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

次のトピックで、アプリケーション レイヤ プロトコル インスペクションを設定する方法について説明します。

アプリケーション レイヤ プロトコル インスペクション

インスペクション エンジンは、ユーザのデータ パケット内に IP アドレッシング情報を埋め込むサービスや、ダイナミックに割り当てられるポート上でセカンダリ チャネルを開くサービスに必要です。これらのプロトコルでは、高速パスでパケットを渡すのではなく、ASA で詳細なパケット インスペクションを行う必要があります（高速パスの詳細については、一般的な操作のコンフィギュレーション ガイドを参照してください）。そのため、インスペクション エンジンがスループット全体に影響を与えることがあります。ASA では、デフォルトでいくつかの一般的なインスペクション エンジンがイネーブルになっていますが、ネットワークによっては他のインスペクション エンジンをイネーブルにしなければならない場合があります。

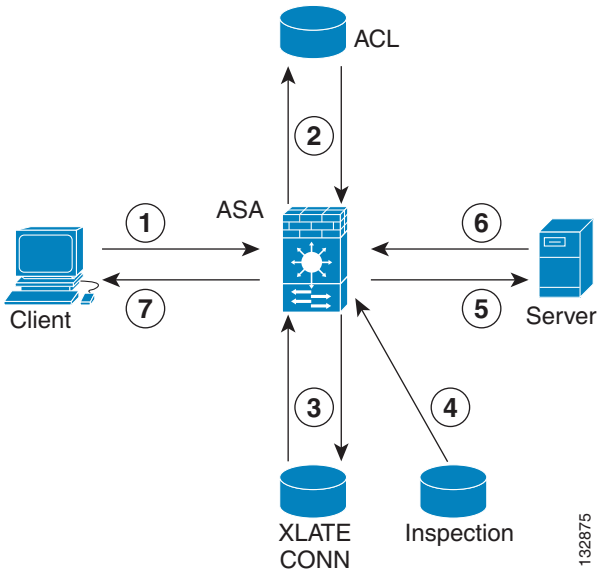
次のトピックで、アプリケーション インスペクションについて詳しく説明します。

インスペクション エンジンの動作

次の図に示すように、ASA は基本動作を行うために 3 つのデータベースを使用します。

- ACL：特定のネットワーク、ホスト、およびサービス（TCP/UDP ポート番号）に基づく接続の認証と許可のために使用されます。
- インスペクション：事前定義済みの一連のスタティックなアプリケーションレベルのインスペクション機能を含みます。
- 接続（XLATE および CONN テーブル）：確立済みの各接続についての状態および他の情報を保持します。この情報は、確立済みのセッション内でトラフィックを効率的に転送するため、アダプティブ セキュリティ アルゴリズムおよびカットスルー プロキシによって使用されます。

図 7-1　インスペクション エンジンの動作



この図では、動作の発生順に番号が付けられています。

1. TCP SYN パケットが ASA に到着して、新しい接続を確立します。
2. ASA は ACL データベースをチェックして、接続が許可されるかどうかを判定します。
3. ASA は接続データベース（XLATE および CONN テーブル）に新しいエントリを作成します。
4. ASA はインスペクション データベースをチェックして、接続にアプリケーションレベルのインスペクションが必要かどうかを判定します。
5. アプリケーション インスペクション エンジンがパケットに必要な処理を完了した後、ASA はパケットを宛先システムに転送します。
6. 宛先システムは初期要求に応答します。
7. ASA は応答パケットを受信し、接続データベースで接続を検索して、確立済みのセッションに属しているのでパケットを転送します。

ASA のデフォルト コンフィギュレーションには、サポートされるプロトコルを特定の TCP または UDP ポート番号と関連付けて、必要とされる特殊な処理を識別する、一連のアプリケーション インスペクション エントリが含まれます。

アプリケーション プロトコル インスペクションを使用するタイミング

ユーザが接続を確立すると、ASA は ACL と照合してパケットをチェックし、アドレス変換を作成し、高速パスでのセッション用にエントリを作成して、後続のパケットが時間のかかるチェックをバイパスできるようにします。ただし、高速パスは予測可能なポート番号に基づいており、パケット内部のアドレス変換を実行しません。

多くのプロトコルは、セカンダリの TCP ポートまたは UDP ポートを開きます。既知のポートで初期セッションが使用され、動的に割り当てられたポート番号がネゴシエーションされます。

パケットに IP アドレスを埋め込むアプリケーションもあります。この IP アドレスは送信元アドレスと一致する必要があり、通常、ASA を通過するときに変換されます。

これらのアプリケーションを使用する場合は、アプリケーション インスペクションをイネーブルにする必要があります。

IP アドレスを埋め込むサービスに対してアプリケーション インスペクションをイネーブルにすると、ASA は埋め込まれたアドレスを変換し、チェックサムや変換の影響を受けたその他のフィールドを更新します。

ダイナミックに割り当てられたポートを使用するサービスに対してアプリケーション インスペクションをイネーブルにすると、ASA はセッションをモニタしてダイナミックに割り当てられたポートを特定し、所定のセッションの間、それらのポートでのデータ交換を許可します。

インスペクション ポリシー マップ

*インスペクション ポリシー マップ*を使用して、多くのアプリケーション インスペクションで実行される特別なアクションを設定できます。これらのマップはオプションです。インスペクション ポリシー マップをサポートするプロトコルに関しては、マップを設定しなくてもインスペクションをイネーブルにできます。デフォルトのインスペクション アクション以外のことが必要な場合にのみ、これらのマップが必要になります。

インスペクション ポリシー マップをサポートするアプリケーションのリストについては、「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」（P.7-11）を参照してください。

