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。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。

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

サービス ポリシーのポリシー マップでインスペクションが有効になっている場合、ポリシー マップの交換は2つのステップからなるプロセスです。最初に、インスペクションを削除する必要があります。次に、新しいポリシー マップ名でそれを再度追加します。

たとえば、SIP インスペクションで sip-map1 を sip-map2 と交換するには、次のコマンド シーケンスを使用します。

```
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
```

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クラスまたはダイレクト マッチに一致する場合、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、同一の複数の一致基準と照合される場合は、ポリシーマップ内のそれらのコマンド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 オプション（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 マップまたはダイレクト マッチ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オプションがある場合、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 オプション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

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、M3UA、および SIP は例外です。ステートフル フェールオーバーを取得するために、M3UA インスペクションで厳密なアプリケーション サーバ プロセス（ASP）のステート チェックを設定する必要があります。

クラスタ

次のインスペクションはクラスタリングではサポートされていません。

- CTIQBE
- H323、H225、および RAS
- IPsec パススルー
- MGCP
- MMP
- RTSP

- SCCP（Skinny）
- WAAS

IPv6

IPv6 は次のインスペクションでサポートされています。

- Diameter
- DNS over UDP
- FTP
- GTP
- HTTP
- ICMP
- IPSec パススルー
- IPv6
- M3UA
- SCCP（Skinny）
- SCTP
- SIP
- SMTP
- VXLAN

NAT64 は次のインスペクションでサポートされています。

- DNS over UDP
- FTP
- HTTP
- ICMP
- SCTP

その他のガイドライン

- 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 （6 ページ）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。
- TCP接続にインスペクションが必要であるとシステムが判断した場合、システムはそれらのインスペクションの前に、パケット上でMSSおよび選択的確認応答（SACK）オプションを除き、すべての TCP オプションをクリアします。その他のオプションは、接続に適用されている TCP マップで許可されているとしてもクリアされます。
- 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

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 *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 プロトコル、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。	—	—
Diameter	TCP/3868 TCP/5868（TCP/TLS 用） SCTP/3868	NAT/PAT なし。	RFC 6733	キャリア ライセンスが必要です。
DNS over UDP DNS over TCP	UDP/53 TCP/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	DNS over TCP を検査するには、DNS インスペクション ポリシーマップで DNS/TCP インスペクションを有効にする必要があります。
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（GTPv0） UDP/2123（GTPv1+）	拡張 PAT はサポートされません。 NAT なし。	—	キャリア 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	（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。 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	—

アプリケーション	デフォルト プロトコル、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	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	ICMP	—	—	—
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	RSVP	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	—
LISP	—	NAT および PAT はサポートされません。	—	—
M3UA	SCTP/2905	埋め込まれたアドレスに対する NAT または PAT はなし。	RFC 4666	キャリア ライセンスが必要です。
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	—

アプリケーション	デフォルト プロトコル、ポート	NAT に関する制限事項	標準	注
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	UDP/1646	NAT64 なし。	RFC 2865	—
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/443	—	—	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。
SCTP	SCTP	—	RFC 4960	キャリア ライセンスが必要です。 SCTP トラフィックでスタティック ネットワーク オブジェクト NAT を実行できますが（ダイナミック NAT/PAT なし）、インスペクション エンジンは NAT には使用されません。

アプリケーション	デフォルト プロトコル、ポート	NAT に関する制限事項	標準	注
SIP	TCP/5060 UDP/5060	セキュリティ レベルが同じインターフェイス、または低セキュリティ レベルから高セキュリティ レベルに至るインターフェイス上の NAT/PAT はサポートされません。 拡張 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。 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	—
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
STUN	TCP/3478 UDP/3478	（WebRTC）スタティック NAT/PAT44 のみ。 （Cisco Spark）スタティック NAT/PAT44 と 64、およびダイナミック NAT/PAT。	RFC 5245、5389	—

アプリケーション	デフォルト プロトコル、ポート	NAT に関する制限事項	標準	注
Sun RPC	TCP/111 UDP/111	拡張 PAT はサポートされません。 NAT64 なし。	—	—
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。	—	—
VXLAN	UDP/4789	N/A	RFC 7348	Virtual Extensible Local Area Network。

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。

インスペクションは、一部のアプリケーションの標準のポートとプロトコルに関しては、デフォルトですべてのインターフェイスでグローバルに有効になっています。デフォルトのインスペクションの詳細については、デフォルト インスペクションと NAT に関する制限事項 （6 ページ） を参照してください。インスペクションの設定をカスタマイズする一般的な方法は、デフォルトのグローバル ポリシーをカスタマイズすることです。または、たとえばインターフェイス固有のポリシーなど、必要に応じて新しいサービス ポリシーを作成することもできます。

