

サービス ポリシー

モジュラ ポリシー フレームワークを使用したサービス ポリシーにより、一貫性のある柔軟な方法で ASA の機能を設定できます。たとえば、サービス ポリシーを使用すると、すべての TCP アプリケーションに適用されるタイムアウト コンフィギュレーションではなく、特定の TCP アプリケーションに固有のタイムアウト コンフィギュレーションを作成できます。サービス ポリシーは、1 つのインターフェイスに適用されるか、またはグローバルに適用される複数のアクションまたはルールで構成されます。

- サービス ポリシーについて （1 ページ）
- サービス ポリシーのガイドライン （9 ページ）
- サービス ポリシーのデフォルト （11 ページ）
- サービス ポリシーの設定 （12 ページ）
- サービス ポリシーのモニタリング （21 ページ）
- サービス ポリシー（モジュラ ポリシー フレームワーク）の例 （21 ページ）
- サービス ポリシーの履歴 （24 ページ）

サービス ポリシーについて

次の各トピックでは、サービス ポリシーの仕組みについて説明します。

サービス ポリシーのコンポーネント

サービス ポリシーのポイントは、許可しているトラフィックに高度なサービスを適用することです。アクセス ルールによって許可されるトラフィックにサービス ポリシーを適用し、サービス モジュールへのリダイレクトやアプリケーション インスペクションの適用などの特別な処理を実行できます。

次のタイプのサービス ポリシーを使用できます。

- すべてのインターフェイスに適用される 1 つのグローバル ポリシー。
- インターフェイスごとに適用される 1 つのサービス ポリシー。このポリシーは、デバイスを通過するトラフィックを対象とするクラスと、ASA インターフェイスに向けられた（イ

ンターフェイスを通過するのではない）管理トラフィックを対象とするクラスの組み合わせである場合があります。

各サービス ポリシーは、次の要素で構成されます。

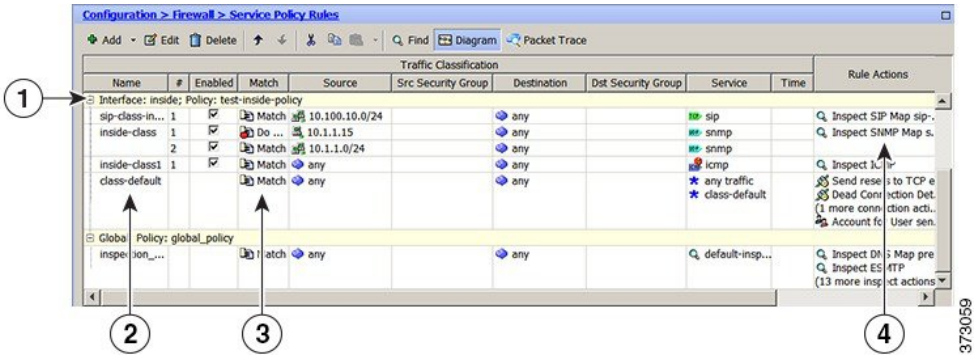
1. サービス ポリシー マップ。これはルールの順序セットであり、**service-policy** コマンドで命名されます。ASDM では、ポリシー マップは [Service Policy Rules] ページにフォルダとして表示されます。

2. ルール。各ルールは、サービス ポリシー内の、**class** コマンドと **class** に関連するコマンド群で構成されます。ASDM では、各ルールは個別の行に表示され、ルールの名前はクラス名です。

   class コマンドは、ルールのトラフィック照合基準を定義します。

   inspect や set connection timeout などの class 関連のコマンドは、一致するトラフィックに適用するサービスと制約を定義します。inspect コマンドは、検査対象トラフィックに適用するアクションを定義するインスペクション ポリシー マップを指す場合があります。インスペクション ポリシー マップとサービス ポリシー マップは同じではないことに注意してください。

次の例では、サービス ポリシーが CLI と ASDM でどのように表示されるかを比較します。図の吹き出しと CLI の行は 1 対 1 で対応しないことに注意してください。



次の CLI は、上の図に示すルールによって生成されます。

```
: Access lists used in class maps.
: In ASDM, these map to call-out 3, from the Match to the Time fields.
access-list inside_mpc line 1 extended permit tcp 10.100.10.0 255.255.255.0 any eq sip
access-list inside_mpc_1 line 1 extended deny udp host 10.1.1.15 any eq snmp
access-list inside_mpc_1 line 2 extended permit udp 10.1.1.0 255.255.255.0 any eq snmp
access-list inside_mpc_2 line 1 extended permit icmp any any
: SNMP map for SNMP inspection. Denies all but v3.
: In ASDM, this maps to call-out 4, rule actions, for the class-inside policy.
snmp-map snmp-v3only
  deny version 1
  deny version 2
  deny version 2c
: Inspection policy map to define SIP behavior.
: The sip-high inspection policy map must be referred to by an inspect sip command
: in the service policy map.
: In ASDM, this maps to call-out 4, rule actions, for the sip-class-inside policy.
```

```
policy-map type inspect sip sip-high
  parameters
    rtp-conformance enforce-payloadtype
    no traffic-non-sip
    software-version action mask log
    uri-non-sip action mask log
    state-checking action drop-connection log
    max-forwards-validation action drop log
    strict-header-validation action drop log
: Class map to define traffic matching for the inside-class rule.
: In ASDM, this maps to call-out 3, from the Match to the Time fields.
class-map inside-class
  match access-list inside_mpc_1
: Class map to define traffic matching for the sip-class-inside rule.
: In ASDM, this maps to call-out 3, from the Match to the Time fields.
class-map sip-class-inside
  match access-list inside_mpc
: Class map to define traffic matching for the inside-class1 rule.
: In ASDM, this maps to call-out 3, from the Match to the Time fields.
class-map inside-class1
  match access-list inside_mpc_2
: Policy map that actually defines the service policy rule set named test-inside-policy.
: In ASDM, this corresponds to the folder at call-out 1.
policy-map test-inside-policy
: First rule in test-inside-policy, named sip-class-inside. Inspects SIP traffic.
: The sip-class-inside rule applies the sip-high inspection policy map to SIP inspection.
: In ASDM, each rule corresponds to call-out 2.
  class sip-class-inside
    inspect sip sip-high
: Second rule, inside-class. Applies SNMP inspection using an SNMP map.
  class inside-class
    inspect snmp snmp-v3only
: Third rule, inside-class1. Applies ICMP inspection.
  class inside-class1
    inspect icmp
: Fourth rule, class-default. Applies connection settings and enables user statistics.
  class class-default
    set connection timeout embryonic 0:00:30 half-closed 0:10:00 idle 1:00:00
reset dcd 0:15:00 5
    user-statistics accounting
: The service-policy command applies the policy map rule set to the inside interface.
: This command activates the policies.
service-policy test-inside-policy interface inside
```