インスペクション ポリシー マップは、次に示す要素の 1 つ以上で構成されています。インスペクション ポリシー マップで使用可能な実際のオプションは、アプリケーションに応じて決まります。

- トラフィック照合基準：アプリケーション トラフィックをそのアプリケーションに固有の基準（URL 文字列など）と照合し、その後アクションをイネーブルにできます。

  一部のトラフィック照合基準では、正規表現を使用してパケット内部のテキストを照合します。ポリシー マップを設定する前に、正規表現クラス マップ内で、正規表現を単独またはグループで作成およびテストしておいてください。
- インスペクション クラス マップ：一部のインスペクション ポリシー マップでは、インスペクション クラス マップを使用して複数のトラフィック照合基準を含めることができます。その後、インスペクション ポリシー マップ内でインスペクション クラス マップを指定し、そのクラス全体でアクションをイネーブルにします。クラス マップを作成することと、インスペクション ポリシー マップ内で直接トラフィック照合を定義することの違いは、より複雑な一致基準を作成できる点と、クラス マップを再使用できる点です。ただし、異なる照合基準に対して異なるアクションを設定することはできません。
- パラメータ：パラメータは、インスペクション エンジンの動作に影響します。

次のトピックで、詳細に説明します。

- 「使用中のインスペクション ポリシー マップの交換」（P.7-4）
- 「複数のトラフィック クラスの処理方法」（P.7-4）

使用中のインスペクション ポリシー マップの交換

サービス ポリシーですでに使用しているインスペクション ポリシー マップを交換する必要がある場合、次の方法を使用してください。

- すべてのインスペクション ポリシー マップ：使用中のインスペクション ポリシー マップを別のマップ名と交換する場合は、**inspect** *protocol map* コマンドを削除し、新しいマップを使用して再度追加します。次に例を示します。

```
hostname(config)# policy-map test
hostname(config-pmap)# class sip
hostname(config-pmap-c)# no inspect sip sip-map1
hostname(config-pmap-c)# inspect sip sip-map2
```

- HTTP インスペクション ポリシー マップ：使用中の HTTP インスペクション ポリシー マップ（**policy-map type inspect http**）を変更する場合、変更を有効にするには **inspect http** *map* アクションを削除し、再適用する必要があります。たとえば、「http-map」インスペクション ポリシー マップを変更する場合、レイヤ 3/4 ポリシーから **inspect http http-map** コマンドを削除し、再度追加する必要があります。

```
hostname(config)# policy-map test
hostname(config-pmap)# class http
hostname(config-pmap-c)# no inspect http http-map
hostname(config-pmap-c)# inspect http http-map
```

複数のトラフィック クラスの処理方法

インスペクション ポリシー マップには、複数のインスペクション クラス マップや直接照合を指定できます。

1 つのパケットが複数の異なる **match** コマンドまたは **class** コマンドと一致する場合、ASA がアクションを適用する順序は、インスペクション ポリシー マップにアクションが追加された順序ではなく、ASA の内部ルールによって決まります。内部ルールは、アプリケーションのタイプとパケット解析の論理的進捗によって決まり、ユーザが設定することはできません。HTTP トラフィックの場合、Request Method フィールドの解析が Header Host Length フィールドの解析よりも先に行われ、Request Method フィールドに対するアクションは Header Host Length フィールドに対するアクションより先に行われます。たとえば、次の match コマンドは任意の順序で入力できますが、**match request method get** コマンドが最初に照合されます。

```
match request header host length gt 100
  reset
match request method get
  log
```

アクションがパケットをドロップすると、インスペクション ポリシー マップではそれ以降のアクションは実行されません。たとえば、最初のアクションが接続のリセットである場合、それ以降の照合基準との照合は行われません。最初のアクションがパケットのログへの記録である場合、接続のリセットなどの 2 番目のアクションは実行されます。

パケットが、同じ複数の **match** コマンドまたは **class** コマンドと照合される場合は、ポリシー マップ内での順序に従って照合されます。たとえば、ヘッダーの長さが 1001 のパケットの場合は、次に示す最初のコマンドと照合されてログに記録され、それから 2 番目のコマンドと照合されてリセットされます。2 つの **match** コマンドの順序を逆にすると、2 番目の **match** コマンドとの照合前にパケットのドロップと接続のリセットが行われ、ログには記録されません。

```
match request header length gt 100
  log
match request header length gt 1000
  reset
```

クラス マップは、そのクラス マップ内で重要度が最低の **match** コマンド（重要度は、内部ルールに基づきます）に基づいて、別のクラス マップまたは **match** コマンドと同じタイプであると判断されます。クラス マップに、別のクラス マップと同じタイプの重要度が最低の **match** コマンドがある場合、それらのクラス マップはポリシー マップに追加された順序で照合されます。各クラス マップの重要度が最低の照合が異なる場合、重要度が高い **match** コマンドを持つクラス マップが最初に照合されます。たとえば、次の 3 つのクラス マップには、**match request-cmd**（高重要度）と **match filename**（低重要度）という 2 つのタイプの **match** コマンドがあります。ftp3 クラス マップには両方のコマンドが含まれていますが、最低重要度のコマンドである **match filename** に従ってランク付けされています。ftp1 クラス マップには最高重要度のコマンドがあるため、ポリシー マップ内での順序に関係なく最初に照合されます。ftp3 クラス マップは ftp2 クラス マップと同じ重要度としてランク付けされており、**match filename** コマンドも含まれています。これらのクラス マップの場合、ポリシー マップ内での順序に従い、ftp3 が照合されてから ftp2 が照合されます。

```
class-map type inspect ftp match-all ftp1
  match request-cmd get
class-map type inspect ftp match-all ftp2
  match filename regex abc
class-map type inspect ftp match-all ftp3
  match request-cmd get
  match filename regex abc

policy-map type inspect ftp ftp
  class ftp3
    log
  class ftp2
    log
  class ftp1
    log
```

