permit" directives="log" forceResolve="yes"
priorityOverride="default"
rn="rssubjFiltAtt-icmp" tDn="uni/tn-common/flt-icmp" tRn="flt-icmp" tnVzFilterName="icmp"/>

    </vzSubj>
</vzBrCP>
```

GUI を使用した禁止契約拒否ロギングの有効化

次の手順は、GUI を使用して禁止コントラクトの拒否ロギングを有効にする方法を示しています。

手順

ステップ 1 メニュー バーで、**[Tenants]** > **[<tenant name>]** の順に選択します。

ステップ 2 [Navigation] ペインで、[Contracts] を展開します。

ステップ 3 [Taboos] を右クリックし、[Create Taboo Contract] を選択します。

ステップ 4 [Create Taboo Contract] ダイアログ ボックスで、次の操作を実行して禁止契約を指定します。

a) [Name] フィールドに、契約の名前を入力します。

b) オプション。[Description] フィールドに、禁止契約の説明を入力します。

c) [+] アイコンをクリックして、[Subject] を展開します。

ステップ 5 [Create Taboo Contract Subject] ダイアログ ボックスで、次の操作を実行します。

a) [Specify Identity of Subject] 領域に、名前と説明（オプション）を入力します。

b) [+] アイコンをクリックして、[Filters] を展開します。

c) [Name] ドロップダウン リストから、**<tenant_name>/arp**、**<tenant_name>/default**、**<tenant_name>/est**、**<tenant_name>/icmp** などのデフォルト値のいずれかを選択し、以前作成したフィルタか [Create Filter] を選択します。

（注） [Specify Filter Identity] 領域で [Create Filter] を選択した場合、次の操作を実行して、ACL 拒否ルールの基準を指定します。

1. 名前とオプションの説明を入力します。
2. [Entries] を展開し、ルールの名前を入力して、拒否するトラフィックを定義する条件を選択します。
3. [Directives] ドロップダウンリストで [log] を選択します。
4. [Update] をクリックします。
5. [OK] をクリックします。

ステップ 6 [Submit] をクリックします。
ロギングがこの禁止契約に対して有効になります。

NX-OS CLI を使用した禁止契約拒否ロギングの有効化

次の例は、NX-OS CLI を使用して禁止契約拒否ロギングを有効にする方法を示しています。

手順

ステップ 1 禁止契約拒否ルールのためにドロップされたパケットまたはフローのロギングを有効にするには、次のコマンドを使用します。

```
configure
tenant <tenantName>
contract <contractName> type <deny>
subject <subject Name>
access-group <access-list> <both> log
```

例：

次に例を示します。

```
apic1# configure
apic1(config)# tenant BDMode1
apic1(config-tenant)# contract dropFTP type deny
apic1(config-tenant-contract)# subject dropftp
apic1(config-tenant-contract-subj)# access-group ftp both log
```

ステップ 2 拒否ロギングを無効にするには、**no** 形式の access-group コマンドを使用します。たとえば、`no access-group https both log` コマンドを使用します。

REST API を使用した禁止契約拒否ロギングの有効化

次の例は、REST API を使用して禁止契約拒否ロギングを有効にする方法を示しています。

手順

タブー契約を設定するロギングを拒否する、次の例のように XML で post を送信します。

例：

```
<vzTaboo dn="uni/tn-Tenant64/taboo-TCtrctPrefix" name="TCtrctPrefix" scope="context">
    <vzTSubj name="PrefSubj" rn="tsubj-PrefSubj"">
        <vzRsDenyRule directives="log" forceResolve="yes" rn="rsdenyRule-default"
tCl="vzFilter"
tDn="uni/tn-common/flt-default" tRn="flt-default"/>
    </vzTSubj>
</vzTaboo>
```

GUI を使用した ACL 許可および拒否ログの表示

次の手順は、GUI を使用して、トラフィック フローの ACL 許可および拒否ログを（有効になっていれば）表示する方法を示しています。

手順

ステップ 1 メニュー バーで、**[Tenants]** > **[<tenant name>]** の順に選択します。

ステップ 2 [Navigation] ペインで、[Tenant <tenant name>] をクリックします。

ステップ 3 **Tenants <tenant name> [Work]** ペインで、**[Operational]** タブをクリックします。

ステップ 4 [Operational] タブの下で、[Flows] タブをクリックします。
[Flows] タブの下で、いずれかのタブをクリックして、レイヤ 2 許可ログ（[L2 Permit]）、レイヤ 3 許可ログ（[L3 Permit]）、レイヤ 2 拒否ログ（[L2 Drop]）、またはレイヤ 3 拒否ログ（[L3 Drop]）のログ データを表示します。各タブで、トラフィックがフローしていれば、ACL ロギング データを表示できます。データ ポイントは、ログ タイプと ACL ルールに応じて異なります。たとえば、[L3 Permit] ログおよび [L3 Deny] ログには次のデータ ポイントが含まれます。

- VRF
- Alias
- 送信元 IP アドレス
- 宛先 IP アドレス
- プロトコル
- 送信元ポート
- 宛先ポート
- 送信元 MAC アドレス
- 宛先 MAC アドレス

- Node
- 送信元インターフェイス
- VRF Encap
- 送信元 EPG
- 宛先 EPG
- 送信元 PC タグ
- 宛先 PC タグ

（注） また、[Flows] タブの横の [Packets] タブを使用して、シグニチャ、送信元、および宛先が同じであるパケットのグループ（最大 10 個）の ACL ログにアクセスできます。送信されたりドロップされたりするパケットのタイプを確認できます。

REST API を使用した ACL 許可および拒否ログ

次の例は、REST API を使用して、トラフィック フローのレイヤ 2 拒否ログ データを表示する方法を示しています。次の MO を使用してクエリを送信することができます。

- acllogDropL2Flow
- acllogPermitL2Flow
- acllogDropL3Flow
- acllogPermitL3Flow
- acllogDropL2Pkt
- acllogPermitL2Pkt
- acllogDropL3Pkt
- acllogPermitL3Pkt

始める前に

ACL 契約許可および拒否ログのデータを表示する前に、許可または拒否ロギングを有効にする必要があります。

手順

レイヤ 3 ドロップ ログ データを表示するには、REST API を使用して次のクエリを送信します。

```
GET https://apic-ip-address/api/class/acllogDropL3Flow
```

例：

次の例では、サンプル出力をいくつか示します。

```
<?xml version="1.0" encoding="UTF-8"?>
<imdata totalCount="2">
    <acllogPermitL3Flow childAction=""
dn="topology/pod-1/node-101/ndbgs/acllog/tn-common/ctx-inb

/permitl3flow-spctag-333-dpctag-444-sepgname-unknown-depgname-unknown-sip-[100:c000:a00:700:b00:0:f00:0]


-dip-[19.0.2.10]-proto-udp-sport-17459-dport-8721-smac-00:00:15:00:00:28-dmac-00:00:12:00:00:25-sintf-