サービス ポリシーで設定される機能

次の表に、サービス ポリシーを使用して設定する機能を示します。

表 *1*: サービス ポリシーで設定される機能

機能	通過トラフィック用か	管理トラフィック用か	次を参照してください。
アプリケーション インスペクション（複数タイプ）	RADIUS アカウンティングを除く**すべて**	**RADIUS アカウンティング**のみ	• アプリケーション レイヤ プロトコル インスペクションの準備。 • 基本インターネット プロトコルのインスペクション。 • 音声とビデオのプロトコルのインスペクション。 • モバイル ネットワークのインスペクション。
NetFlow セキュア イベント ロギングのフィルタリング	**対応**	**対応**	NetFlow 実装ガイドを参照してください。
QoS 入出力ポリシング	**対応**	非対応	QoS。
QoS 標準プライオリティ キュー	**対応**	非対応	QoS。
TCP と UDP の接続制限値とタイムアウト、および TCP シーケンス番号のランダム化	**対応**	**対応**	接続設定。
TCP の正規化	**対応**	非対応	接続設定。
TCP ステート バイパス	**対応**	非対応	接続設定。
アイデンティティ ファイアウォールのユーザー統計情報	**対応**	**対応**	コマンド リファレンスの user-statistics コマンドを参照してください。

機能の方向性

アクションは、機能に応じて双方向または単方向にトラフィックに適用されます。双方向に適用される機能の場合、トラフィックが両方向のクラスマップと一致した場合に、ポリシーマップを適用するインターフェイスを出入りするすべてのトラフィックが影響を受けます。

（注） グローバル ポリシーを使用する場合は、すべての機能が単方向です。単一インターフェイスに適用する場合に通常双方向の機能は、グローバルに適用される場合、各インターフェイスの入力にのみ適用されます。ポリシーはすべてのインターフェイスに適用されるため、ポリシーは両方向に適用され、この場合の双方向は冗長になります。

QoS プライオリティ キューなど単方向に適用される機能の場合は、ポリシー マップを適用するインターフェイスに出入りする（機能によって異なります）トラフィックだけが影響を受けます。各機能の方向については、次の表を参照してください。

表 *2*：機能の方向性

機能	単一インターフェイスでの方向	グローバルでの方向
アプリケーション インスペクション（複数タイプ）	双方向	入力
NetFlow セキュア イベント ロギングのフィルタリング	該当なし	入力
QoS 入力ポリシング	入力	入力
QoS 出力ポリシング	出力	出力
QoS 標準プライオリティ キュー	出力	出力
TCP と UDP の接続制限値とタイムアウト、および TCP シーケンス番号のランダム化	双方向	入力
TCP の正規化	双方向	入力
TCP ステート バイパス	双方向	入力
アイデンティティ ファイアウォールのユーザー統計情報	双方向	入力

サービス ポリシー内の機能照合

パケットは、次のルールに従って特定のインターフェイスのポリシー マップのクラス マップに一致します。

1. パケットは、各機能タイプのポリシー マップ ルールで、1 つのクラス マップにだけ一致します。
2. パケットが機能タイプのクラス マップに一致した場合、ASA は、その機能タイプの後続のクラス マップとは照合しません。
3. ただし、パケットが別の機能タイプの後続のクラス マップと一致した場合、ASA は、後続のクラスマップのアクションも適用します（サポートされている場合）。サポートされていない組み合わせの詳細については、特定の機能アクションの非互換性 （7 ページ）を参照してください。

（注） アプリケーション インスペクションには、複数のインスペクション タイプが含まれ、ほとんどのタイプは相互に排他的です。組み合わせ可能なインスペクションの場合、各インスペクションは個々の機能と見なされます。

パケット照合の例

次に例を示します。

- パケットが接続制限値のクラス マップ と一致し、アプリケーション インスペクションのクラス マップ とも一致した場合、両方のクラス マップ アクションが適用されます。
- パケットが HTTP インスペクションで 1 つのクラス マップと一致し、HTTP インスペクションを含む別のクラス マップ とも一致した場合、2 番目のクラス マップ のアクションは適用されません。
- パケットが FTP インスペクションで 1 つのクラスマップ と一致し、HTTP インスペクションを含む別のクラス マップ とも一致した場合、HTTP および FTP インスペクションは組み合わせることができないため、2 番目のクラスマップのアクションは適用されません。
- パケットが HTTP インスペクションで 1 つのクラス マップ と一致し、さらに IPv6 インスペクションを含む別のクラス マップ とも一致した場合、IPv6 インスペクションは他のタイプのインスペクションと組み合わせることができるため、両方のアクションが適用されます。

複数の機能アクションが適用される順序

ポリシー マップの各種のアクションが実行される順序は、ポリシー マップ 中に出現する順序とは無関係です。

アクションは次の順序で実行されます。

1. QoS 入力ポリシング
2. TCP の正規化、TCP と UDP の接続制限値とタイムアウト、TCP シーケンス番号のランダム化、および TCP ステート バイパス

（注） ASA がプロキシサービス（AAA など）を実行したり、TCP ペイロード（FTP インスペクションなど）を変更したりするときは、TCP ノーマライザはデュアルモードで動作します。その場合、サービスを変更するプロキシやペイロードの前後で適用されます。

3. 他のインスペクションと組み合わせることができるアプリケーション インスペクション：
   1. IPv6

2. IP オプション
3. WAAS

4. 他のインスペクションと組み合わせることができないアプリケーション インスペクション：詳細については、「特定の機能アクションの非互換性 （7 ページ）」を参照してください。
5. QoS 出力ポリシング
6. QoS 標準プライオリティ キュー

（注） NetFlow セキュア イベント ロギングのフィルタリングとアイデンティティ ファイアウォールのユーザー統計情報は順番に依存しません。

特定の機能アクションの非互換性

一部の機能は同じトラフィックに対して相互に互換性がありません。次のリストには、すべての非互換性が含まれていない場合があります。各機能の互換性については、機能に関する章または項を参照してください。

- QoS プライオリティ キューイングと QoS ポリシングは同じトラフィックの集合に対して設定できません。
- ほとんどのインスペクションは別のインスペクションと組み合わせられないため、同じトラフィックに複数のインスペクションを設定しても、ASA は 1 つのインスペクションだけを適用します。例外は、複数の機能アクションが適用される順序 （6 ページ）に記載されています。