アプリケーション インスペクションのガイドライン

フェールオーバーのガイドライン

インスペクションが必要なマルチメディア セッションのステート情報は、ステートフル フェールオーバーのステート リンク経由では渡されません。ステート リンク経由で複製される GTP および SIP は例外です。

IPv6 のガイドライン

IPv6 は次のインスペクションでサポートされています。

- DNS
- FTP
- HTTP
- ICMP
- SCCP（Skinny）
- SIP
- SMTP
- IPSec パススルー
- IPv6

NAT64 は次のインスペクションでサポートされています。

- DNS
- FTP
- HTTP
- ICMP

その他のガイドラインと制限事項

- 一部のインスペクション エンジンは、PAT、NAT、外部 NAT、または同一セキュリティ インターフェイス間の NAT をサポートしません。NAT サポートの詳細については、「デフォルト インスペクションと NAT に関する制限事項」（P.7-6）を参照してください。
- すべてのアプリケーション インスペクションについて、ASA はアクティブな同時データ接続の数を 200 接続に制限します。たとえば、FTP クライアントが複数のセカンダリ接続を開く場合、FTP インスペクション エンジンはアクティブな接続を 200 だけ許可して 201 番目の接続からはドロップし、適応型セキュリティ アプライアンスはシステム エラー メッセージを生成します。
- 検査対象のプロトコルは高度な TCP ステート トラッキングの対象となり、これらの接続の TCP ステートは自動的には複製されません。スタンバイ装置への接続は複製されますが、TCP ステートを再確立するベスト エフォート型の試行が行われます。
- ASA（インターフェイス）に送信される TCP/UDP トラフィックはデフォルトで検査されます。ただし、インターフェイスに送信される ICMP トラフィックは、ICMP インスペクションをイネーブルにした場合でも検査されません。したがって、ASA がバックアップ デフォルト ルートを介して到達できる送信元からエコー要求が送信された場合など、特定の状況下では、インターフェイスへの ping（エコー要求）が失敗する可能性があります。

アプリケーション インスペクションのデフォルト

次のトピックで、アプリケーション インスペクションのデフォルトの動作について説明します。

- 「デフォルト インスペクションと NAT に関する制限事項」（P.7-6）
- 「デフォルトのインスペクション ポリシー マップ」（P.7-11）

デフォルト インスペクションと NAT に関する制限事項

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、すべて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）。デフォルト アプリケーション インスペクション トラフィックには、各プロトコルのデフォルト ポートへのトラフィックが含まれます。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（標準以外のポートにインスペクションを適用する場合や、デフォルトでイネーブルになっていないインスペクションを追加する場合など）には、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する必要があります。

次の表に、サポートされているすべてのインスペクション、デフォルトのクラス マップで使用されるデフォルト ポート、およびデフォルトでオンになっているインスペクション エンジン（太字）を示します。この表には、NAT に関する制限事項も含まれています。この表の見方は次のとおりです。

- デフォルト ポートに対してデフォルトでイネーブルになっているインスペクション エンジンは太字で表記されています。
- ASA は、これらの指定された標準に準拠していますが、検査対象のパケットには準拠を強制しません。たとえば、各 FTP コマンドは特定の順序である必要がありますが、ASA によってその順序を強制されることはありません。

表 7-1　サポートされているアプリケーション インスペクション エンジン

アプリケーション	デフォルト ポート	NAT に関する制限事項	標準	注
CTIQBE	TCP/2748	拡張 PAT はサポートされません。 NAT64 なし。 （クラスタリング）スタティック PAT はサポートされません。	—	—
DCERPC	TCP/135	NAT64 なし。	—	—
DNS over UDP	UDP/53	NAT サポートは、WINS 経由の名前解決では使用できません。	RFC 1123	—
FTP	TCP/21	（クラスタリング）スタティック PAT はサポートされません。	RFC 959	—
GTP	UDP/3386 UDP/2123	拡張 PAT はサポートされません。 NAT なし。	—	特別なライセンスが必要です。
H.323 H.225 および **RAS**	TCP/1720 UDP/1718 UDP (RAS) 1718-1719	ダイナミック NAT または PAT はサポートされません。 スタティック PAT は機能しない可能性があります。 （クラスタリング）スタティック PAT はサポートされません。 拡張 PAT はサポートされません。 Per-Session PAT はサポートされません。 同一セキュリティのインターフェイス上の NAT はサポートされません。 NAT64 なし。	ITU-T H.323、H.245、H225.0、Q.931、Q.932	—

