セキュリティ ポリシー

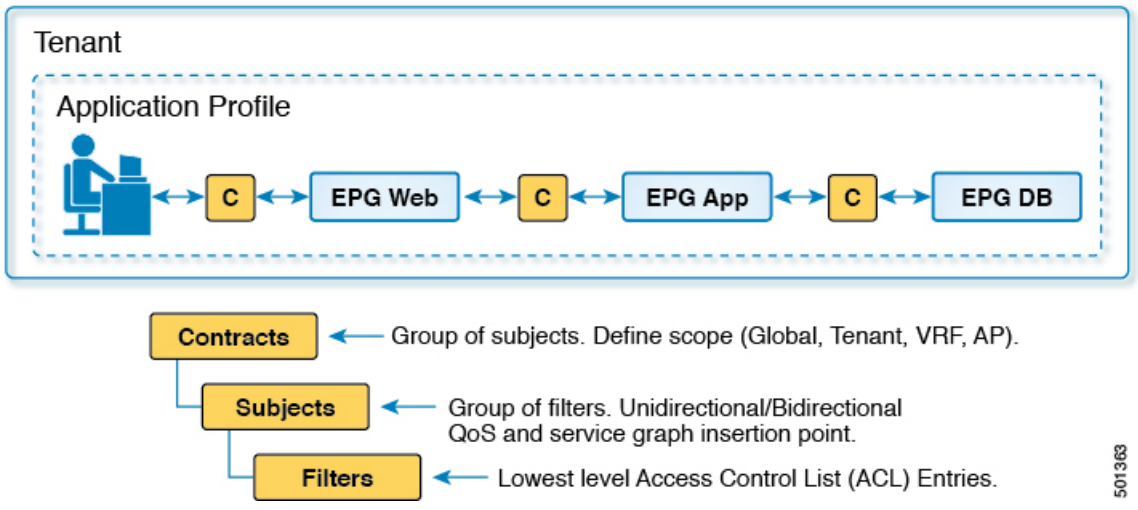
この章は、次の項で構成されています。

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 *1*: 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。

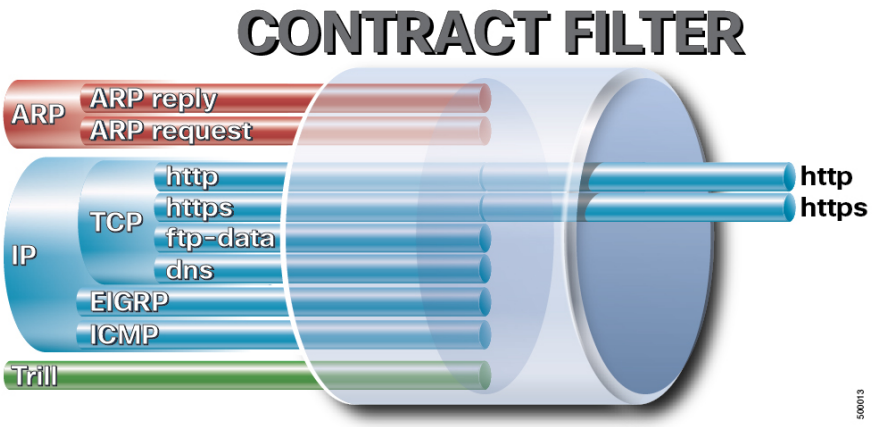
また EPG とコントラクトの関係には方向性があります。EPG はコントラクトを提供または消費できます。コントラクトを提供する EPG は通常、一連のクライアント デバイスにサービスを提供する一連のエンドポイントです。そのサービスによって使用されるプロトコルはコントラクトで定義されます。コントラクトを消費する EPG は通常、そのサービスのクライアントである一連のエンドポイントです。クライアント エンドポイント (コンシューマ) がサーバ エンドポイント (プロバイダー) に接続しようとすると、コントラクトはその接続が許可されるかどうかを確認します。特に指定のない限り、そのコントラクトは、サー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