        [port-channel5]-vrfencap-VXLAN: 2097153" dstEpgName="unknown" dstIp="19.0.2.10"
 dstMacAddr="00:00:12:00:00:25"
        dstPcTag="444" dstPort="8721" lcOwn="local" modTs="never" monPolDn=""
protocol="udp" srcEpgName="unknown"
        srcIntf="port-channel5" srcIp="100:c000:a00:700:b00:0:f00:0"
srcMacAddr="00:00:15:00:00:28" srcPcTag="333"
        srcPort="17459" status="" vrfEncap="VXLAN: 2097153"/>
    <acllogPermitL3Flow childAction=""
dn="topology/pod-1/node-102/ndbgs/acllog/tn-common/ctx-inb

/permitl3flow-spctag-333-dpctag-444-sepgname-unknown-depgname-unknown-sip-[100:c000:a00:700:b00:0:f00:0]-dip-


[19.0.2.10]-proto-udp-sport-17459-dport-8721-smac-00:00:15:00:00:28-dmac-00:00:12:00:00:25-sintf-

        [port-channel5]-vrfencap-VXLAN: 2097153" dstEpgName="unknown" dstIp="19.0.2.10"
 dstMacAddr="00:00:12:00:00:25"
        dstPcTag="444" dstPort="8721" lcOwn="local" modTs="never" monPolDn=""
protocol="udp" srcEpgName="unknown"
        srcIntf="port-channel5" srcIp="100:c000:a00:700:b00:0:f00:0"
srcMacAddr="00:00:15:00:00:28" srcPcTag="333"
        srcPort="17459" status="" vrfEncap="VXLAN: 2097153"/>
</imdata>
```

NX-OS CLI を使用した ACL 許可および拒否ログの表示

次の手順は、NX-OS スタイル CLI **show acllog** コマンドを使用して ACL ログの詳細を表示する方法を示しています。

レイヤ 3 コマンドの構文は、**show acllog {permit | deny} l3 {pkt | flow} tenant <tenant_name> vrf <vrf_name> srcip <source_ip> dstip <destination_ip> srcport <source_port> dstport <destination_port> protocol <protocol> srcintf <source_interface> start-time <startTime> end-time <endTime> detail** です。

レイヤ 2 コマンドの構文は、**show acllog {permit | deny} l2 {flow | pkt} tenant <tenant_name> vrf <VRF_name> srcintf <source_interface> vlan <VLAN_number> detail** です。

（注） **show acllog** コマンドの完全な構文は、第二世代 Cisco Nexus 9000 シリーズ スイッチ（N9K-C93180LC-EX など名前の最後に EX または FX がつく。もしくはそれ以降のシリーズ）および Cisco APIC リリース 3.2 以降でのみ使用できます。第一世代のスイッチ（名前の最後に EX または FX が付かない）または 3.2 以前の Cisco APIC リリースでは、使用可能な構文は上記の通りです。

Cisco APIC 3.2 以降では、追加のキーワードが **detail** keyword:**[dstEpgName <destination_EPG_name>| dstmac <destination_MAC_address> | dstpctag <destination_PCTag> | srcEpgName <source_EPG_name> | srcmac <source_MAC_address> | srcpctag <source_PCTag>|** とともにコマンドの両方のバージョンに追加されます。

手順

ステップ 1 次の例では、**show acllog drop l3 flow tenant common vrf default detail** コマンドを使用して、共通テナントのレイヤ 3 拒否ログに関する詳細情報を表示する方法を示します。

例：

```
apic1# show acllog deny l3 flow tenant common vrf default detail
SrcPcTag  : 49153
DstPcTag  : 32773
SrcEPG    : uni/tn-TSW_Tenant0/ap-tsw0AP0/epg-tsw0ctx0BD0epg6
DstEPG    : uni/tn-TSW_Tenant0/ap-tsw0AP0/epg-tsw0ctx0BD0epg5
SrcIp     : 16.0.2.10
DstIp     : 19.0.2.10
Protocol  : udp
SrcPort   : 17459
DstPort   : 8721
SrcMAC    : 00:00:15:00:00:28
DstMAC    : 00:00:12:00:00:25
Node      : 101
SrcIntf   : port-channel5
VrfEncap  : VXLAN: 2097153
```

この例では第二世代のスイッチまたは 3.2 以前の Cisco APIC リリースでの出力を示します。

ステップ 2 次の例では、**show acllog deny l2 flow tenant common vrf tsw0connctx0 detail** コマンドを使用して、共通テナントのレイヤ 3 拒否ログに関する詳細情報を表示する方法を示します。

例：

```
apic1# show acllog deny l2 flow tenant common vrf tsw0connctx0 detail
SrcPcTag DstPcTag   SrcEPG           DstEPG          SrcMAC            DstMAC
Node   SrcIntf   vlan
--------- --------- ----------------  --------------  ------------------
------------------   -----   ---------  ------
 32773    49153   uni/tn-TSW       uni/tn-TSW  00:00:11:00:00:11  11:00:32:00:00:33
101     port-     2
                  _Tenant0/ap-     _Tenant0/ap-
       channel8
                  tsw0AP0/epg-     tsw0AP0/epg-
                  tsw0ctx0BD0epg5  tsw0ctx0BD0epg6
```

この例では第二世代のスイッチまたは 3.2 以前の Cisco APIC リリースでの出力を示します。

ステップ 3 次の例では、**show acllog permit l3 pkt tenant <tenant name> vrf <vrf name> [detail]** コマンドを使用して、送信された一般的な VRF ACL レイヤ 3 許可パケットに関する詳細情報を表示する方法を示しています。

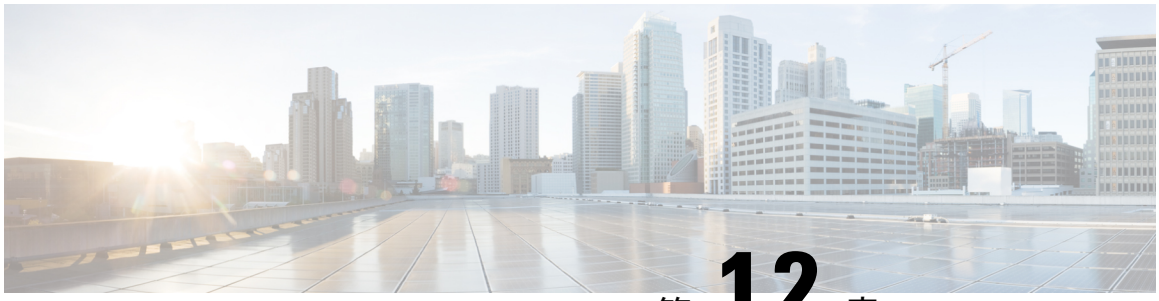
```
apic1# show acllog permit l3 pkt tenant common vrf default detail acllog permit l3 packets
 detail:
srcIp      : 10.2.0.19
dstIp      : 10.2.0.16
protocol   : udp
srcPort    : 13124
dstPort    : 4386
srcIntf    : port-channel5
vrfEncap   : VXLAN: 2097153
pktLen     : 112
srcMacAddr : 00:00:15:00:00:28
dstMacAddr : 00:00:12:00:00:25
timeStamp  : 2015-03-17T21:31:14.383+00:00
```

この例では第一世代のスイッチまたは 3.2 以前の Cisco APIC リリースでの出力を示します。

ステップ 4 次の例では、**show acllog permit l2 pkt tenant <tenant name> vrf <vrf name> srcintf <s interface>** コマンドを使用して、インターフェイス ポートチャネル 15 から送信されたデフォルトの VRF レイヤ 2 パケットに関する情報を表示する方法を示しています。