（注） デフォルト グローバル ポリシーで使用される **match default-inspection-traffic** コマンドは、デフォルト ポートをすべてのインスペクションと照合する特別な CLI ショートカットです。ポリシーマップで使用すると、このクラスマップでは、トラフィックの宛先ポートに基づいて、各パケットに正しいインスペクションが適用されます。たとえば、宛先がポート 69 の UDP トラフィックが ASA に到達すると、ASA は TFTP インスペクションを適用し、宛先がポート 21 の TCP トラフィックが到着すると、ASA は FTP インスペクションを適用します。そのため、この場合に限って同じクラス マップに複数のインスペクションを設定できます。通常、ASA は、ポート番号を使用して適用するインスペクションを決定しないため、標準以外のポートなどにも柔軟にインスペクションを適用できます。

誤った設定例は、同じポリシー マップに複数のインスペクションを設定しても、default-inspection-traffic ショートカットを使用しないことです。最初の例では、ポート 21 宛てのトラフィックが、FTP インスペクションと HTTP インスペクションの両方に誤って設定されています。2 番目の例では、ポート 80 宛てのトラフィックが、FTP インスペクションと HTTP

インスペクションの両方に誤って設定されています。どちらの誤った設定例の場合も、FTPインスペクションだけが適用されています。これは、適用されたインスペクションの順序では、FTP が HTTP よりも先になるためです。

例 1：**FTP パケットの誤設定（HTTP インスペクションも設定されている）**

```
class-map ftp
   match port tcp eq 21
class-map http
   match port tcp eq 21 [it should be 80]
policy-map test
   class ftp
     inspect ftp
   class http
     inspect http
```

例 2：**HTTP パケットの誤設定（FTP インスペクションも設定されている）**

```
class-map ftp
   match port tcp eq 80 [it should be 21]
class-map http
   match port tcp eq 80
policy-map test
   class ftp
     inspect ftp
   class http
     inspect http
```

複数のサービス ポリシーの機能照合

TCP および UDP トラフィック（およびステートフル ICMP インスペクションがイネーブルの場合は ICMP）の場合、サービス ポリシーはトラフィック フローに対して作用し、個々のパケットに限定されません。トラフィックが、1 つのインターフェイスのポリシーで定義されている機能に一致する既存の接続の一部である場合、そのトラフィック フローを別のインターフェイスのポリシーにある同じ機能と照合することはできません。最初のポリシーのみが使用されます。

たとえば、HTTP トラフィックが、HTTP トラフィックを検査する内部インターフェイスのポリシーと一致するときに、HTTP インスペクション用の外部インターフェイスに別のポリシーがある場合、そのトラフィックが外部インターフェイスの出力側でも検査されることはありません。同様に、その接続のリターン トラフィックが外部インターフェイスの入力ポリシーによって検査されたり、内部インターフェイスの出力ポリシーによって検査されたりすることもありません。

ステートフル ICMP インスペクションをイネーブルにしない場合の ICMP のように、フローとして扱われないトラフィックの場合は、リターン トラフィックを戻り側のインターフェイスの別のポリシー マップと照合できます。

サービス ポリシーのガイドライン

インスペクションのガイドライン

アプリケーション インスペクションのサービス ポリシーに関する詳細なガイドラインを提供する単独のトピックがあります。アプリケーション インスペクションのガイドラインを参照してください。

IPv6 のガイドライン

IPv6 は次の機能でサポートされています。

- 複数の、しかしすべてではないプロトコルに対するアプリケーション インスペクション。詳細については、アプリケーション インスペクションのガイドラインを参照してください。
- NetFlow セキュア イベント ロギングのフィルタリング
- SCTP ステート バイパス
- TCP と UDP の接続制限値とタイムアウト、および TCP シーケンス番号のランダム化
- TCP の正規化
- TCP ステート バイパス
- アイデンティティ ファイアウォールのユーザー統計情報

クラスマップ（トラフィック クラス）のガイドライン

すべてのタイプのクラスマップ（トラフィック クラス）の最大数は、シングル モードでは 255 個、マルチ モードではコンテキストごとに 255 個です。クラス マップには、次のタイプがあります。

- レイヤ 3/4 クラスマップ（通過トラフィックと管理トラフィック向け）。
- インスペクション クラス マップ
- 正規表現クラス マップ
- **match** インスペクション ポリシー マップ下で直接使用されるコマンド

この制限には、すべてのタイプのデフォルト クラス マップも含まれ、ユーザー設定のクラス マップを約 235 に制限します。

ポリシー マップのガイドライン

ポリシー マップを使用する場合は、次のガイドラインを参考にしてください。

- 各インターフェイスには、ポリシーマップを1つだけ割り当てることができますローエンドのファイアウォールの場合、設定で最大 64 個のポリシーマップを作成できます。より強力なファイアウォールの場合、最大で 128 個まで作成できます。

  各インターフェイスには、ポリシー マップを 1 つだけ割り当てることができますCisco Secure Firewall 3100 のようなローエンドのファイアウォールの場合、設定では最大 64 個のポリシーマップを作成できます。Firepower 4100 など、より強力なファイアウォールでは、最大で 128 個まで作成できます。

- 同一のポリシー マップを複数のインターフェイスに適用できます。
- 1 つのレイヤ 3/4 ポリシー マップで最大 63 のレイヤ 3/4 クラス マップを識別できます。
- クラスマップごとに、1つ以上の機能タイプから複数のアクションを割り当てることができます（サポートされている場合）。特定の機能アクションの非互換性 （7 ページ）を参照してください。

サービス ポリシーのガイドライン

- 入力インターフェイスのインターフェイス サービス ポリシーは、特定の機能に対するグローバル サービス ポリシーより優先されます。たとえば、FTPインスペクションのグローバル ポリシーと、TCP 正規化のインターフェイス ポリシーがある場合、FTP インスペクションと TCP 正規化の両方がインターフェイスに適用されます。これに対し、FTP インスペクションのグローバル ポリシーと、FTP インスペクションの入力インターフェイス ポリシーがある場合は、入力インターフェイス ポリシーの FTP インスペクションだけがそのインターフェイスに適用されます。入力またはグローバル ポリシーが機能を実装していない場合は、機能を指定する出力インターフェイスのインターフェイス サービス ポリシーが適用されます。
- 適用できるグローバル ポリシーは 1 つだけです。たとえば、機能セット 1 が含まれたグローバル ポリシーと、機能セット 2 が含まれた別のグローバル ポリシーを作成できません。すべての機能は 1 つのポリシーに含める必要があります。
- コンフィギュレーションに対してサービス ポリシーの変更を加えた場合は、すべての新しい接続で新しいサービス ポリシーが使用されます。既存の接続では、その接続が確立された時点で設定されていたポリシーの使用が続行されます。show コマンドの出力には、古い接続に関するデータは含まれません。