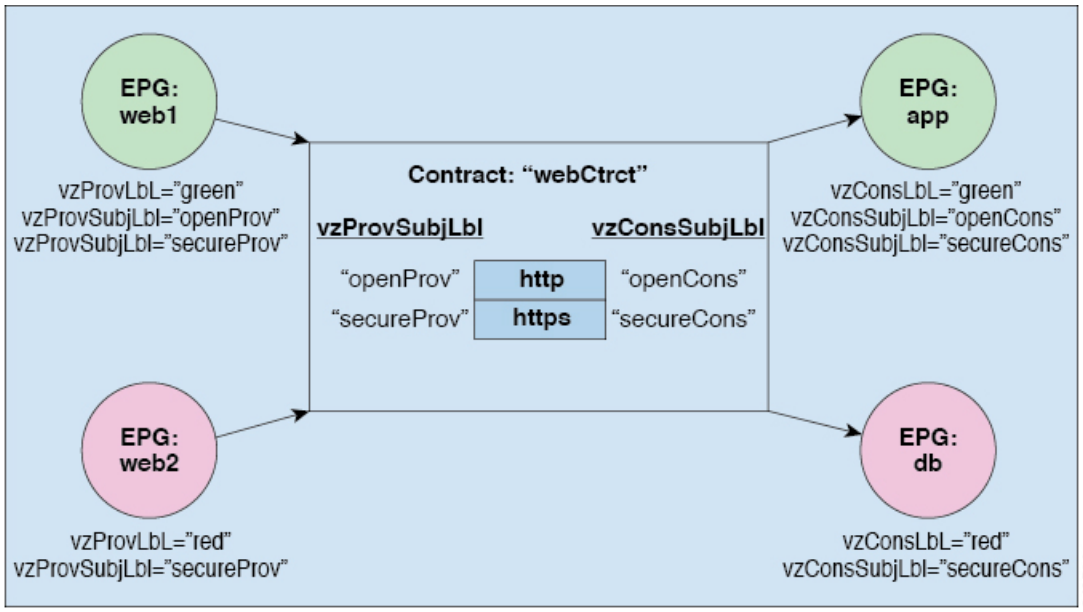
コントラクトは階層的に構築されます。1 つ以上のサブジェクトで構成され、各サブジェクトには 1 つ以上のフィルタが含まれ、各フィルタは 1 つ以上のプロトコルを定義できます。

図 *2:* 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 *3: 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 セットの情報カテゴリを持つ webCtrct と呼ばれるコントラクトを作成できます。openProv と openCons are が HTTP フィルタが含まれる情報カテゴリです。secureProv と secureCons は HTTPS フィルタが含まれる情報カテゴリです。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。

フィルタ エントリの設定

このセクションでは、次のフィルタエントリ構成オプションについて説明します。

- 部分的にのみ一致
- 一致の DSCP
- TCP フラグ
- ステートフル

- ポートゼロエントリ

各フィルタには 1 つ以上のフィルタエントリを含めることができます。フィルタの場所は **[テナント（Tenant）] > [コントラクト（Contract）] > [フィルタ（Filters）] > [Filter_name]** です。フィルタエントリごとの構成の場所は **[テナント（Tenant）] > [コントラクト（Contract）] > [フィルタ（Filters）] > [Filter_name] > [Filter_entry_name]** です。

部分的にのみ一致

[フラグメントにのみマッチング（Match Only Fragments）] オプションは、オフセットが 0 より大きいフラグメント（最初のフラグメントを除くすべてのフラグメント）を照合します。

[フラグメントにのみマッチング（Match Only Fragments）] オプションは、デフォルトでは無効になっています。これは、デフォルトではフィルタ構成がすべてのパケット（すべてのフラグメントを含む）に適用されることを意味します。したがって、デフォルトでは、フィルタにマッチするすべてのパケットを、コントラクトアクションに基づいて許可、ドロップ、コピー、リダイレクトすることができます。[フラグメントにのみマッチング（Match Only Fragments）] オプションが有効になっている場合、フィルタ構成は最初のフラグメントを除くすべてのフラグメントに適用されます。

（注） TCP/UDP ポート情報は、最初のフラグメントでのみチェックできます。

次にいくつか例を示します。

- 許可コントラクトの IP フィルタで [フラグメントにのみマッチング（Match Only Fragments）] が無効になっている場合（デフォルト）、すべてのフラグメントを含むすべての IP パケットが許可されます。
- 許可コントラクトに [フラグメントにのみマッチング（Match Only Fragments）] が有効になっている IP フィルタがある場合、オフセットが 0 より大きい IP フラグメント（最初のフラグメントを除くすべての IP フラグメント）のみが許可されます。そのようなわけで、別の許可コントラクトがない限り、最初のフラグメントは暗黙の拒否ルールによってドロップされます。
- 許可コントラクトに特定の TCP ポートフィルタ（接続先 TCP ポート 80 など）があり、その許可コントラクトで [フラグメントにのみマッチング（Match Only Fragments）] が無効になっている場合（デフォルト）、特定の TCP ポートにマッチするすべての TCP トラフィックが許可されます。TCPポート情報が最初のフラグメントのみにあるため、別の許可コントラクトがない限り、最初のフラグメント以外のフラグメントは暗黙の拒否ルールによってドロップされます。
- 特定の TCP/UDP ポートフィルタで [フラグメントにのみマッチング（Match Only Fragments）] を有効にするのは、有効な構成の組み合わせではありません。TCP/UDP ポート情報は最初のフラグメントでしかチェックできないのに対し、[フラグメントにのみマッチング（Match Only Fragments）] は、最初のものを除いた、すべてのフラグメントとマッチングを行うようにとの指定だからです。