始める前に

一部のアプリケーションでは、インスペクション ポリシー マップを設定することで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。これらの拡張機能を設定する場合は、インスペクションを設定する前にマップを作成します。

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する場合を除き、L3/L4 クラスマップを作成して、インスペクションを適用するトラフィックを識別します。

```
class-map name
match parameter
```

例：

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

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

デフォルト グローバル ポリシーの inspection_default クラス マップは、すべてのインスペクション タイプのデフォルト ポートを含む特別なクラス マップです（**match default-inspection-traffic**）。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。このマップをデフォルト ポリシーまたは新しいサービス ポリシーで使用する場合は、このステップを省略できます。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 （19 ページ）を参照してください。

照合ステートメントについては、通過トラフィック用のレイヤ 3/4 クラス マップの作成を参照してください。管理レイヤ 3/4 クラスを使用する RADIUS アカウンティング インスペクションの場合は、RADIUS アカウンティング インスペクションの設定 を参照してください。

ステップ 2 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：**policy-map** *name*

例：

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

デフォルト設定では、global_policy ポリシー マップはすべてのインターフェイスにグローバルに割り当てられます。global_policy を編集する場合は、ポリシー名として global_policy を入力します。

ステップ 3 インスペクションに使用する L3/L4 クラス マップを指定します：**class** *name*

例：

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

デフォルト ポリシーを編集する場合、または新しいポリシーで特別な inspection_default クラス マップを使用する場合は、*name* として **inspection_default** を指定します。それ以外の場合は、この手順ですでに作成したクラスを指定します。

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 4 アプリケーション インスペクションを有効にします：**inspect** *protocol*

protocol には、次のいずれかの値を指定します。

表 *2:* インスペクション プロトコル キーワード

キーワード	注記
ctiqbe	CTIQBE インスペクションを参照してください。
dcerpc [*map_name*]	DCERPC インスペクション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。
diameter [*map_name*][**tls-proxy** *proxy_name*]	Diameter インスペクションを参照してください。 Diameter インスペクション ポリシー マップの設定 に従って Diameter 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。 **tls-proxy** *proxy_name* には、このインスペクションに使用する TLS プロキシを指定します。TLS プロキシは、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場合にのみ必要です。
dns [*map_name*] [**dynamic-filter-snoop**]	DNS インスペクション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。 **dynamic-filter-snoop** は、ボットネット トラフィック フィルタによってのみ使用される動的フィルタのスヌーピングをイネーブルにします。ボットネット トラフィック フィルタリングを使用する場合に限り、このキーワードを指定します。DNS スヌーピングは、外部 DNS 要求が送信されるインターフェイスでだけイネーブルにすることを推奨します。すべての UDP DNS トラフィック（内部 DNS サーバへの送信トラフィックを含む）に対して DNS スヌーピングをイネーブルにすると、ASA で不要な負荷が発生します。
esmtp [*map_name*]	SMTP および拡張 SMTP インスペクション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。

キーワード	注記
ftp [**strict** [*map_name*]]	FTP インスペクション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、「Strict FTP」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 インスペクションの概要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。
h323 h225 [*map_name*]	H.323 インスペクション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 インスペクションを参照してください。
icmp error	ICMP エラー インスペクションを参照してください。
ils	ILS インスペクション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。

キーワード	注記
ip-options [*map_name*]	IP オプション インスペクション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 インスペクションを参照してください。 IPsec パススルー インスペクション ポリシー マップの設定に従って 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。
ipv6 [*map_name*]	IPv6 インスペクション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。
lisp [*map_name*]	インスペクションなどの LISP を設定する詳細については、全般設定ガイドのクラスタリングの章を参照してください。 LISP 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。
m3ua [*map_name*]	M3UA インスペクションを参照してください。 M3UA インスペクション ポリシー マップの設定 に従って M3UA 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 インスペクションを参照してください。 MGCP インスペクション ポリシー マップの設定に従って 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 インスペクションを参照してください。 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 インスペクションを参照してください。