  たとえば、インターフェイスから QoS サービス ポリシーを削除し、変更したバージョンを追加した場合、**show service-policy** コマンドには、新しいサービス ポリシーに一致する新しい接続に関連付けられた QoS カウンタだけが表示されます。古いポリシーの既存の接続はコマンド出力には表示されなくなります。

  すべての接続が新しいポリシーを確実に使用するように、現在の接続を解除し、新しいポリシーを使用して再度接続できるようにします。**clear conn** または **clear local-host** コマンドを使用します。

サービス ポリシーのデフォルト

次の各トピックでは、サービス ポリシーとモジュラ ポリシー フレームワークのデフォルト設定について説明します。

デフォルトのサービス ポリシー設定

デフォルトでは、すべてのデフォルト アプリケーション インスペクション トラフィックに一致するポリシーがコンフィギュレーションに含まれ、特定のインスペクションがすべてのインターフェイスのトラフィックに適用されます（グローバル ポリシー）。すべてのインスペクションがデフォルトでイネーブルになっているわけではありません。適用できるグローバル ポリシーは1つだけなので、グローバル ポリシーを変更する場合は、デフォルトのポリシーを編集するか、デフォルトのポリシーをディセーブルにして新しいポリシーを適用します。（特定の機能では、グローバル ポリシーはインターフェイス ポリシーより優先されます）。

デフォルト ポリシーには、次のアプリケーション インスペクションが含まれます。

- DNS
- FTP
- H323（H225）
- H323（RAS）
- RSH
- RTSP
- ESMTP
- SQLnet
- Skinny（SCCP）
- SunRPC
- SIP
- NetBios
- TFTP
- IP オプション

デフォルトのクラス マップ（トラフィック クラス）

設定には、ASA が default-inspection-traffic Default Inspection Trafficというデフォルト グローバル ポリシーで使用するデフォルトのレイヤ 3/4 クラス マップ（トラフィック クラス）が含まれます。このクラス マップは、デフォルトのインスペクション トラフィックを照合します。

デフォルト グローバル ポリシーで使用されるこのクラスは、デフォルト ポートをすべてのインスペクションと照合する特別なショートカットです。

ポリシーで使用すると、このクラスでは、トラフィックの宛先ポートに基づいて、各パケットに正しいインスペクションが適用されます。たとえば、宛先がポート 69 の UDP トラフィックが ASA に到達すると、ASA は TFTP インスペクションを適用し、宛先がポート 21 の TCP トラフィックが到着すると、ASA は FTP インスペクションを適用します。そのため、この場合に限って同じクラス マップに複数のインスペクションを設定できます。通常、ASA は、ポート番号を使用して適用するインスペクションを決定しないため、標準以外のポートなどにも柔軟にインスペクションを適用できます。

```
class-map inspection_default
 match default-inspection-traffic
```

デフォルト コンフィギュレーションにある別のクラス マップは、class-default と呼ばれ、すべてのトラフィックと一致します。このクラス マップは、すべてのレイヤ 3/4 ポリシー マップの最後に示され、原則的に、他のすべてのトラフィックでどのようなアクションも実行しないように ASA に通知します。class-default クラス（ASDM の任意の通信クラス）を使用する場合は、TCP シーケンスのランダム化、TTL デクリメント、またはその他のオプション属性の変更を有効にするなど、より広範な目的で使用してください。インスペクションには使用しないでください。使用すると、通信の中断が発生する可能性があります。一部の機能は class-default でしか使用できません。

```
class-map class-default
 match any
```

サービス ポリシーの設定

モジュラ ポリシー フレームワークを使用してサービス ポリシーを設定するには、次の手順を実行します。

手順

ステップ 1 トラフィックの特定（レイヤ 3/4 クラス マップ） （14 ページ） の説明に従って、レイヤ 3/4 クラス マップを作成して、操作対象のトラフィックを特定します。

たとえば、ASA を通過するすべてのトラフィックでアクションを実行したり、10.1.1.0/24 から任意の宛先アドレスまでのトラフィックで特定のアクションだけを実行したりできます。



ステップ 2 必要に応じて、あるインスペクション トラフィックで追加のアクションを実行します。

実行するアクションの 1 つがアプリケーション インスペクションで、一部のインスペクション トラフィックで追加のアクションを実行する場合、インスペクション ポリシー マップを作成します。インスペクション ポリシー マップはトラフィックを特定し、そのトラフィックで何をするかを指定します。