一致の DSCP

このオプションは、EtherType、IP プロトコル、送信元ポート、宛先ポートに加えて、トラフィックで照合する DSCP（差別化サービスコードポイント、Differentiated Services Code Point）値を指定します。このオプションを使用すると、送信元 EPG、接続先 EPG、フィルタマッチングなどの他のパラメータが同じであっても、パケット内の DSCP 値に応じて異なるアクションを実行できます。このオプションは、デフォルトでは未指定です（Cisco ACI における従来の IOS または NX-OS 用語では、[任意（Any）] に相当します）。これには、「EX」または「FX」以降のリーフノードが必要です。

TCP フラグ

このオプションは、EtherType、IP プロトコル、送信元ポート、宛先ポートに加えて、トラフィックで照合する TCP フラグ値を指定します。使用可能な TCP フラグは次のとおりです。

- Synchronize：SYN
- Established: ACK または RST
- Acknowledgement: ACK
- Reset: RST
- Finish: FIN

ステートフル

[ステートフル（Stateful）] オプションは、ACK フラグが設定されている場合にのみ、プロバイダーからコンシューマへの TCP パケットを許可します。このオプションは、デフォルトで無効です。セキュリティ向上のため、TCP フィルタエントリではステートフルオプションを有効にすることをお勧めします。ただし、[ポリシー圧縮の有効化（Enable Policy Compression）] が必要な場合を除きます。ステートフルオプションを有効にすると、ポリシー圧縮を適用できないからです。

コンシューマが特定のプロバイダーの TCP ポートにアクセスできるようにするには、管理者はコンシューマ側の TCP ポート（コントラクトフィルタの送信元ポート構成）をワイド範囲として構成する必要があります。ウェルノウンポートではない送信元ポートも対応可能にするためです。次の例には、2 つのゾーニングルールがあります。1 つは任意の送信元 TCP ポートを使用するコンシューマから接続先 TCP ポート 80 を使用するプロバイダーへのトラフィックを許可するルール、もう 1 つは逆方向のルールです。プロバイダーのエンドポイントが送信元 TCP ポート80を使用してコンシューマのエンドポイントに SYN 攻撃を実行した場合でも、トラフィックが ACI ファブリックによって自動的にドロップされることはありません。送信元 TCP ポート 80 を使用するプロバイダーから、任意の宛先 TCP ポートを使用するコンシューマへのトラフィックは、コントラクトによって許可されているからです。

プロバイダーからコンシューマへの通常の TCP パケットが許可される場合、次のようになります。

- データパケット（3 ウェイハンドシェイク後）：これらのパケットには ACK ビットが設定されているため、リーフノードはパケットを許可します。

- RST パケット：RST パケットにも ACK ビットが設定されているため、リーフノードは RST パケットを許可します。
- FIN パケット：ACK ビットが設定された FIN パケットは許可されます。ACK のない FIN パケットはドロップされます。ACK のない FIN パケットの処理は、オペレーティングシステムのタイプによって異なるため、オペレーティングシステムを特定する目的での FIN スキャン攻撃が行われることがあります。このようなパケットをドロップすれば、攻撃を防ぐことができます。

「show zoning-rule」コマンドの CLI 出力は、ステートフルオプションが有効になっているリーフでプログラムされたポリシーの例です。

```
Pod1-Leaf1# show zoning-rule scope 2850817
+---------+--------+--------+----------+---------+---------+---------+------------------+
| Rule ID | SrcEPG | DstEPG | FilterID |   Dir   |  operSt |  Scope  |        Name
  |  Action  |       Priority       |
+---------+--------+--------+----------+---------+---------+---------+------------------+
|   4250  |   0    |   0    | implicit | uni-dir | enabled | 2850817 |
  | deny,log |   any_any_any(21)    |
|   4246  |   0    |   0    | implarp  | uni-dir | enabled | 2850817 |
  |  permit  |  any_any_filter(17)  |
|   4208  |   0    |   15   | implicit | uni-dir | enabled | 2850817 |
  | deny,log | any_vrf_any_deny(22) |
|   4247  |   0    | 32777  | implicit | uni-dir | enabled | 2850817 |
  |  permit  |   any_dest_any(16)   |
|   4222  | 32774  | 32775  |    71    | uni-dir | enabled | 2850817 | tenant1:Contract1
 |  permit  |    fully_qual(7)     |
|   4244  | 32775  | 32774  |    69    | uni-dir | enabled | 2850817 | tenant1:Contract1
 |  permit  |    fully_qual(7)     |
+---------+--------+--------+----------+---------+---------+---------+------------------+
```