表 7-1　サポートされているアプリケーション インスペクション エンジン（続き）

アプリケーション	デフォルト ポート	NAT に関する制限事項	標準	注
HTTP	TCP/80	—	RFC 2616	ActiveX と Java を除去する場合の MTU 制限に注意してください。MTU が小さすぎて Java タグまたは ActiveX タグを 1 つのパケットに納められない場合は、除去の処理は行われません。
ICMP	—	—	—	ASA インターフェイスに送信される ICMP トラフィックは検査されません。
ICMP ERROR	—	—	—	—
ILS（LDAP）	TCP/389	拡張 PAT はサポートされません。 NAT64 なし。	—	—
Instant Messaging（IM; インスタントメッセージ）	クライアントにより異なる	拡張 PAT はサポートされません。 NAT64 なし。	RFC 3860	—
IP オプション	—	NAT64 なし。	RFC 791、RFC 2113	—
IPsec Pass Through	UDP/500	PAT はサポートされません。 NAT64 なし。	—	—
IPv6	—	NAT64 なし。	RFC 2460	—
MGCP	UDP/2427、2727	拡張 PAT はサポートされません。 NAT64 なし。 （クラスタリング）スタティック PAT はサポートされません。	RFC 2705bis-05	—
MMP	TCP 5443	拡張 PAT はサポートされません。 NAT64 なし。	—	—
NetBIOS Name Server over IP	UDP/137、138（送信元ポート）	拡張 PAT はサポートされません。 NAT64 なし。	—	NetBIOS は、NBNS UDP ポート 137 および NBDS UDP ポート 138 に対してパケットの NAT 処理を実行することでサポートされます。
PPTP	TCP/1723	NAT64 なし。 （クラスタリング）スタティック PAT はサポートされません。	RFC 2637	—
RADIUS アカウンティング	1646	NAT64 なし。	RFC 2865	—

表 7-1　サポートされているアプリケーション インスペクション エンジン（続き）

アプリケーション	デフォルト ポート	NAT に関する制限事項	標準	注
RSH	TCP/514	PAT はサポートされません。 NAT64 なし。 （クラスタリング）スタティック PAT はサポートされません。	Berkeley UNIX	—
RTSP	TCP/554	拡張 PAT はサポートされません。 NAT64 なし。 （クラスタリング）スタティック PAT はサポートされません。	RFC 2326、2327、1889	HTTP クローキングは処理しません。
ScanSafe（クラウド Web セキュリティ）	TCP/80 TCP/413	—	—	これらのポートは、ScanSafe インスペクションの default-inspection-traffic クラスには含まれません。
SIP	TCP/5060 UDP/5060	同一セキュリティのインターフェイス上の NAT はサポートされません。 拡張 PAT はサポートされません。 Per-Session PAT はサポートされません。 NAT64 または NAT46 はなし。 （クラスタリング）スタティック PAT はサポートされません。	RFC 2543	一定の条件下で、Cisco IP Phone 設定をアップロード済みの TFTP は処理しません。
SKINNY（SCCP）	TCP/2000	同一セキュリティのインターフェイス上の NAT はサポートされません。 拡張 PAT はサポートされません。 Per-Session PAT はサポートされません。 NAT64、NAT46、または NAT66 はなし。 （クラスタリング）スタティック PAT はサポートされません。	—	一定の条件下で、Cisco IP Phone 設定をアップロード済みの TFTP は処理しません。
SMTP および **ESMTP**	TCP/25	NAT64 なし。	RFC 821、1123	—

表 7-1　サポートされているアプリケーション インスペクション エンジン（続き）

アプリケーション	デフォルト ポート	NAT に関する制限事項	標準	注
SNMP	UDP/161、162	NAT および PAT はサポートされません。	RFC 1155、1157、1212、1213、1215	v.2 RFC 1902 ～ 1908、v.3 RFC 2570 ～ 2580
SQL*Net	TCP/1521	拡張 PAT はサポートされません。 NAT64 なし。 （クラスタリング）スタティック PAT はサポートされません。	—	v.1 および v.2
Sun RPC over UDP および TCP	UDP/111	拡張 PAT はサポートされません。 NAT64 なし。	—	デフォルトのルールには UDP ポート 111 が含まれています。TCP ポート 111 の Sun RPC インスペクションをイネーブルにする場合は、TCP ポート 111 を照合する新しいルールを作成し、Sun RPC インスペクションを実行する必要があります。
TFTP	UDP/69	NAT64 なし。 （クラスタリング）スタティック PAT はサポートされません。	RFC 1350	ペイロード IP アドレスは変換されません。
WAAS	TCP/1-65535	拡張 PAT はサポートされません。 NAT64 なし。	—	—
XDMCP	UDP/177	拡張 PAT はサポートされません。 NAT64 なし。 （クラスタリング）スタティック PAT はサポートされません。	—	—

デフォルト ポリシー コンフィギュレーションには、次のコマンドが含まれます。

```
class-map inspection_default
 match default-inspection-traffic
policy-map type inspect dns preset_dns_map
 parameters
    message-length maximum client auto
    message-length maximum 512
    dns-guard
    protocol-enforcement
    nat-rewrite
policy-map global_policy
 class inspection_default
  inspect dns preset_dns_map
  inspect ftp
  inspect h323 h225 _default_h323_map
  inspect h323 ras _default_h323_map
  inspect ip-options _default_ip_options_map
  inspect netbios
  inspect rsh
  inspect rtsp
  inspect skinny
  inspect esmtp _default_esmtp_map
  inspect sqlnet
  inspect sunrpc
  inspect tftp
  inspect sip
  inspect xdmcp
```

デフォルトのインスペクション ポリシー マップ

一部のインスペクション タイプは、非表示のデフォルト ポリシー マップを使用します。たとえば、マップを指定しないで ESMTP インスペクションをイネーブルにした場合、_default_esmtp_map が使用されます。

デフォルトのインスペクションは、各インスペクション タイプについて説明しているセクションで説明されています。これらのデフォルト マップは、**show running-config all policy-map** コマンドを使用して表示できます。

DNS インスペクションは、明示的に設定されたデフォルト マップ preset_dns_map を使用する唯一のインスペクションです。

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

サービス ポリシーにアプリケーション インスペクションを設定します。サービス ポリシーでは、一貫性と柔軟性を備えた方法で ASA 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。アプリケーションによっては、インスペクションをイネーブルにすると特別なアクションを実行できるものがあります。サービス ポリシーに関する一般的な情報については、第 1 章「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」を参照してください。