```
apic1# show acllog permit l2 pkt tenant common vrf default srcintf port-channel5

acllog permit L2 Packets
      Node          srcIntf       pktLen     timeStamp
 --------------  --------------  --------  --------------
                 port-channel5      1      2015-03-17T21:
                                           31:14.383+00:00
```

この例では第一世代のスイッチまたは 3.2 以前の Cisco APIC リリースでの出力を示します。



第 12 章

データ プレーン ポリシング

この章の内容は、次のとおりです。

概要

この記事では、データ プレーン ポリシングを設定する方法の例について説明します。

データ プレーン ポリシング（DPP）を使用して、ACI ファブリック アクセス インターフェイスの帯域幅使用量を管理します。DPP ポリシーは出力トラフィック、入力トラフィック、またはその両方に適用できます。DPP は特定のインターフェイスのデータ レートを監視します。データ レートがユーザ設定値を超えると、ただちにパケットのマーキングまたはドロップが発生します。ポリシングではトラフィックがバッファリングされないため、伝搬遅延への影響はありません。トラフィックがデータ レートを超えた場合、ACI ファブリックは、パケットのドロップか、パケット内 QoS フィールドのマーキングのどちらかを実行できます。

3.2 リリース以前、同じポリサーが L2 および L3 ケースのリーフに割り当てられているとき、ポリサーの標準的な動作は EPG に適用される DPP ポリシーのケースの各 EPG メンバーになっていました。この区別は、L2/L3 ケースの DPP ポリサーがすでにインターフェイスごとになっていると想定されたため行われました。そのため、異なるのインターフェイスは、別のポリサーを取得できると想定されました。EPG あたりの DPP ポリシーが導入されましたが、特定のリーフで複数のメンバーが存在可能なことが明確になり、その後不要なドロップを避けるため、ポリサーは各メンバーごとになりました。

3.2 のリリース以降、明確なセマンティクスはデータ プレーン ポリサー ポリシー自体になり、同じように CLI に示されるように共有モード設定を導入する新しいフラグです。基本的に、データ プレーン ポリサーが L2/L3 または各 EPG に適用される場合、異なる暗黙の動作はありません。現在、ユーザーは動作の管理が可能です。共有モードが [shared] に設定されている場

合、同じデータ プレーン ポリサーを参照するリーフ上のすべてのエンティティが同じ HW ポリサーを共有します。共有モードが [dedicated] に設定されている場合、リーフ上で各 L2、L3 または EPG のメンバーに異なる HW ポリサーが割り当てられます。ポリサーは、制限する必要があるエンティティ専用です。

DPP ポリシーは、シングルレート、デュアルレート、カラー対応のいずれかになります。シングルレートポリシーは、トラフィックの認定情報レート（CIR）を監視します。デュアルレート ポリサーは、CIR と最大情報レート（PIR）の両方を監視します。また、システムは、関連するバースト サイズもモニタします。指定したデータ レート パラメータに応じて、適合（グリーン）、超過（イエロー）、違反（レッド）の 3 つのカラー、つまり条件が、パケットごとにポリサーによって決定されます。

通常、DPP ポリシーは、サーバやハイパーバイザなどの仮想または物理デバイスへの物理または仮想レイヤ 2 接続に適用されます。ルータについてはレイヤ 3 接続で適用されます。リーフ スイッチ アクセス ポートに適用された DPP ポリシーは、ACI ファブリックのファブリック アクセス（infra）部分で設定します。設定はファブリック管理者が行う必要があります。ボーダー リーフ スイッチ アクセス ポート（l3extOut または l2extOut）上のインターフェイスに適用される DPP ポリシーは、ACI ファブリックのテナント（fvTenant）部分で設定します。テナント管理者がその設定を行うことができます。

エンドポイントのグループから Cisco ACI ファブリックに入るトラフィックを EPG のメンバー アクセス インターフェイスごとに限定されるように、データ プレーン ポリサーも、EPG に適用できます。これは、1 つ EPG のさまざまな Epg でアクセス リンクを共有する場所の monopolization を防ぐために役立ちます。

各状況に設定できるアクションは 1 つだけです。たとえば、DPP ポリシーを最大 200 ミリ秒のバーストで、256,000 bps のデータ レートに適合させることが可能です。この場合、システムは、このレートの範囲内のトラフィックに対して適合アクションを適用し、このレートを超えるトラフィックに対して違反アクションを適用します。カラー対応ポリシーは、トラフィックが以前にカラーによってすでにマーキングされているものと見なします。次に、このタイプのポリサーが実行するアクションの中で、その情報が使用されます。

（注） 次は EPG ポリシングの制限事項と考慮事項です。

- 機能サポートは、EX または FX で終わるスイッチ モデルおよびそれ以降の後続モデルから開始されます（例：N9K-C93180YC-EX）。
- EPG レベル ポリサーでは、出力トラフィック ポリシングはサポートされていません。
- ポリサー モード packet-per-second はサポートされていません。
- ポリサー タイプ 2R3C はサポートされていません。
- **内部 EPG 分離**が施行されている場合、ポリサーは EPG に適用されます。
- スケール制限は、ノードごとに 128 EPG ポリサーです。
- **調整**の統計情報およびに考慮事項には次が含まれます。
  - 許可/ドロップされたパケットを認識することは、移行に関する問題やリソースの多用を知るために重要です。
  - 統計情報は、統計情報のインフラストラクチャを使用して UI で提供されます。統計情報は、Cisco ACI ファブリックで REST API を使用した任意の統計としてエクスポートされます。
  - 統計情報は各 EPG メンバーで使用でき、データ プレーン ポリサー ポリシーが **[専用]** タイプの場合に便利です。その代わり、リーフ上で使用すると統計情報がすべてのポートの統計を反映します。

データ プレーンのレイヤ 2 の GUI を使用してのポリシングの設定

始める前に

データ プレーン ポリシング ポリシーを設定するテナント、VRF、外部ルーテッド ネットワークはすでに作成されています。

データ プレーン ポリシング ポリシーは、L2 DPP ポリシーを適用するには、ポリシー グループおよびインターフェイスのプロファイルにマッピングされたポリシーグループを追加する必要があります。

手順

ステップ 1 [Navigation]ペインで、**[FABRIC]** > **[External AccessPolicies]** をクリックします。

ステップ 2 **[Policies]** > **[Interface]** > **[Data Plane Policing]**を展開し、次のアクションを実行します。

a) [Data Plane Policing Policing] を右クリックし、 [Create a Data Plane Policing Policy] をクリックします。
b) [Create a Data Plane Policing Policy] ダイアログボックスの [Name] フィールドに、ポリシーの名前を入力します。
c) [Administrative State] フィールドで、[enabled] をクリックします。
d) [Policer Mode] の隣にある [Bit Policer] または [Packet Policer] のどちらかのボタンを選択します。
e) [Type] の隣にある [1 Rate 2 Color] または [2 Rate 3 Color] のボタンを選択します。
f) 管理者は、[Conform] と [Violate] フィールドの CoS および DSCP 値を設定できます。
g) [Sharing Mode] フィールドで、ポリサー モードを選択します。

（注） 共有ポリサー モード機能を使用すると、同じポリシング パラメータを複数のインターフェイスに同時に適用できます。

h) [Burst]、[Excessive Burst]、[Rate] フィールドの隣にあるドロップダウン矢印を選択し、[1 Rate 2 Color] ポリシー タイプの各パケット レートを設定します。

（注） [2 Rate 3 Color] ポリシー タイプでは、[Peal Rate] フィールドが追加されます。

i) [Submit] をクリックします。これは、L2 の DPP 設定を完了します。データ プレーンのポリシーが L2インターフェイスにマップするインターフェイス ポリシー グループにマッピングできます。

APIC GUI を使用したレイヤ 3 のデータ プレーン ポリシングの設定

始める前に

データ プレーン ポリシング ポリシーを設定するテナント、VRF、外部ルーテッド ネットワークはすでに作成されています。

データ プレーン ポリシング ポリシーは、インターフェイス プロファイルにマッピングされたポリシー グループおよびポリシー グループに追加され、L3 DPP ポリシーを適用する必要があります。

手順

ステップ 1 **[ナビゲーション]** ペインで、**[Tenant_name]** > **[ネットワー キング]** > **[外部ルーテッド ネットワーク]** > **[Network_name]** > **[論理ノード プロファイル]** > **[論理ノード生成]** > **[論理インターフェイス プロファイル]** をクリックして、次のアクションを実行します。

a) **[論理インターフェイス プロファイル]** を右クリックして、**[インターフェイス プロファイルの作成]** を選択します。
b) [Create Interface Profile] ダイアログボックスの [Name] フィールドに、プロファイルの名前を入力します。
c) [Ingress Data Plane Policing] の隣にある [Create Data Plane Policing Policy] を選択します。
d) [Name] フィールドにポリシーの名前を入力します。
e) [Administrative State] フィールドで、[enabled] をクリックします。
f) [Policer Mode] の隣にある [Bit Policer] または [Packet Policer] のどちらかのボタンを選択します。
g) [Type] の隣にある [1 Rate 2 Color] または [2 Rate 3 Color] のボタンを選択します。
h) 管理者は、[Conform] と [Violate] フィールドの CoS および DSCP 値を設定できます。
i) [Sharing Mode] フィールドで、ポリサー モードを選択します。

（注） 共有ポリサーモード機能を使用すると、同じポリシングパラメータを複数のインターフェイスに同時に適用できます。

j) [Burst]、[Excessive Burst]、[Rate] フィールドの隣にあるドロップダウン矢印を選択し、[1 Rate 2 Color] ポリシー タイプの各パケット レートを設定します。

（注） **[2 レート 3 色]** ポリシー タイプでは、**[ピーク レート]** フィールが追加されます。

k) [Submit] をクリックします。

ステップ 2 **[ルーテッド インターフェイス]** 表を展開して、**[パス]** フィールドでインターフェイスに移動し、ポリシーを適用して、次のアクションを実行します。

a) **[IPv4 または Ipv6 優先アドレス]** の隣にあるサブネット IP アドレスを入力します。
b) [OK] をクリックします。
c) **[SVI]** タブをクリックして展開し、**[パス]** フィールドでインターフェイスに移動し、ポリシーを適用します。
d) **[Encap]** の隣に VLAN 名を入力します。
e) **[IPv4 または Ipv6 優先アドレス]** の隣にあるサブネット IP アドレスを入力します。
f) [OK] をクリックします。
g) **[ルーティング サブインターフェイス]** タブを展開し、ルーテッドインターフェイスとして同じ設定手順を実行します。
h) [OK] をクリックします。これにより L3 の DPP 設定を完了します。

REST API を使用したデータ プレーン ポリシングの設定

ポリシング、L2 のリーフに着信したトラフィック。