キーワード	注記
radius-accounting *map_name*	RADIUS アカウンティング インスペクションの概要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 マップの設定を参照してください。
rsh	RSH インスペクションを参照してください。
rtsp [*map_name*]	RTSP インスペクション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 ポリシーの設定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。
sctp [*map_name*]	SCTP アプリケーション レイヤのインスペクションを参照してください。 SCTP インスペクション ポリシーマップの設定 に従ってSCTP 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。
sip [*map_name*] [**tls-proxy** *proxy_name*]	SIP インスペクション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。 **tls-proxy** *proxy_name* には、このインスペクションに使用する TLS プロキシを指定します。TLS プロキシは、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場合にのみ必要です。

キーワード	注記
skinny [*map_name*] [**tls-proxy** *proxy_name*]	Skinny（SCCP）インスペクションを参照してください。 Skinny（SCCP）インスペクション ポリシー マップの設定に従って 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。 **tls-proxy** *proxy_name* には、このインスペクションに使用する TLS プロキシを指定します。TLS プロキシは、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場合にのみ必要です。
snmp [*map_name*]	SNMP Inspection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 インスペクションを参照してください。
stun	STUN インスペクションを参照してください。
sunrpc	Sun RPC インスペクション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 インスペクションを参照してください。
vxlan	VXLAN インスペクション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。

例：

```
hostname(config-class)# no inspect sip
hostname(config-class)# inspect sip sip-map
```

ステップ 5 既存のサービス ポリシー（たとえば、global_policy という名前のデフォルト グローバル ポリシー）を編集している場合は、以上で終了です。それ以外の場合は、1 つまたは複数のインターフェイスでポリシー マップをアクティブにします。

service-policy *policymap_name* {**global** | **interface** *interface_name*}

例：

```
hostname(config)# service-policy global_policy global
```

global キーワードはポリシー マップをすべてのインターフェイスに適用し、**interface** はポリシーを 1 つのインターフェイスに適用します。グローバル ポリシーは 1 つしか許可されません。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のプロトコルおよび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 （4 ページ）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 （6 ページ）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

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。

始める前に

疑問符（?）やタブなど、CLIの特殊文字をすべてエスケープするには、Ctrl+Vを使用します。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で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 *3:* 正規表現のメタ文字

文字	説明	注意
.	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。

文字	説明	注意
[^*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、大文字以外の任意の 1 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。
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：**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。**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：**class-map type regex match-any** *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。

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）説明をクラス マップに追加します：**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：**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。トラフィックに文字列「example.com」または「example2.com」が含まれる場合、トラフィックはクラス マップと一致します。

```
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
```

インスペクション ポリシーのモニタリング

インスペクション サービス ポリシーをモニタするには、次のコマンドを入力します。構文の詳細と例については、Cisco.com のコマンド リファレンスを参照してください。

- **show service-policy inspect** *protocol*

  インスペクション サービス ポリシーの統計情報を表示します。protocol 、dns**dns** どの inspect コマンドからのプロトコルです。ただし、すべてのインスペクション プロトコルでこのコマンドを使用して統計情報が表示されるわけではありません。次に例を示します。

```
asa# show service-policy inspect dns

Global policy:
  Service-policy: global_policy
    Class-map: inspection_default
      Inspect: dns preset_dns_map, packet 0, lock fail 0, drop 0, reset-drop 0,
5-min-pkt-rate 0 pkts/sec, v6-fail-close 0
        message-length maximum client auto, drop 0
        message-length maximum 512, drop 0
        dns-guard, count 0
        protocol-enforcement, drop 0
        nat-rewrite, count 0
asa#
```

- **show conn**

  デバイスを通過するトラフィックの現在の接続を示します。さまざまなプロトコルに関する情報を取得できるように、このコマンドにはさまざまなキーワードがあります。

- 特定の検査対象プロトコルの追加コマンドは次のとおりです。

  - **show ctiqbe**

    CTIQBEインスペクションエンジンによって割り当てられたメディア接続に関する情報を表示します。

  - **show h225**

    H.225 セッションの情報を表示します。

  - **show h245**

    スロースタートを使用しているエンドポイントによって確立されたH.245セッションの情報を表示します。

  - **show h323 ras**

    ゲートキーパーとその H.323 エンドポイントの間に確立されている H.323 RAS セッションの接続情報を表示します。

  - **show mgcp** {**commands** | **sessions** }

    コマンド キュー内の MGCP コマンドの数、または既存の MGCP セッションの数を表示します。

  - **show sip**

    SIP セッションの情報を表示します。

  - **show skinny**

    Skinny（SCCP）セッションに関する情報を表示します。

  - **show sunrpc-server active**

    Sun RPC サービス用に開けられているピンホールを表示します。

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。

機能名	リリース	説明
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。