一部のアプリケーションでは、デフォルトでインスペクションがイネーブルになっています。詳細については、「デフォルト インスペクションと NAT に関する制限事項」（P.7-6）を参照してください。この項を参照してインスペクション ポリシーを変更してください。

手順

ステップ 1　既存のクラス マップにインスペクションを追加しようとしている場合を除いて、通過トラフィックまたは管理トラフィック向けのレイヤ 3/4 クラス マップでインスペクションを適用したいトラフィックを指定します。

詳細については、「通過トラフィック用のレイヤ 3/4 クラス マップの作成」（P.1-14）および「管理トラフィック用のレイヤ 3/4 クラス マップの作成」（P.1-16）を参照してください。管理レイヤ 3/4 クラス マップは、RADIUS アカウンティングのインスペクションだけで使用できます。

選択するクラス マップに関する重要な関連事項があります。inspection_default クラスにのみ複数のインスペクションを設定できます。また、デフォルトのインスペクションを適用する既存のグローバル ポリシーを編集するだけの場合もあります。選択するクラス マップに関する詳細情報については、「インスペクションの適切なトラフィック クラスの選択」（P.7-17）を参照してください。

ステップ 2　（任意）一部のインスペクション エンジンでは、トラフィックにインスペクションを適用するときの追加パラメータを制御できます。この手順の後半の表に、インスペクション ポリシー マップを使用できるプロトコルを示します。また、それらの設定手順へのポインタも記載しています。

ステップ 3　クラス マップ トラフィックで実行するアクションを設定するレイヤ 3/4 ポリシー マップを追加または編集します。

```
hostname(config)# policy-map name
hostname(config-pmap)#
```

デフォルトのポリシー マップの名前は「global_policy」です。このポリシー マップには、「デフォルト インスペクションと NAT に関する制限事項」（P.7-6）で示されているデフォルトのインスペクションが含まれています。デフォルトのポリシーを変更する場合（インスペクションを追加または削除する場合や、追加のクラス マップを特定してアクションを割り当てる場合など）は、**global_policy** を名前として入力します。