```
<!--  api/node/mo/uni/.xml  -->
<infraInfra>
```

```
<qosDppPol name="infradpp5" burst="2000" rate="2000" be="400" sharingMode="shared"/>
<!--
 List of nodes. Contains leaf selectors. Each leaf selector contains list of node blocks

-->
<infraNodeP name="leaf1">
<infraLeafS name="leaf1" type="range">
<infraNodeBlk name="leaf1" from_="101" to_="101"/>
</infraLeafS>
<infraRsAccPortP tDn="uni/infra/accportprof-portselector1"/>
</infraNodeP>
<!--
 PortP contains port selectors. Each port selector contains list of ports. It
       also has association to port group policies
-->
<infraAccPortP name="portselector1">
<infraHPortS name="pselc" type="range">
<infraPortBlk name="blk" fromCard="1" toCard="1" fromPort="48" toPort="49"></infraPortBlk>
<infraRsAccBaseGrp tDn="uni/infra/funcprof/accportgrp-portSet2"/>
</infraHPortS>
</infraAccPortP>
<!--  FuncP contains access bundle group policies  -->
<infraFuncP>
<infraAccPortGrp name="portSet2">
<infraRsQosIngressDppIfPol tnQosDppPolName="infradpp5"/>
</infraAccPortGrp>
</infraFuncP>
</infraInfra>
```

ポリシング、L2 トラフィックをリーフから。

```
<!--  api/node/mo/uni/.xml  -->
<infraInfra>
<qosDppPol name="infradpp2" burst="4000" rate="4000"/>
<!--
 List of nodes. Contains leaf selectors. Each leaf selector contains list of node blocks

-->
<infraNodeP name="leaf1">
<infraLeafS name="leaf1" type="range">
<infraNodeBlk name="leaf1" from_="101" to_="101"/>
</infraLeafS>
<infraRsAccPortP tDn="uni/infra/accportprof-portselector2"/>
</infraNodeP>
<!--
 PortP contains port selectors. Each port selector contains list of ports. It
       also has association to port group policies
-->
<infraAccPortP name="portselector2">
<infraHPortS name="pselc" type="range">
<infraPortBlk name="blk" fromCard="1" toCard="1" fromPort="37" toPort="38"></infraPortBlk>
<infraRsAccBaseGrp tDn="uni/infra/funcprof/accportgrp-portSet2"/>
</infraHPortS>
</infraAccPortP>
<!--  FuncP contains access bundle group policies  -->
<infraFuncP>
<infraAccPortGrp name="portSet2">
<infraRsQosEgressDppIfPol tnQosDppPolName="infradpp2"/>
</infraAccPortGrp>
</infraFuncP>
</infraInfra>
```

ポリシング、L3 のリーフに着信したトラフィック。