たとえば、本文の長さが 1000 バイトを上回るすべての HTTP 要求をドロップできます。

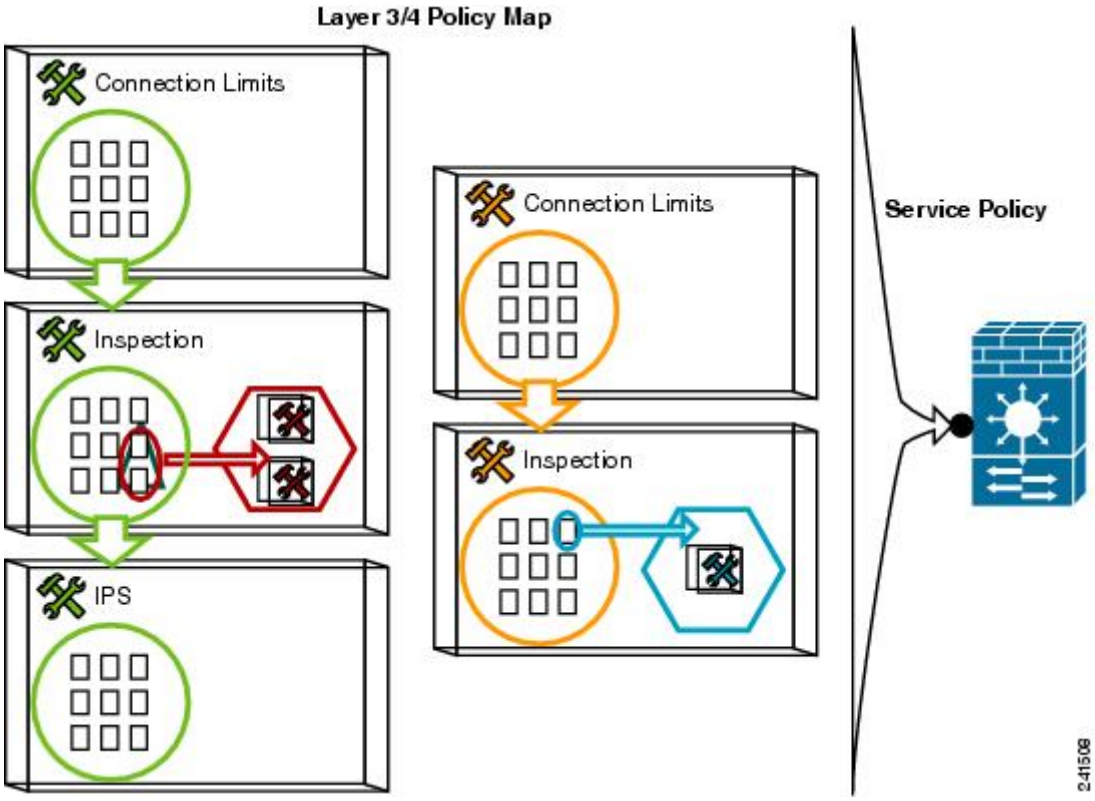


match コマンドでトラフィックを直接特定する独立したインスペクション ポリシー マップを作成したり、再利用のために、またはより複雑な照合のためにインスペクション クラス マップを作成したりできます。たとえば、正規表現または正規表現のグループ（正規表現クラス マップ）を使用して検査対象のパケット内のテキストを照合し、より限定された基準に基づいてアクションの対象を設定できます。たとえば、「example.com」というテキストが含まれた URL を持つすべての HTTP 要求をドロップできます。



アプリケーション レイヤ プロトコル インスペクションの設定を参照してください。

ステップ 3 アクションの定義（レイヤ 3/4 ポリシー マップ） （18 ページ）の説明に従って、レイヤ 3/4 ポリシーマップを作成して、各レイヤ 3/4 クラスマップで実行するアクションを定義します。



ステップ 4 インターフェイス（サービス ポリシー）へのアクションの適用 （20 ページ）の説明に従って、ポリシーマップを適用するインターフェイスを決定するか、ポリシーマップをグローバルに適用します。

トラフィックの特定（レイヤ 3/4 クラス マップ）

レイヤ 3/4 クラス マップにより、アクションを適用するレイヤ 3 および 4 のトラフィックを特定します。1 つのレイヤ 3/4 ポリシー マップに複数のレイヤ 3/4 クラス マップを作成できます。

通過トラフィック用のレイヤ 3/4 クラス マップの作成

レイヤ 3/4 クラス マップでは、プロトコル、ポート、IP アドレス、およびレイヤ 3 またはレイヤ 4 の他の属性に基づいてトラフィックを照合します。

ヒント トラフィック インスペクションは、アプリケーション トラフィックが発生するポートだけで行うことをお勧めします。**match any** などを使用してすべてのトラフィックを検査すると、ASA のパフォーマンスに影響が出る場合があります。

手順

ステップ 1 レイヤ 3/4 クラス マップを作成します。**class-map** *class_map_name*

class_map_name は、最大 40 文字の文字列です。

「class-default」という名前は予約されています。すべてのタイプのクラス マップで同じ名前スペースが使用されるため、別のタイプのクラスマップですでに使用されている名前は再度使用できません。CLI はクラスマップ コンフィギュレーション モードに移行します。

例：

```
hostname(config)# class-map all_udp
```

ステップ 2 （任意）説明をクラス マップに追加します。

description *string*

例：

```
hostname(config-cmap)# description All UDP traffic
```

ステップ 3 次のいずれかのコマンドを使用してトラフィックを照合します。特に指定がない場合、クラスマップに含めることができる **match** コマンドは 1 つだけです。

- **match any**：すべてのトラフィックを照合します。

  ```
  hostname(config-cmap)# match any
  ```

- **match access-list** *access_list_name*：拡張アクセス リストで指定されているトラフィックを照合します。

  ```
  hostname(config-cmap)# match access-list udp
  ```

- **match port** {**tcp** | **udp** | **sctp**} {**eq** *port_num* | **range** *port_num port_num*}：指定されたプロトコルに対し、宛先ポート（単一のポートまたは連続する範囲のポート）を照合します。複数の非連続ポートを使用するアプリケーションに対しては、match access-list コマンドを使用して、各ポートと一致する ACE を定義します。

  ```
  hostname(config-cmap)# match tcp eq 80
  ```

- **match default-inspection-traffic**：インスペクション用のデフォルト トラフィックを照合します（ASA が検査可能なすべてのアプリケーションによって使用されるデフォルトの TCP および UDP ポート）。

  ```
  hostname(config-cmap)# match default-inspection-traffic
  ```

デフォルト グローバル ポリシーで使用されるこのコマンドは、ポリシー マップで使用されると、トラフィックの宛先ポートに基づいて各パケットに正しいインスペクションを適用する特別な CLI ショートカットです。たとえば、宛先がポート 69 の UDP トラフィックが ASA に到達すると、ASA は TFTP インスペクションを適用し、宛先がポート 21 の TCP トラフィックが到着すると、ASA は FTP インスペクションを適用します。そのため、この場合に限って同じクラスマップに複数のインスペクションを設定できます（他のインスペクションとともに設定可能な WAAS インスペクションを除きます。アクションの組み合わせの詳細については、特定の機能アクションの非互換性 （7 ページ）を参照してください）。通常、ASA は、ポート番号を使用して適用するインスペクションを決定しないため、標準以外のポートなどにも柔軟にインスペクションを適用できます。

デフォルト ポートのリストについては、デフォルト インスペクションと NAT に関する制限事項を参照してください。**match default-inspection-traffic** コマンドにポートが含まれているすべてのアプリケーションが、ポリシーマップでデフォルトでイネーブルになっているわけではありません。

match access-list コマンドを **match default-inspection-traffic** コマンドとともに指定すると、一致するトラフィックを絞り込むことができます。**match default-inspection-traffic** コマンドによって照合するポートとプロトコルが指定されるため、ACL のポートとプロトコルはすべて無視されます。

- **match dscp** *value1* [*value2*] [...] [*value8*]：IP ヘッダーの DSCP 値（最大 8 個の DSCP 値）と照合します。

```
hostname(config-cmap)# match dscp af43 cs1 ef
```

- **match precedence** *value1* [*value2*] [*value3*] [*value4*]：IP ヘッダーの TOS バイトによって表される、最大 4 つの Precedence 値を照合します。Precedence 値は 0 ～ 7 に指定できます。

```
hostname(config-cmap)# match precedence 1 4
```