ステップ 4　アクションを割り当てたいクラス マップを指定します。

```
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

デフォルトのポリシー マップを編集する場合、デフォルトのポリシー マップには inspection_default クラス マップが含まれています。このクラスのアクションを編集する場合は、**inspection_default** を名前として入力します。このポリシー マップに別のクラス マップを追加する場合は、異なる名前を指定してください。

必要に応じて同じポリシー内に複数のクラス マップを組み合わせることができるため、照合するトラフィックに応じたクラス マップを作成することができます。ただし、トラフィックがインスペクション コマンドを含むクラス マップと一致し、その後同様にインスペクション コマンドを含む別のクラス マップとも一致した場合、最初に一致したクラスだけが使用されます。たとえば、SNMP では inspection_default クラス マップを照合します。SNMP インスペクションをイネーブルにするには、デフォルト クラスの SNMP インスペクションをイネーブルにします。SNMP を照合する他のクラスを追加しないでください。

ステップ 5　アプリケーション インスペクションをイネーブルにします。

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

protocol には、次のいずれかの値を指定します。

表 7-2　　protocol のキーワード

キーワード	注
ctiqbe	「CTIQBE インスペクション」（P.9-1）を参照してください。
dcerpc [*map_name*]	「DCERPC インスペクション」（P.11-1）を参照してください。 「DCERPC インスペクション ポリシー マップの設定」（P.11-2）に従って DCERPC インスペクション ポリシー マップを追加した場合は、このコマンドでマップ名を特定します。
dns [*map_name*] [**dynamic-filter-snoop**]	「DNS インスペクション」（P.8-1）を参照してください。 「DNS インスペクション ポリシー マップの設定」（P.8-3）に従って DNS インスペクション ポリシー マップを追加した場合は、このコマンドでマップ名を特定します。デフォルトの DNS インスペクション ポリシー マップの名前は「preset_dns_map」です。 ボットネット トラフィック フィルタの DNS スヌーピングをイネーブルにするには、**dynamic-filter-snoop** キーワードを入力します。
esmtp [*map_name*]	「SMTP および拡張 SMTP インスペクション」（P.8-44）を参照してください。 「ESMTP インスペクション ポリシー マップの設定」（P.8-46）に従って ESMTP インスペクション ポリシー マップを追加した場合は、このコマンドでマップ名を特定します。
ftp [**strict** [*map_name*]]	「FTP インスペクション」（P.8-9）を参照してください。 **strict** キーワードを使用して、Web ブラウザが FTP 要求内の埋め込みコマンドを送信できないようにすることで、保護されたネットワークのセキュリティを強化できます。詳細については、「厳密な FTP」（P.8-10）を参照してください。 「FTP インスペクション ポリシー マップの設定」（P.8-11）に従って FTP インスペクション ポリシー マップを追加した場合は、このコマンドでマップ名を特定します。
gtp [*map_name*]	「GTP インスペクション」（P.11-5）を参照してください。 「GTP インスペクション ポリシー マップの設定」（P.11-7）に従って GTP インスペクション ポリシー マップを追加した場合は、このコマンドでマップ名を特定します。

表 7-2　protocol のキーワード（続き）

キーワード	注
h323 h225 [*map_name*]	「H.323 インスペクション」（P.9-3）を参照してください。 「H.323 インスペクション ポリシー マップの設定」（P.9-7）に従って H323 インスペクション ポリシー マップを追加した場合は、このコマンドでマップ名を特定します。
h323 ras [*map_name*]	「H.323 インスペクション」（P.9-3）を参照してください。 「H.323 インスペクション ポリシー マップの設定」（P.9-7）に従って H323 インスペクション ポリシー マップを追加した場合は、このコマンドでマップ名を特定します。
http [*map_name*]	「HTTP インスペクション」（P.8-16）を参照してください。 「HTTP インスペクション ポリシー マップの設定」（P.8-17）に従って HTTP インスペクション ポリシー マップを追加した場合は、このコマンドでマップ名を特定します。
icmp	「ICMP インスペクション」（P.8-23）を参照してください。
icmp error	「ICMP エラー インスペクション」（P.8-23）を参照してください。
ils	「ILS インスペクション」（P.10-1）を参照してください。
im [*map_name*]	「インスタント メッセージ インスペクション」（P.8-24）を参照してください。 「インスタント メッセージ インスペクション ポリシー マップの設定」（P.8-24）に従ってインスタント メッセージ インスペクション ポリシー マップを追加した場合は、このコマンドでマップ名を特定します。
ip-options [*map_name*]	「IP オプション インスペクション」（P.8-29）を参照してください。 「IP オプション インスペクション ポリシー マップの設定」（P.8-31）に従って IP オプション インスペクション ポリシー マップを追加した場合は、このコマンドでマップ名を特定します。
ipsec-pass-thru [*map_name*]	「IPsec パススルー インスペクション」（P.8-33）を参照してください。 「IPsec パススルー インスペクション」（P.8-33）に従って IPsec パススルー インスペクション ポリシー マップを追加した場合は、このコマンドでマップ名を特定します。

表 7-2　protocol のキーワード（続き）

キーワード	注
ipv6 [*map_name*]	「IPv6 インスペクション」（P.8-36）を参照してください。 「IPv6 インスペクション ポリシー マップの設定」（P.8-37）に従って IPv6 インスペクション ポリシー マップを追加した場合は、このコマンドでマップ名を特定します。
mgcp [*map_name*]	「MGCP インスペクション」（P.9-13）を参照してください。 「インスペクション制御を追加するための MGCP インスペクション ポリシー マップの設定」（P.9-15）に従って MGCP インスペクション ポリシー マップを追加した場合は、このコマンドでマップ名を特定します。
netbios [*map_name*]	「NetBIOS インスペクション」（P.8-40）を参照してください。 「インスペクション制御を追加するための NetBIOS インスペクション ポリシー マップの設定」（P.8-41）に従って NetBIOS インスペクション ポリシー マップを追加した場合は、このコマンドでマップ名を特定します。
pptp	「PPTP インスペクション」（P.8-43）を参照してください。
radius-accounting *map_name*	「RADIUS アカウンティング インスペクション」（P.11-13）を参照してください。 **radius-accounting** キーワードは、管理クラス マップだけで使用できます。RADIUS アカウンティング インスペクション ポリシー マップを指定する必要があります。「RADIUS アカウンティング インスペクション ポリシー マップの設定」（P.11-14）を参照してください。
rsh	「RSH インスペクション」（P.11-17）を参照してください。
rtsp [*map_name*]	「RTSP インスペクション」（P.9-19）を参照してください。 「RTSP インスペクション ポリシー マップの設定」（P.9-21）に従って RTSP インスペクション ポリシー マップを追加した場合は、このコマンドでマップ名を特定します。
scansafe [*map_name*] [**fail-open** | **fail-closed**]	ScanSafe（クラウド Web セキュリティ）をイネーブルにしたい場合、この手順ではなく、「クラウド Web セキュリティにトラフィックを送信するサービス ポリシーの方法」（P.15-11）で説明している手順を使用してください。前述の手順では、ポリシー インスペクション マップの設定方法を含む、完全なポリシー設定について説明しています。

表 7-2　protocol のキーワード（続き）

キーワード	注
sip [*map_name*] [**tls-proxy** *proxy_name*]	「SIP インスペクション」（P.9-25）を参照してください。 「SIP インスペクション ポリシー マップの設定」（P.9-28）に従って SIP インスペクション ポリシー マップを追加した場合は、このコマンドでマップ名を特定します。暗号化されたトラフィックのインスペクションをイネーブルにするには、TLS プロキシを指定します。
skinny [*map_name*] [**tls-proxy** *proxy_name*]	「Skinny（SCCP）検査」（P.9-34）を参照してください。 「インスペクション制御を追加するための Skinny（SCCP）インスペクション ポリシー マップの設定」（P.9-36）に従って Skinny インスペクション ポリシー マップを追加した場合は、このコマンドでマップ名を特定します。暗号化されたトラフィックのインスペクションをイネーブルにするには、TLS プロキシを指定します。
snmp [*map_name*]	「SNMP インスペクション」（P.11-17）を参照してください。 SNMP インスペクション ポリシー マップを追加した場合は、このコマンドでマップ名を特定します。
sqlnet	「SQL*Net インスペクション」（P.10-2）を参照してください。
sunrpc	「Sun RPC インスペクション」（P.10-3）を参照してください。 デフォルトのクラス マップには UDP ポート 111 が含まれています。TCP ポート 111 の Sun RPC インスペクションをイネーブルにするには、TCP ポート 111 を照合する新しいクラス マップを作成し、クラスをポリシーに追加してから、そのクラスに **inspect sunrpc** コマンドを適用する必要があります。
tftp	「TFTP インスペクション」（P.8-50）を参照してください。
waas	TCP オプション 33 解析をイネーブルにします。Cisco Wide Area Application Services 製品を導入するときに使用します。
xdmcp	「XDMCP インスペクション」（P.11-19）を参照してください。

（注） 別のインスペクション ポリシー マップを使用するためにデフォルト グローバル ポリシー（または使用中のポリシー）を編集する場合、**no inspect** *protocol* コマンドを使用して古いインスペクションを削除し、新しいインスペクション ポリシー マップ名でインスペクションを再度追加する必要があります。

ステップ 6　1 つ以上のインターフェイスでポリシー マップをアクティブにするには、次のコマンドを入力します。