```
<!--  api/node/mo/uni/.xml  -->
<fvTenant name="dppTenant">
<qosDppPol name="gmeo" burst="2000" rate="2000"/>
<l3extOut name="Outside">
<l3extInstP name="extroute"/>
<l3extLNodeP name="borderLeaf">
<l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="10.0.0.1">
<ipRouteP ip="0.0.0.0">
<ipNexthopP nhAddr="192.168.62.2"/>
</ipRouteP>
</l3extRsNodeL3OutAtt>
<l3extLIfP name="portProfile">
<l3extRsPathL3OutAtt addr="192.168.40.1/30" ifInstT="l3-port"
tDn="topology/pod-1/paths-101/pathep-[eth1/40]"/>
<l3extRsPathL3OutAtt addr="192.168.41.1/30" ifInstT="l3-port"
tDn="topology/pod-1/paths-101/pathep-[eth1/41]"/>
<l3extRsIngressQosDppPol tnQosDppPolName="gmeo"/>
</l3extLIfP>
</l3extLNodeP>
</l3extOut>
</fvTenant>
```

ポリシング、L3 トラフィックをリーフから。

```
<!--  api/node/mo/uni/.xml  -->
<fvTenant name="dppTenant">
<qosDppPol name="gmeo" burst="2000" rate="2000"/>
<l3extOut name="Outside">
<l3extInstP name="extroute"/>
<l3extLNodeP name="borderLeaf">
<l3extRsNodeL3OutAtt tDn="topology/pod-1/node-101" rtrId="10.0.0.1">
<ipRouteP ip="0.0.0.0">
<ipNexthopP nhAddr="192.168.62.2"/>
</ipRouteP>
</l3extRsNodeL3OutAtt>
<l3extLIfP name="portProfile">
<l3extRsPathL3OutAtt addr="192.168.40.1/30" ifInstT="l3-port"
tDn="topology/pod-1/paths-101/pathep-[eth1/40]"/>
<l3extRsPathL3OutAtt addr="192.168.41.1/30" ifInstT="l3-port"
tDn="topology/pod-1/paths-101/pathep-[eth1/41]"/>
<l3extRsEgressQosDppPol tnQosDppPolName="gmeo"/>
</l3extLIfP>
</l3extLNodeP>
</l3extOut>
</fvTenant>
```

NX-OS スタイル CLI を使用したデータ プレーン ポリシングの設定

手順

ステップ 1 1 つの EPG の伝送に L2 ポートを設定します。

例：

```
apic1# conf t
apic1(config)# vlan-domain test
```

```
apic1(config-vlan)# vlan 1000-2000
apic1(config-vlan)# exit
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/10
apic1(config-leaf-if)# vlan-domain member test
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
apic1(config)# tenant test1
apic1(config-tenant)# vrf context v1
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# bridge-domain bd1
apic1(config-tenant-bd)# vrf member v1
apic1(config-tenant-bd)# exit
apic1(config-tenant)# application ap1
apic1(config-tenant-app)# epg e1
apic1(config-tenant-app-epg)# bridge-domain member bd1
apic1(config-tenant-app-epg)# exit
apic1(config-tenant-app)# exit
apic1(config-tenant)# exit
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/10
apic1(config-leaf-if)# switchport trunk allowed vlan 1001 tenant test1 application ap1
epg e1
apic1(config-leaf-if)# switchport trunk allowed vlan 1501 tenant test1 application ap1
epg e1
# Now the port leaf 101 ethernet 1/10 carries two vlan mapped both to the same
Tenant/Application/EPG
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
```

a) インターフェイスに適用するポリシー マップを作成します。

例：

```
apic1(config)# policy-map type data-plane qosTest
apic1(config-pmap-dpp)# set burst 2400 mega
apic1(config-pmap-dpp)# set cir 70 mega

apic1(config-pmap-dpp)# set sharing-mode dedicated
apic1(config-pmap-dpp)# exit
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/10
apic1(config-leaf-if)# service-policy type data-plane input qosTest
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
apic1(config)# policy-map type data-plane qosTest2
apic1(config-pmap-dpp)# set cir 78 mega
apic1(config-pmap-dpp)# exit
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/10
apic1(config-leaf-if)# service-policy type data-plane output qosTest2
apic1(config-leaf-if)# end
```

b) 設定されたポリシーを可視化します。

例：

```
apic1# show policy-map type data-plane infra
Type data-plane policy-maps
====================
Global Policy
policy-map type data-plane default
    set burst unspecified
```

```
    set conform-cos-transmit unspecified
    set conform-dscp-transmit unspecified
    set conform transmit
    set excessive-burst unspecified
    set exceed-cos-transmit unspecified
    set exceed-dscp-transmit unspecified
    set exceed drop
    set mode byte
    set pir 0
    set cir 78 mega
    set type 1R2C
    set violate-cos-transmit unspecified
    set violate-dscp-transmit unspecified
    set violate drop
Global Policy
policy-map type data-plane qosTest
    set burst 2400 mega
    set cir 78 mega
    set conform-cos-transmit unspecified
    set conform-dscp-transmit unspecified
    set conform transmit
    set excessive-burst unspecified
    set exceed-cos-transmit unspecified
    set exceed-dscp-transmit unspecified
    set exceed drop
    set mode byte
    set pir 0
    set type 1R2C
    set violate-cos-transmit unspecified
    set violate-dscp-transmit unspecified
    set violate drop
Global Policy
policy-map type data-plane qosTest2
    set burst unspecified
    set conform-cos-transmit unspecified
    set conform-dscp-transmit unspecified
    set conform transmit
    set excessive-burst unspecified
    set exceed-cos-transmit unspecified
    set exceed-dscp-transmit unspecified
    set exceed drop
    set mode byte
    set pir 0
    set cir 78 mega
    set type 1R2C
    set violate-cos-transmit unspecified
    set violate-dscp-transmit unspecified
    set violate drop
```

c) show running-config.

例：

```
apic1# show runn policy-map
# Command: show running-config policy-map
# Time: Fri Jan 29 19:26:18 2016
  policy-map type data-plane default
    exit
  policy-map type data-plane qosTest
    set burst 2400 mega
    set cir 78 mega
    no shutdown
    exit
  policy-map type data-plane qosTest2
    set cir 78 mega
```

```
    no shutdown
    exit
apic1# show runn leaf 101
# Command: show running-config leaf 101
# Time: Fri Jan 29 19:26:29 2016
  leaf 101
    interface ethernet 1/10
      vlan-domain member test
      switchport trunk allowed vlan 1501 tenant test1 application ap1 epg e1
      service-policy type data-plane input qosTest
      service-policy type data-plane output qosTest2
      exit
    exit
```

ステップ 2 L3 ポートを設定する準備します。

例：