- **match rtp** *starting_port range*：RTP トラフィックを照合します。*starting_port* には、2000 ～ 65534 の間の偶数の UDP 宛先ポートを指定します。*range* には、*starting_port* よりも上の追加 UDP ポートの数を 0 ～ 16383 で指定します。

```
hostname(config-cmap)# match rtp 4004 100
```

- **match tunnel-group** *name*：QoS を適用する VPN トンネル グループ トラフィックを照合します。

  トラフィック照合を調整するために、**match** コマンドをもう 1 つ指定できます。上記のコマンドのいずれかを指定できますが、**match any**、**match access-list**、および **match default-inspection-traffic** コマンドは指定できません。または、match flow ip destination-address コマンドを入力して、各 IP アドレス宛てのトンネル グループのフローを照合することもできます。

```
hostname(config-cmap)# match tunnel-group group1
```

```
hostname(config-cmap)# match flow ip destination-address
```

例

次に **class-map** コマンドの例を示します。

```
hostname(config)# access-list udp permit udp any any
hostname(config)# access-list tcp permit tcp any any
hostname(config)# access-list host_foo permit ip any 10.1.1.1 255.255.255.255

hostname(config)# class-map all_udp
hostname(config-cmap)# description "This class-map matches all UDP traffic"
hostname(config-cmap)# match access-list udp

hostname(config-cmap)# class-map all_tcp
hostname(config-cmap)# description "This class-map matches all TCP traffic"
hostname(config-cmap)# match access-list tcp

hostname(config-cmap)# class-map all_http
hostname(config-cmap)# description "This class-map matches all HTTP traffic"
hostname(config-cmap)# match port tcp eq http

hostname(config-cmap)# class-map to_server
hostname(config-cmap)# description "This class-map matches all traffic to server 10.1.1.1"
hostname(config-cmap)# match access-list host_foo
```

管理トラフィック用のレイヤ 3/4 クラス マップの作成

ASAへの管理トラフィックに対して、この種類のトラフィックに特有のアクションの実行が必要になる場合があります。管理クラス マップを指定して、ACL または TCP や UDP のポートと照合できます。ポリシー マップの管理クラス マップで設定可能なアクションのタイプは、管理トラフィック専用です。

手順

ステップ 1 管理クラス マップを作成します。**class-map type management** *class_map_name*

class_map_name は、最大 40 文字の文字列です。

「class-default」という名前は予約されています。すべてのタイプのクラス マップで同じ名前スペースが使用されるため、別のタイプのクラスマップですでに使用されている名前は再度使用できません。CLI はクラスマップ コンフィギュレーション モードに移行します。

例：

```
hostname(config)# class-map management all_udp
```

ステップ 2 （任意）説明をクラス マップに追加します。

description *string*

例：

```
hostname(config-cmap)# description All UDP traffic
```

ステップ 3 次のいずれかのコマンドを使用してトラフィックを照合します。

- **match access-list** *access_list_name*：拡張アクセス リストで指定されているトラフィックを照合します。

```
hostname(config-cmap)# match access-list udp
```

- **match port** {**tcp** | **udp** | **sctp**} {**eq** *port_num* | **range** *port_num port_num*}：指定されたプロトコルに対し、宛先ポート（単一のポートまたは連続する範囲のポート）を照合します。複数の非連続ポートを使用するアプリケーションに対しては、match access-list コマンドを使用して、各ポートと一致する ACE を定義します。

```
hostname(config-cmap)# match tcp eq 80
```

アクションの定義（レイヤ 3/4 ポリシー マップ）

トラフィックを識別するレイヤ 3/4 クラス マップを設定したら、レイヤ 3/4 ポリシー マップを使用してそれらのクラスにアクションを関連付けます。

ヒント ポリシー マップの最大数は 64 ですが、各インターフェイスには、ポリシー マップを 1 つだけ適用できます。

手順

ステップ 1 ポリシー マップを追加します。**policy-map** *policy_map_name*

policy_map_name は、最大 40 文字のポリシー マップ名です。すべてのタイプのポリシー マップで同じ名前スペースが使用されるため、別のタイプのポリシーマップですでに使用されている名前は再度使用できません。CLI はポリシー マップ コンフィギュレーション モードに入ります。

例：

```
hostname(config)# policy-map global_policy
```

ステップ 2 以前に設定したレイヤ 3/4 クラス マップを指定します。**class** *class_map_name*

class_map_name には、クラス マップの名前を指定します。

クラス マップを追加するには、トラフィックの特定（レイヤ 3/4 クラス マップ） （14 ページ）を参照してください。

例：

```
hostname(config-pmap)# class all_http
```

ステップ 3 このクラス マップに、1 つ以上のアクションを指定します。

サービス ポリシーで設定される機能 （3 ページ）を参照してください。

（注） クラス マップに **match default-inspection-traffic** コマンドがない場合、そのクラスに最大 1 つの **inspect** コマンドを設定できます。

ステップ 4 このポリシー マップに含めるクラス マップごとに、この手順を繰り返します。

例

接続ポリシーの **policy-map** コマンドの例を次に示します。このコマンドは、Web サーバー 10.1.1.1 への接続許可数を制限します。

```
hostname(config)# access-list http-server permit tcp any host 10.1.1.1
hostname(config)# class-map http-server
hostname(config-cmap)# match access-list http-server

hostname(config)# policy-map global-policy
hostname(config-pmap)# description This policy map defines a policy concerning
connection to http server.
hostname(config-pmap)# class http-server
hostname(config-pmap-c)# set connection conn-max 256
```

次の例は、ポリシー マップでの複数の照合の動作を示しています。

```
hostname(config)# class-map inspection_default
hostname(config-cmap)# match default-inspection-traffic
hostname(config)# class-map http_traffic
hostname(config-cmap)# match port tcp eq 80

hostname(config)# policy-map outside_policy
hostname(config-pmap)# class inspection_default
hostname(config-pmap-c)# inspect http http_map
hostname(config-pmap-c)# inspect sip
hostname(config-pmap)# class http_traffic
hostname(config-pmap-c)# set connection timeout idle 0:10:0
```

次の例は、トラフィックが最初の利用可能なクラスマップと一致した場合に、同じ機能ドメインのアクションが指定されている後続のクラスマップと照合されないことを示しています。

```
hostname(config)# class-map telnet_traffic
hostname(config-cmap)# match port tcp eq 23
hostname(config)# class-map ftp_traffic
hostname(config-cmap)# match port tcp eq 21
hostname(config)# class-map tcp_traffic
hostname(config-cmap)# match port tcp range 1 65535
hostname(config)# class-map udp_traffic
hostname(config-cmap)# match port udp range 0 65535
hostname(config)# policy-map global_policy
hostname(config-pmap)# class telnet_traffic
hostname(config-pmap-c)# set connection timeout idle 0:0:0
hostname(config-pmap-c)# set connection conn-max 100
hostname(config-pmap)# class ftp_traffic
hostname(config-pmap-c)# set connection timeout idle 0:5:0
hostname(config-pmap-c)# set connection conn-max 50
hostname(config-pmap)# class tcp_traffic
hostname(config-pmap-c)# set connection timeout idle 2:0:0
hostname(config-pmap-c)# set connection conn-max 2000
```