```
hostname(config)# service-policy policymap_name {global | interface interface_name}
```

ここで、**global** はポリシー マップをすべてのインターフェイスに適用し、**interface** は 1 つのインターフェイスに適用します。デフォルトでは、デフォルト ポリシー マップの「global_policy」は全体的に適用されます。グローバル ポリシーは 1 つしか適用できません。インターフェイスのグローバル ポリシーは、そのインターフェイスにサービス ポリシーを適用することで上書きできます。各インターフェイスには、ポリシー マップを 1 つだけ適用できます。

インスペクションの適切なトラフィック クラスの選択

通過トラフィックのデフォルトのレイヤ 3/4 クラス マップの名前は「inspection_default」です。このクラス マップは、特殊な **match** コマンド（**match default-inspection-traffic**）を使用して、トラフィックを各アプリケーション プロトコルのデフォルト ポートと照合します。このトラフィック クラスは（インスペクションには通常使用されない **match any** とともに）、IPv6 をサポートするインスペクションについて IPv4 および IPv6 トラフィックの両方を照合します。IPv6 がイネーブルなインスペクションのリストについては、「アプリケーション インスペクションのガイドライン」（P.7-5）を参照してください。

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、照合するトラフィックを特定の IP アドレスに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が指定されるため、ACL のポートはすべて無視されます。

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。

標準以外のポートを照合する場合は、標準以外のポート用に新しいクラス マップを作成してください。各インスペクション エンジンの標準ポートについては、「デフォルト インスペクションと NAT に関する制限事項」（P.7-6）を参照してください。必要に応じて同じポリシー内に複数のクラス マップを組み合わせることができるため、照合するトラフィックに応じたクラス マップを作成することができます。ただし、トラフィックがインスペクション コマンドを含むクラス マップと一致し、その後同様にインスペクション コマンドを含む別のクラス マップとも一致した場合、最初に一致したクラスだけが使用されます。たとえば、SNMP では inspection_default クラスを照合します。SNMP インスペクションをイネーブルにするには、デフォルト クラスの SNMP インスペクションをイネーブルにします。SNMP を照合する他のクラスを追加しないでください。

たとえば、デフォルトのクラス マップを使用して、インスペクションを 10.1.1.0 から 192.168.1.0 へのトラフィックに限定するには、次のコマンドを入力します。

```
hostname(config)# access-list inspect extended permit ip 10.1.1.0 255.255.255.0
192.168.1.0 255.255.255.0
hostname(config)# class-map inspection_default
hostname(config-cmap)# match access-list inspect
```

次のコマンドを使用して、クラス マップ全体を表示します。

```
hostname(config-cmap)# show running-config class-map inspection_default
!
class-map inspection_default
 match default-inspection-traffic
 match access-list inspect
!
```

ポート 21 とポート 1056（標準以外のポート）の FTP トラフィックを検査するには、それらのポートを指定する ACL を作成し、新しいクラス マップに割り当てます。