```
apic1#
apic1# conf t
apic1(config)# vlan-domain l3ports
apic1(config-vlan)# vlan 3000-3001
apic1(config-vlan)# exit
apic1(config)# tenant l3test1
apic1(config-tenant)# vrf context v1
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# exit
apic1(config)# leaf 102
apic1(config-leaf)# vrf context tenant l3test1 vrf v1
apic1(config-leaf-vrf)# exit
# Configure a physical L3 port
apic1(config-leaf)# interface ethernet 1/20
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# vlan-domain member l3ports
apic1(config-leaf-if)# vrf member tenant l3test1 vrf v1
apic1(config-leaf-if)# ip address 56.1.1.1/24
apic1(config-leaf-if)# ipv6 address 2000::1/64  preferred
apic1(config-leaf-if)# exit
# Configure base interface for L3 subinterfaces
apic1(config-leaf)# interface ethernet 1/21
apic1(config-leaf-if)# vlan-domain member l3ports
apic1(config-leaf-if)# no switchport
apic1(config-leaf-if)# exit
# Configure a L3 subinterface
apic1(config-leaf)# interface ethernet 1/21.3001
apic1(config-leaf-if)# vrf member tenant l3test1 vrf v1
apic1(config-leaf-if)# ip address 60.1.1.1/24
apic1(config-leaf-if)# ipv6 address 2001::1/64 preferred
apic1(config-leaf-if)# exit
# Configure a Switched Vlan Interface
apic1(config-leaf)# interface vlan 3000
apic1(config-leaf-if)# vrf member tenant l3test1 vrf v1
apic1(config-leaf-if)# ip address 70.1.1.1/24
apic1(config-leaf-if)# ipv6 address 3000::1/64 preferred
apic1(config-leaf-if)# exit
apic1(config-leaf)#exit
```

a) L3 使用率のテナントで、ポリサーを設定します。

例：

```
apic1(config)# tenant l3test1
apic1(config-tenant)# policy-map type data-plane iPol
```

```
apic1(config-tenant-pmap-dpp)# set cir 56 mega
apic1(config-tenant-pmap-dpp)# set burst 2000 kilo
apic1(config-tenant-pmap-dpp)# exit
apic1(config-tenant)# policy-map type data-plane ePol
apic1(config-tenant-pmap-dpp)# set burst 2000 kilo
apic1(config-tenant-pmap-dpp)# set cir 56 mega
apic1(config-tenant-pmap-dpp)# exit
apic1(config-tenant)# exit
```

b) L3 インターフェイスでポリサーを適用します。

例：

```
apic1(config)# leaf 102
apic1(config-leaf)# interface ethernet 1/20
apic1(config-leaf-if)# service-policy type data-plane input iPol
apic1(config-leaf-if)# service-policy type data-plane output ePol
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface ethernet 1/21.3001
apic1(config-leaf-if)# service-policy type data-plane input iPol
apic1(config-leaf-if)# service-policy type data-plane output ePol
apic1(config-leaf-if)# exit
apic1(config-leaf)# interface vlan 3000
apic1(config-leaf-if)# service-policy type data-plane input iPol
apic1(config-leaf-if)# service-policy type data-plane output ePol
apic1(config-leaf-if)# end
```

c) L3 インターフェイスで使用されているポリサーのコマンドを表示します。

例：

```
apic1# show tenant l3test1 policy-map type data-plane
Type data-plane policy-maps
====================
Policy in Tenant: l3test1
policy-map type data-plane ePol
    set burst 2000 kilo
    set conform-cos-transmit unspecified
    set conform-dscp-transmit unspecified
    set conform transmit
    set excessive-burst unspecified
    set exceed-cos-transmit unspecified
    set exceed-dscp-transmit unspecified
    set exceed drop
    set mode byte
    set pir 0
    set cir 56 mega
    set type 1R2C
    set violate-cos-transmit unspecified
    set violate-dscp-transmit unspecified
    set violate drop
Policy in Tenant: l3test1
policy-map type data-plane iPol
    set burst 2000 kilo
    set burst unspecified
    set conform-cos-transmit unspecified
    set conform-dscp-transmit unspecified
    set conform transmit
    set excessive-burst unspecified
    set exceed-cos-transmit unspecified
    set exceed-dscp-transmit unspecified
    set exceed drop
    set mode byte
    set pir 0
```

```
    set cir 56 mega
    set type 1R2C
    set violate-cos-transmit unspecified
    set violate-dscp-transmit unspecified
    set violate drop
```

d) L3 に使用されるポリサーの show running-config です。

例：

```
apic1# show runn tenant l3test1
# Command: show running-config tenant l3test1
# Time: Fri Jan 29 19:48:20 2016
  tenant l3test1
    vrf context v1
      exit
    policy-map type data-plane ePol
      set burst 2000 kilo
      set cir 56 mega
      no shutdown
      exit
    policy-map type data-plane iPol
      set burst 2000 kilo
      set cir 56 mega
      no shutdown
      exit
    exit
apic1# show running-config leaf 102
# Command: show running-config leaf 102
# Time: Fri Jan 29 19:48:33 2016
  leaf 102
    vrf context tenant l3test1 vrf v1
      exit
    interface vlan 3000
      vrf member tenant l3test1 vrf v1
      ip address 70.1.1.1/24
      ipv6 address 3000::1/64 preferred
      bfd ip tenant mode
      bfd ipv6 tenant mode
      service-policy type data-plane input iPol
      service-policy type data-plane output ePol
      exit
    interface ethernet 1/20
      vlan-domain member l3ports
      no switchport
      vrf member tenant l3test1 vrf v1
      ip address 56.1.1.1/24
      ipv6 address 2000::1/64 preferred
      bfd ip tenant mode
      bfd ipv6 tenant mode
      service-policy type data-plane input iPol
      service-policy type data-plane output ePol
      exit
    interface ethernet 1/21
      vlan-domain member l3ports
      no switchport
      bfd ip tenant mode
      bfd ipv6 tenant mode
      exit
    interface ethernet 1/21.3001
      vrf member tenant l3test1 vrf v1
      ip address 60.1.1.1/24
      ipv6 address 2001::1/64 preferred
      bfd ip tenant mode
      bfd ipv6 tenant mode
```