Telnet 接続は、開始時に **class telnet_traffic** と一致します。同様に FTP 接続は、開始時に **class ftp_traffic** と一致します。Telnet および FTP 以外の TCP 接続の場合は、**class tcp_traffic** と一致します。Telnet 接続または FTP 接続は **class tcp_traffic** と一致しますが、すでに他のクラスと一致しているため、ASA はこの照合を行いません。

インターフェイス（サービス ポリシー）へのアクションの適用

レイヤ 3/4 ポリシー マップをアクティブにするには、1 つ以上のインターフェイスに適用するサービス ポリシー、またはすべてのインターフェイスにグローバルに適用するサービス ポリシーを作成します。次のコマンドを使用します。

service-policy *policy_map_name* {**global** | **interface** *interface_name*} [**fail-close**]

それぞれの説明は次のとおりです。

- *policy_map_name* は、ポリシー マップの名前です。
- **global** は、特定のポリシーを持たないすべてのインターフェイスに適用するサービス ポリシーを作成します。

  適用できるグローバル ポリシーは 1 つだけなので、グローバル ポリシーを変更する場合は、デフォルトのポリシーを編集するか、デフォルトのポリシーをディセーブルにして新しいポリシーを適用します。デフォルトでは、すべてのデフォルト アプリケーション インスペクション トラフィックに一致するグローバル ポリシーがコンフィギュレーションに含まれ、すべてのインスペクションがトラフィックにグローバルに適用されます。デフォルト サービス ポリシーには、**service-policy global_policy global** コマンドが含まれます。

- **interface** *interface_name* は、インターフェイスにポリシー マップを関連付けてサービス ポリシーを作成します。
- **fail-close** は、IPv6 トラフィックをサポートしないアプリケーション インスペクションによってドロップされた IPv6 トラフィックの syslog（767001）を生成します。デフォルトでは、syslog が生成されません。

例

たとえば、次のコマンドは、外部インターフェイスで inbound_policy ポリシー マップをイネーブルにします。

```
hostname(config)# service-policy inbound_policy interface outside
```

次のコマンドは、デフォルト グローバル ポリシーをディセーブルにし、新しいポリシー new_global_policy をイネーブルにします。

```
hostname(config)# no service-policy global_policy global
hostname(config)# service-policy new_global_policy global
```

サービス ポリシーのモニタリング

サービス ポリシーをモニターするには、次のコマンドを入力します。

- **show service-policy**

  サービス ポリシーの統計情報を表示します。

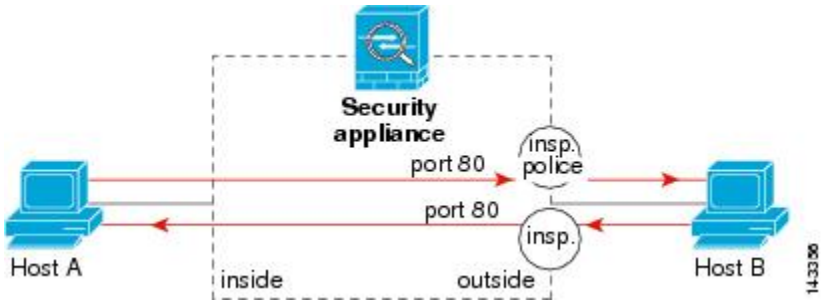
サービス ポリシー（モジュラ ポリシー フレームワーク）の例

このセクションでは、モジュラ ポリシー フレームワークの例をいくつか示します。

HTTP トラフィックへのインスペクションと QoS ポリシングの適用

この例では、外部インターフェイスを通過して ASA を出入りするすべての HTTP 接続（ポート 80 の TCP トラフィック）が HTTP インスペクション対象として分類されます。外部インターフェイスを出るすべての HTTP トラフィックがポリシング対象として分類されます。

図 *1: HTTP* インスペクションと *QoS* ポリシング



この例について、次のコマンドを参照してください。

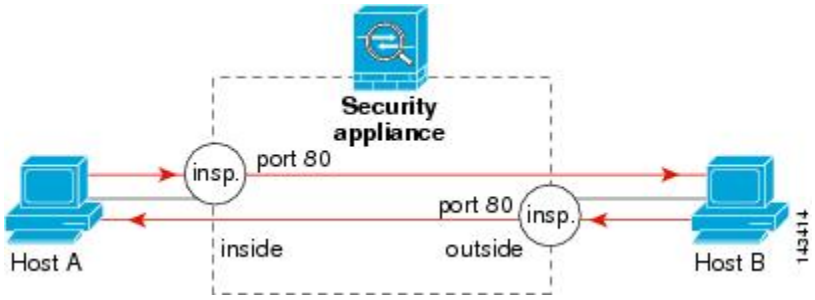
```
hostname(config)# class-map http_traffic
hostname(config-cmap)# match port tcp eq 80

hostname(config)# policy-map http_traffic_policy
hostname(config-pmap)# class http_traffic
hostname(config-pmap-c)# inspect http
hostname(config-pmap-c)# police output 250000
hostname(config)# service-policy http_traffic_policy interface outside
```

HTTP トラフィックへのインスペクションのグローバルな適用

この例では、任意のインターフェイスを通過して ASA に入るすべての HTTP 接続（ポート 80 の TCP トラフィック）が HTTP インスペクション対象として分類されます。このポリシーはグローバルポリシーなので、インスペクションが発生するのは各インターフェイスにトラフィックが入ったときだけです。