```
hostname(config)# access-list ftp_inspect extended permit tcp any any eq 21
hostname(config)# access-list ftp_inspect extended permit tcp any any eq 1056
hostname(config)# class-map new_inspection
hostname(config-cmap)# match access-list ftp_inspect
```

正規表現の設定

正規表現は、テキスト文字列のパターン照合を定義します。一部のプロトコル インスペクション マップでは、正規表現を使用して、URL や特定のヘッダー フィールドのコンテンツなどの文字列に基づいてパケットを照合できます。

- 「正規表現の作成」（P.7-18）
- 「正規表現クラス マップの作成」（P.7-21）

正規表現の作成

正規表現は、ストリングそのものとしてテキスト ストリングと文字どおりに照合することも、*メタ文字*を使用してテキスト ストリングの複数のバリアントと照合することもできます。正規表現を使用して特定のアプリケーション トラフィックの内容と照合できます。たとえば、HTTP パケット内部の URL 文字列と照合できます。

はじめる前に

Ctrl キーを押した状態で **V** キーを押すと、CLI において、疑問符（?）やタブなどの特殊文字をすべてエスケープできます。たとえば、コンフィギュレーションで **d?g** と入力するには、**d[Ctrl+V]?g** とキー入力します。

正規表現をパケットと照合する場合のパフォーマンスへの影響については、コマンド リファレンスの **regex** コマンドを参照してください。一般的に、長い入力文字列と照合したり、多くの正規表現と照合しようとすると、システム パフォーマンスが低下します。

（注） 最適化のために、ASA では、難読化解除された URL が検索されます。難読化解除では、複数のスラッシュ（/）が単一のスラッシュに圧縮されます。通常、「http://」のようなダブル スラッシュが使用される文字列では、代わりに「http:/」を検索してください。

次の表に、特別な意味を持つメタ文字を示します。

表 7-3　正規表現のメタ文字

文字	説明	注
.	ドット	任意の単一文字と一致します。たとえば、**d.g** は、dog、dag、dtg、およびこれらの文字を含む任意の単語（doggonnit など）に一致します。
(*exp*)	サブ表現	サブ表現は、文字を周囲の文字から分離して、サブ表現に他のメタ文字を使用できるようにします。たとえば、**d(o|a)g** は dog および dag に一致しますが、**do|ag** は do および ag に一致します。また、サブ表現を繰り返し限定作用素とともに使用して、繰り返す文字を区別できます。たとえば、**ab(xy){3}z** は、abxyxyxyz に一致します。
|	代替	このメタ文字によって区切られている複数の表現のいずれかと一致します。たとえば、**dog|cat** は、dog または cat に一致します。
?	疑問符	直前の表現が 0 または 1 個存在することを示す修飾子。たとえば、**lo?se** は、lse または lose に一致します。
*	アスタリスク	直前の表現が 0、1、または任意の個数存在することを示す修飾子。たとえば、**lo*se** は、lse、lose、loose などに一致します。
+	プラス	直前の表現が少なくとも 1 個存在することを示す修飾子。たとえば、**lo+se** は、lose および loose に一致しますが、lse には一致しません。
{*x*} または {*x*,}	最小繰り返し限定作用素	少なくとも *x* 回繰り返します。たとえば、**ab(xy){2,}z** は、abxyxyz や abxyxyxyz などに一致します。
[*abc*]	文字クラス	カッコ内の任意の文字と一致します。たとえば、**[abc]** は、a、b、または c に一致します。
[^*abc*]	否定文字クラス	角カッコに含まれていない単一文字と一致します。たとえば、**[^abc]** は、a、b、c 以外の任意の文字に一致します。**[^A-Z]** は、大文字のアルファベット文字以外の任意の単一の文字に一致します。
[*a-c*]	文字範囲クラス	範囲内の任意の文字と一致します。**[a-z]** は、任意の小文字のアルファベット文字に一致します。文字と範囲を組み合わせて使用することもできます。**[abcq-z]** および **[a-cq-z]** は、a、b、c、q、r、s、t、u、v、w、x、y、z に一致します。 ダッシュ（-）文字は、角カッコ内の最初の文字または最後の文字である場合にのみリテラルとなります（**[abc-]** や **[-abc]**）。
“”	引用符	文字列の末尾または先頭のスペースを保持します。たとえば、**" test"** は、一致を検索する場合に先頭のスペースを保持します。
^	キャレット	行の先頭を指定します。
\	エスケープ文字	メタ文字とともに使用すると、リテラル文字と一致します。たとえば、**\[** は左角カッコに一致します。

表 7-3　正規表現のメタ文字（続き）

文字	説明	注
char	文字	文字がメタ文字でない場合は、リテラル文字と一致します。
\r	復帰	復帰 0x0d と一致します。
\n	改行	改行 0x0a と一致します。
\t	タブ	タブ 0x09 と一致します。
\f	改ページ	フォーム フィード 0x0c と一致します。
\x*NN*	エスケープされた 16 進数	16 進数（厳密に 2 桁）を使用した ASCII 文字と一致します。
NNN	エスケープされた 8 進数	8 進数（厳密に 3 桁）としての ASCII 文字と一致します。たとえば、文字 040 はスペースを表します。

手順

ステップ 1　正規表現が一致すべきものと一致するかどうかをテストします。

```
hostname(config)# test regex input_text regular_expression
```

input_text 引数は、正規表現を使用して照合する、長さが最大で 201 文字の文字列です。

regular_expression 引数の長さは、最大 100 文字です。

Ctrl+V を使用して、CLI の特殊文字をすべてエスケープします。たとえば、**test regex** コマンドの入力文字にタブを入力するには、**test regex "test[Ctrl+V Tab]" "test\t"** と入力する必要があります。

正規表現が入力テキストと一致する場合は、次のメッセージが表示されます。

```
INFO: Regular expression match succeeded.
```

正規表現が入力テキストと一致しない場合は、次のメッセージが表示されます。

```
INFO: Regular expression match failed.
```

ステップ 2　テスト後に正規表現を追加するには、次のコマンドを入力します。

```
hostname(config)# regex name regular_expression
```

name 引数の長さは、最大 40 文字です。

regular_expression 引数の長さは、最大 100 文字です。

例

次に、インスペクション ポリシー マップで使用する 2 つの正規表現を作成する例を示します。

```
hostname(config)# regex url_example example\.com
hostname(config)# regex url_example2 example2\.com
```

正規表現クラス マップの作成

正規表現クラス マップは、1 つ以上の正規表現を特定します。正規表現クラス マップは、正規表現オブジェクトを集めているにすぎません。多くの場合、正規表現オブジェクトの代わりに正規表現クラス マップを使用できます。

手順

ステップ 1　正規表現クラス マップを作成します。

```
hostname(config)# class-map type regex match-any class_map_name
hostname(config-cmap)#
```

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。

match-any キーワードにより、トラフィックが少なくとも 1 つの正規表現と一致する場合には、そのトラフィックがクラス マップと一致するように指定します。

ステップ 2　（任意）クラス マップに説明を追加します。

```
hostname(config-cmap)# description string
```

ステップ 3　正規表現ごとに次のコマンドを入力して、クラス マップに含める正規表現を指定します。

```
hostname(config-cmap)# match regex regex_name
```

例

次に、2 つの正規表現を作成し、これを正規表現クラス マップに追加する例を示します。文字列「example.com」または「example2.com」が含まれる場合は、トラフィックはクラス マップに一致します。

```
hostname(config)# regex url_example example\.com
hostname(config)# regex url_example2 example2\.com
hostname(config)# class-map type regex match-any URLs
hostname(config-cmap)# match regex url_example
hostname(config-cmap)# match regex url_example2
```

アプリケーション インスペクションの履歴

機能名	リリース	説明
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。