```
      service-policy type data-plane input iPol
      service-policy type data-plane output ePol
    exit
    exit
apic1#
```

エンドポイントのグループ レベルでのデータ プレーン ポリシング

データ プレーン ポリシング（DPP）は、エンドポイント グループ（EPG）に適用できます。トラフィックのポリシングは、EPG が展開されているすべてのリーフ スイッチ上のすべての EPG メンバに適用されます。

3.2 までは各 EPG メンバが独自のポリサーを取得しており、3.2 リリースよりデータ プレーン ポリサーの共有モード プロパティにより動作が異なります（CLI で設定されている場合）。**[専用]** に設定されている場合、3.2 以前の状況に似ています。共有モードが **[共有]** に設定されており、すべてのメンバが同じデータ プレーン ポリサー ポリシーを使用している場合、リーフで HW ポリサーを使用します。

たとえば、EPG には次のメンバがあります。

- Leaf 101,Eth1/1,vlan-300
- Leaf 101,Eth1/2,vlan-301
- Leaf 102,Eth1/2,vlan-500

この状況では、データ プレーン ポリサーが共有モードを **[専用]** に設定している場合（デフォルト）、各メンバがポリサーに従いトラフィックを制限し、他のメンバから独立します。ただし、データ プレーン ポリサーが共有モードを **[共有]** に設定している場合、上記すべてのメンバはリーフ上で1つのポリサーのみを使用します。弊害は、大量のトラフィックを取得するメンバ 1 つが他のメンバをスレーブ化する可能性があります。

EPG の DPP は、L2/L3 ポリサーのように両方向ではなく、ファブリックに入るトラフィッ クのみポリシングします。

（注） 以下は、EPG レベルでのデータ プレーン ポリシングの制限です。

- EPGポリサー機能のサポートは、名前がEX/FXで終わるスイッチモデルよりも新しいモデルから開始します。

- 出力トラフィック ポリシングでは EPG レベル ポリサーはサポートされていません。
- ポリサー モード **Packet-per-second** はサポートされていません。
- ポリサー タイプ 2R3C はEPG ポリサーではサポートされていません。
- **intra-EPG isolation-enforced** が EPG に適用されている場合、ポリサーはサポートされません。
- スケール制限では、ノードごとに 128 EPG ポリサーがサポートできます。

CLI を使用したエンドポイント グループ レベルでのデータ プレーン ポリシングの設定

手順

ポリサーの定義：

例：

```
apic1# conf t
apic1(config)# vlan-domain test
apic1(config-vlan)# vlan 1000-2000
apic1(config-vlan)# exit
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/10
apic1(config-leaf-if)# vlan-domain member test
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
apic1(config)# tenant test1
apic1(config-tenant)# vrf context v1
apic1(config-tenant-vrf)# exit
apic1(config-tenant)# bridge-domain bd1
apic1(config-tenant-bd)# vrf member v1
apic1(config-tenant-bd)# exit
apic1(config)# policy-map type data-plane pol1
apic1(config-pmap-dpp)# set burst 2400 mega
apic1(config-pmap-dpp)# set cir 78 mega
apic1(config-pmap-dpp)# exit
apic1(config-tenant)# application ap1
apic1(config-tenant-app)# epg e1
apic1(config-tenant-app-epg)# bridge-domain member bd1
apic1(config-tenant-app-epg)# service-policy type data-plane pol1
apic1(config-tenant-app-epg)# exit
apic1(config-tenant-app)# exit
apic1(config-tenant)# exit
apic1(config)# leaf 101
apic1(config-leaf)# interface ethernet 1/10
apic1(config-leaf-if)# switchport trunk allowed vlan 1001 tenant test1 application ap1
epg e1
```

```
apic1(config-leaf-if)# exit
apic1(config-leaf)# exit
```

データ プレーン APIC GUI を使用してエンドポイント グループ レベルでのポリシングの設定

手順

[Tenants] ペインで、**[Tenant_name]** > **[Policies]** > **[Protocol]** > **[Data Plane Policing]** をクリックします。[Data Plane Policing] を右クリックし、 [Create Data Plane Policing Policy] をクリックします。

a) [Name] フィールドにポリシーの名前を入力します。
b) [Administrative State] フィールドで、[enabled] をクリックします。
c) [Policer Mode] の隣にある [Bit Policer] または [Packet Policer] のどちらかのボタンを選択します。
d) [Type] の隣にある [1 Rate 2 Color] または [2 Rate 3 Color] のボタンを選択します。
e) [Conform Action] で 、[Drop]、 [Mark]、または [Transmit] を選択します 。
f) 管理者は、[Conform] と [Violate] フィールドの CoS および DSCP 値を設定できます。
g) [Burst]、[Excessive Burst]、[Rate] フィールドの隣にあるドロップダウン矢印をクリックして、次のいずれかを選択します。
   - バイト/パケット
   - キロ バイト/パケット
   - メガ バイト/パケット
   - ギガ バイト/パケット
   - ミリ秒
   - マイクロ秒

データ プレーンの Rest API を使用したエンドポイント グループ レベルでのポリシングの設定

リーフ スイッチに着信するトラフィックを規制します。