図 *2:* グローバル *HTTP* インスペクション



この例について、次のコマンドを参照してください。

```
hostname(config)# class-map http_traffic
hostname(config-cmap)# match port tcp eq 80

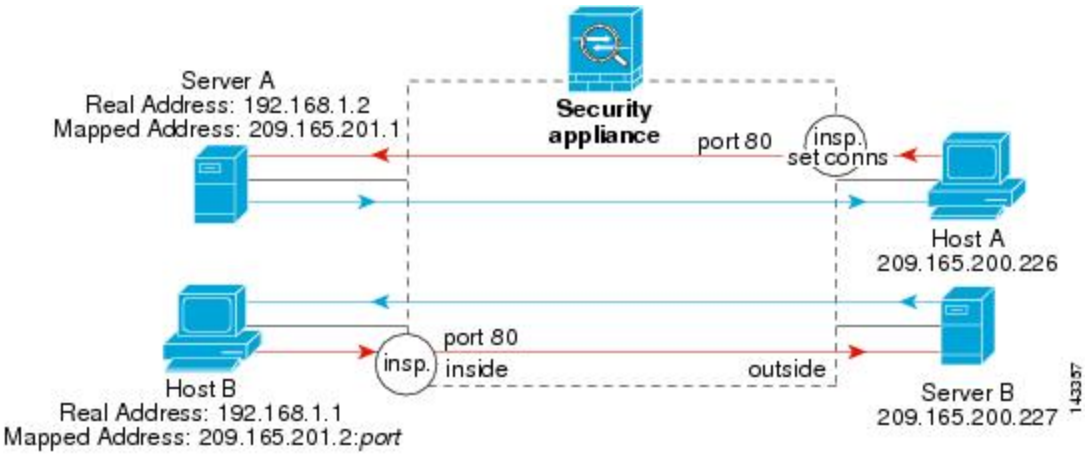
hostname(config)# policy-map http_traffic_policy
hostname(config-pmap)# class http_traffic
hostname(config-pmap-c)# inspect http
hostname(config)# service-policy http_traffic_policy global
```

特定のサーバーへの HTTP トラフィックに対するインスペクションと接続制限値の適用

この例では、外部インターフェイスを通過してASAに入るサーバーA宛てのHTTP接続（ポート 80 の TCP トラフィック）が HTTP インスペクションおよび最大接続数制限値の対象として分類されます。サーバー A から発信されたホスト A への接続は、クラスマップの ACL と一致しないので、影響を受けません。

内部インターフェイスを通じて ASA に入るサーバー B 宛てのすべての HTTP 接続は、HTTP インスペクション対象として分類されます。サーバーBから発信されたホストBへの接続は、クラスマップの ACL と一致しないので、影響を受けません。

図 *3*：特定のサーバーに対する *HTTP* インスペクションと接続制限値



この例について、次のコマンドを参照してください。

```
hostname(config)# object network obj-192.168.1.2
hostname(config-network-object)# host 192.168.1.2
hostname(config-network-object)# nat (inside,outside) static 209.165.201.1
hostname(config)# object network obj-192.168.1.0
hostname(config-network-object)# subnet 192.168.1.0 255.255.255.0
hostname(config-network-object)# nat (inside,outside) dynamic 209.165.201.2
hostname(config)# access-list serverA extended permit tcp any host 209.165.201.1 eq 80
hostname(config)# access-list ServerB extended permit tcp any host 209.165.200.227 eq
80

hostname(config)# class-map http_serverA
hostname(config-cmap)# match access-list serverA
hostname(config)# class-map http_serverB
hostname(config-cmap)# match access-list serverB

hostname(config)# policy-map policy_serverA
hostname(config-pmap)# class http_serverA
hostname(config-pmap-c)# inspect http
hostname(config-pmap-c)# set connection conn-max 100
hostname(config)# policy-map policy_serverB
hostname(config-pmap)# class http_serverB
```

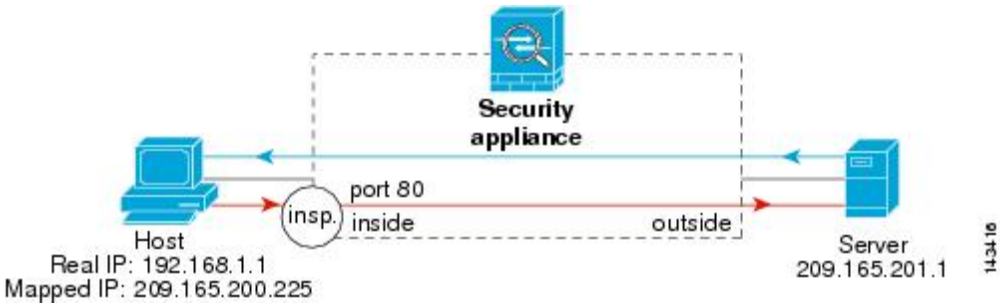
```
hostname(config-pmap-c)# inspect http

hostname(config)# service-policy policy_serverB interface inside
hostname(config)# service-policy policy_serverA interface outside
```

NAT による HTTP トラフィックへのインスペクションの適用

この例では、ネットワーク内のホストに 2 つのアドレスがあります。1 つは、実際の IP アドレスの 192.168.1.1 です。もう 1 つは、外部ネットワークで使用するマッピング IP アドレスの 209.165.200.225 です。クラス マップの ACL の実際の IP アドレスを使用する必要があります。outside インターフェイスに適用する場合にも、実際のアドレスを使用します。

図 *4: NAT* による *HTTP* インスペクション



この例について、次のコマンドを参照してください。

```
hostname(config)# object network obj-192.168.1.1
hostname(config-network-object)# host 192.168.1.1
hostname(config-network-object)# nat (VM1,outside) static 209.165.200.225

hostname(config)# access-list http_client extended permit tcp host 192.168.1.1 any eq
80

hostname(config)# class-map http_client
hostname(config-cmap)# match access-list http_client

hostname(config)# policy-map http_client
hostname(config-pmap)# class http_client
hostname(config-pmap-c)# inspect http

hostname(config)# service-policy http_client interface inside
```

サービス ポリシーの履歴

機能名	リリース	説明
モジュラ ポリシー フレームワーク	7.0(1)	モジュラ ポリシー フレームワークが導入されました。

機能名	リリース	説明
RADIUSアカウンティング トラフィックで使用する管理クラス マップ	7.2(1)	RADIUSアカウンティング トラフィックで使用する管理クラス マップが導入されました。**class-map type management** コマンドおよび**inspect radius-accounting** コマンドが導入されました。
インスペクション ポリシー マップ	7.2(1)	インスペクション ポリシー マップが導入されました。**class-map type inspect** コマンドが導入されました。
正規表現およびポリシー マップ	7.2(1)	インスペクション ポリシー マップで使用される正規表現およびポリシー マップが導入されました。**class-map type regex** コマンド、**regex** コマンド、および**match regex** コマンドが導入されました。
インスペクション ポリシー マップの match any	8.0(2)	インスペクション ポリシー マップで使用される **match any** キーワードが導入されました。トラフィックを 1 つ以上の基準に照合してクラスマップに一致させることができます。以前は、**match all** だけが使用可能でした。

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。

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。