これらの行は、EPG Web と EPG アプリ間の Contract1 によって作成されたものです。フィルタエントリ情報の詳細は、「show zoning-filter filter **FilterID**」コマンドを使用して確認できます。プロバイダーからコンシューマへの方向で使用されるフィルタ ID 71 には、TcpRules の「ack」があります。

```
Pod1-Leaf1# show zoning-filter filter 69
+----------+------+--------+-------------+------+-------------+----------+-------------+
| FilterId | Name | EtherT |    ArpOpc   | Prot | ApplyToFrag | Stateful |  SFromPort
|   SToPort    | DFromPort | DToPort |  Prio |    Icmpv4T    |    Icmpv6T    | TcpRules |
+----------+------+--------+-------------+------+-------------+----------+-------------+
|    69    | 69_0 |   ip   | unspecified | tcp  |      no     |   yes    | unspecified
| unspecified |     22    |     22   | dport | unspecified | unspecified |          |
+----------+------+--------+-------------+------+-------------+----------+-------------+

Pod1-Leaf1# show zoning-filter filter 71
+----------+------+--------+-------------+------+-------------+----------+-------------+
| FilterId | Name | EtherT |    ArpOpc   | Prot | ApplyToFrag | Stateful | SFromPort |
SToPort |  DFromPort  |   DToPort   |  Prio |    Icmpv4T    |    Icmpv6T    | TcpRules |
+----------+------+--------+-------------+------+-------------+----------+-------------+
|    71    | 71_0 |   ip   | unspecified | tcp  |      no     |   yes    |     22    |
   22   | unspecified | unspecified | flags | unspecified | unspecified |   ack    |
+----------+------+--------+-------------+------+-------------+----------+-------------+
```

次のリストは、ステートフルオプションの使用に関連する設計上の重要な考慮事項の一部をまとめたものです。

- ステートフルオプションは、TCP トラフィックにのみ適用されます。
- ステートフルオプションは ACK フラグしかチェックしないので、ステートフルファイアウォールとは異なり、プロバイダーからの SYN + ACK 攻撃は防止できません。
- ステートフルが有効になっている場合、双方向ルール圧縮 は適用できません。

ポートゼロエントリ

各フィルタには、1 つ以上のフィルタエントリを含めることができます。これは **[テナント（Tenant）] > [コントラクト（Contract）] > [フィルタ（Filters）] > [Filter_name]** にあります。

APIC リリース6.0(4)以降、ポートゼロエントリが導入されました。一般的なフィルタエントリとポートゼロエントリの違いは次のとおりです。

- 一般のフィルタエントリでポートが「未指定」または「0」に設定されている場合、ポート範囲は「0 ～ 65535」です。
- ポートゼロエントリは、ポート「0」のフィルタエントリ用です。これは、ポート「0」がインターネット番号割当機関（Internet Assigned Numbers Authority、IANA）によって予約済みポートとして定義されており、使用が想定されていないため、主にこのようなトラフィックを拒否するためのものです。

ポートゼロエントリには、次の [方向（Direction）] オプションがあります。

- [両方向（Direction Both、デフォルト）]：送信元ポート「0」も宛先ポートも「0」。
- [宛先方向（Direction Destination）]：送信元ポートは「0」、宛先ポートは「すべて」（0 ～ 65535）。
- [送信元方向（Direction Source）]：送信元ポートは「すべて」（0 ～ 65535）、宛先ポートは「0」。

（注） 送信元ポート「0」と宛先ポート「80」のフィルタなど、送信元または宛先ポートのいずれかが「0」で、他方を特定のポートに指定しているフィルタエントリは、汎用フィルタエントリまたはポートゼロエントリではサポートされません。

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
- ACI ファブリックの ACL コントラクト許可は、EX または FX で終わる名前の Nexus 9000 シリーズ スイッチ、およびそれ以降のすべてのモデルでのみサポートされます。たとえば、N9K-C93180LC-EX や N9K-C9336C-FX のように指定してください。
- ACI ファブリックでのログの拒否は、すべてのプラットフォームでサポートされています。
- 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 **[送信（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 **[送信（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。

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。