```
<!-- api/node/mo/.xml -->
<polUni>
  <fvTenant  name="t1">
```

```
    <qosDppPol name="gmeo" burst="2000" rate="2000"/>
    <fvAp name="ap1">
      <fvAEPg name="ep1">
        <fvRsDppPol tnQosDppPolName="gmeo"/>
      </fvAEPg>
    </fvAp>
  </fvTenant>
</polUni>
```

GUI のエンドポイント グループ レベルでデータ プレーン ポリサーの統計情報へのアクセス

EPG レベルの DPP は、EPG メンバ レベルのトラフィックを規制するために使用されます。その結果、統計情報はポリサーが存在するトラフィックをドロップすることを保証する整数です。統計情報は、EPG メンバ レベルで詳細に報告されます。

手順

ステップ 1 **[テナント]** ペインで、**[Tenant_name]** > **[アプリケーション EPG]** > **[EPG メンバ]** > **[スタティック EPG メンバ]** をクリックします。

ステップ 2 ノードを選択します。

ステップ 3 **[統計情報の選択]** をクリックします。

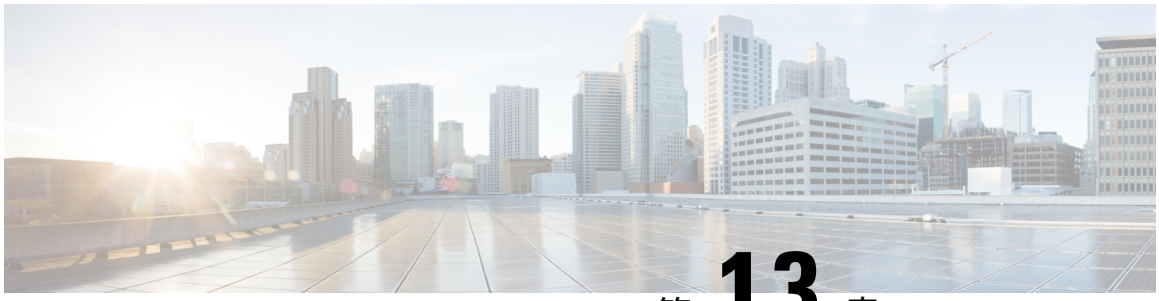
a) **[サンプリング間隔]** 時間単位を選択します。

b) **[利用可能]** ポリサー属性から、矢印を使用して属性を選択します。最大 2 種類の属性を選択できます。

c) [Submit] をクリックします。

次のタスク

DPP 統計情報がグラフィカル表示されます。



第 13 章

HTTPS アクセス

この章の内容は、次のとおりです。

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

- ワイルドカード証明書（*.cisco.com など。複数のデバイス間で使用）およびそれに関連する他の場所で生成される秘密キーは、APIC ではサポートされません。これは、APIC に秘密キーまたはパスワードを入力するためのサポートがないためです。また、ワイルドカード証明書などのいかなる証明書の秘密キーもエクスポートできません。
- 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。APIC は、送信された証明書が設定されている 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。

- APIC で公開キーと秘密キーを再利用する場合は、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関して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。

- 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。
- ポッドあたり 1 つの証明書ベースのルートのみをアクティブにすることができます。
- 任意のリリースからリリース 4.0(1) へのダウングレードを実行する場合は、事前に証明書ベースの認証を無効にしておく必要があります。
- 証明書ベースの認証セッションを終了するには、ユーザはログアウトして CAC カードを削除する必要があります。

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 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、APIC とスイッチの接続には影響しません。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。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。この操作中にファブリック内のすべての Web サーバの再起動が予期されます。

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。

手順

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 [Work] ペインで、**[Public Key Management]** > **[Certificate Authorities]** > **[Create Certificate Authority]** を選択します。

ステップ 4 [Create Certificate Authority] ダイアログボックスの [Name] フィールドに、認証局の名前を入力します。

ステップ 5 [Certificate Chain] フィールドに、Application Policy Infrastructure Controller（APIC）の証明書署名要求（CSR）に署名する認証局の中間証明書および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 6 [Submit] をクリックします。

ステップ 7 [Navigation] ペインで、**[Public Key Management]** > **[Key Rings]** の順に選択します。

ステップ 8 [Work] ペインで、**[Actions]** > **[Create Key Ring]** の順に選択します。

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 [Certificate] フィールドには、コンテンツを追加しないでください。

ステップ 11 [Modulus] フィールドで、目的のキー強度のラジオボタンをクリックします。

ステップ 12 [Certificate Authority] フィールドのドロップダウン リストから、前に作成した認証局を選択します。[Submit] をクリックします。

（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。

[Work] ペインの [Key Rings] 領域では、作成したキー リングに対する [Admin State] に [Started] と表示されます。

ステップ 13 [Navigation] ペインで、**[Public Key Management]** > **[Key Rings]** > ***[key_ring_name]*** の順に選択します。

ステップ 14 [Work] ペインで、**[Actions]** > **[Create Certificate Request]** の順に選択します。

ステップ 15 [Subject] フィールドに、APIC の完全修飾ドメイン名（FQDN）を入力します。

ステップ 16 必要に応じて、残りのフィールドに入力します。

（注） 使用可能なパラメータの説明については、[Create Certificate Request] ダイアログボックスでオンライン ヘルプ情報を確認してください。

ステップ 17 [Submit] をクリックします。
[Navigation] ペインでは、前に作成したキー リングの下にオブジェクトが作成され、表示されます。[Navigation] ペインでそのオブジェクトをクリックすると、[Work] ペインの [Properties] 領域の [Request] フィールドにその CSR が表示されます。認証局に送信するコンテンツをフィールドからコピーします。

ステップ 18 [Navigation] ペインで、**[Public Key Management]** > **[Key Rings]** > ***[key_ring_name]*** の順に選択します。

ステップ 19 [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 20 [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 21 メニュー バーで、**[Fabric]** > **[Fabric Policies]** の順に選択します。

ステップ 22 [Navigation] ペインで、**[Pod Policies]** > **[Policies]** > **[Management Access]** > **[default]** の順に選択します。

ステップ 23 [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 24 （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 25 [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、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

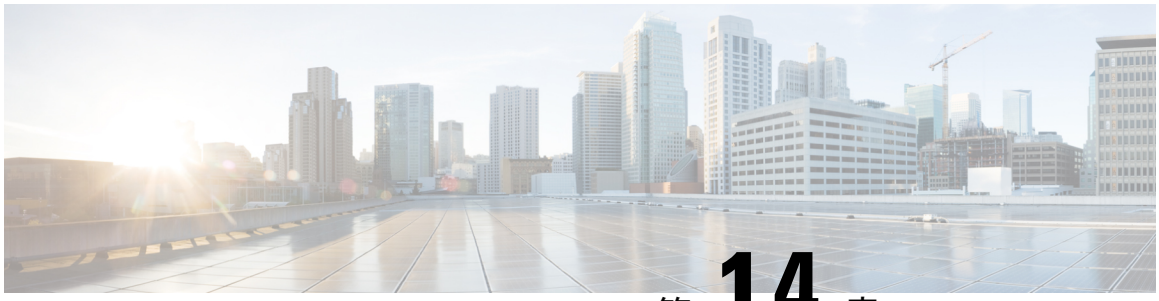
手順

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
```



第 14 章

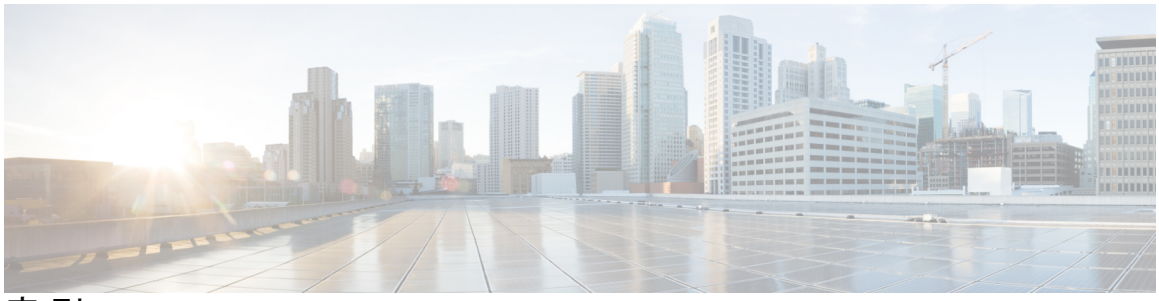
その他の ACI セキュリティ機能

このページには、機能の一覧を示します。

その他のセキュリティ機能

現在 ACI でサポートされているその他のセキュリティ機能は次のリストのとおりです。それぞれの詳細については、他の構成ガイドで説明されています（https://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html）。

- **コントラクト**の設定については、『*Cisco APIC Basic Configuration Guide, Release 3.x*（Cisco APIC 基本設定ガイド リリース 3.x）』および『*Operating Cisco Application Centric Infrastructure*（シスコ アプリケーション セントリック インフラストラクチャの運用）』を参照してください。
- **EPG 通信ルール**については、ナレッジベースの記事『*Use vzAny to Automatically Apply Communication Rules to all EPGs in a VRF*（vzAny を使用して通信ルールを VRF 内のすべての EPG に自動的に適用する ）』を参照してください。
- **インバンドおよびアウトオブバンドの管理アクセス**については、ナレッジベースの記事『*Cisco APIC and Static Management Access*（Cisco APIC と静的管理アクセス）』および『*Cisco APIC Layer 4 to Layer 7 Services Deployment Guide, Release 2.2(3)*（Cisco APIC レイヤ 4 ～ レイヤ 7 サービス導入ガイド、リリース 2.2(3)）』を参照してください。
- **EPG 内での分離適用**については、『*Cisco ACI Virtualization Guide, Release 3.0(1)*（Cisco ACI 仮想化ガイド、リリース 3.0 (1)）』を参照してください。
- **トラフィック ストーム制御**については、『*Cisco APIC Layer 2 Networking Configuration Guide*（Cisco APIC レイヤ 2 ネットワーキング設定ガイド）』を参照してください。



索 引