CHAPTER 47

基本インターネット プロトコルのインスペクションの設定

この章では、アプリケーション レイヤ プロトコル インスペクションを設定する方法について説明します。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。

この章は、次の項で構成されています。

DNS インスペクション

この項では、DNS アプリケーション インスペクションについて説明します。この項は、次の内容で構成されています。

DNS アプリケーション インスペクションの動作

ASA で DNS 応答が転送されるとすぐに、ASA は DNS クエリーに関連付けられた DNS セッションを切断します。ASA はまた、メッセージ交換をモニタして DNS 応答の ID が DNS クエリーの ID と一致することを確認します。

DNS インスペクションをイネーブルにすると（デフォルト）、ASA は次の追加のタスクを実行します。

- **alias**、**static**、および **nat** コマンドを使用して作成されたコンフィギュレーションに基づいて、DNS レコードを変換します（DNS リライト）。変換は、DNS 応答の A レコードだけに適用されるため、DNS リライトによって PTR レコードを必要とする逆ルックアップが影響を受けることはありません。

（注） 1 つの A レコードには複数の PAT ルールが適用可能で、使用する PAT ルールがあいまいなため、DNS リライトは PAT には適用できません。

- 最大 DNS メッセージ長を指定します（デフォルトは 512 バイト、最大長は 65535 バイト）。ASA は必要に応じてリアセンブリを実行し、パケット長が設定されている最大長よりも短いことを確認します。ASA は、最大長を超えるパケットをドロップします。

（注） **maximum-length** オプションを指定せずに **inspect dns** コマンドを入力した場合、DNS パケット サイズはチェックされません。

- ドメイン名の長さを 255 バイトに制限し、ラベルの長さを 63 バイトに制限します。
- DNS メッセージに圧縮ポインタが出現した場合、ポインタが参照するドメイン名の整合性を確認します。
- 圧縮ポインタのループが終了するかどうかを確認します。

複数の DNS セッションが同じ 2 つのホスト間で発生し、それらのセッションの 5 つのタプル（送信元/宛先 IP アドレス、送信元/宛先ポート、およびプロトコル）が同じものである場合、それらのセッションに対しては接続が 1 つだけ作成されます。DNS ID は *app_id* で追跡され、各 app_id のアイドル タイマーは独立して実行されます。

app_id の有効期限はそれぞれ独立して満了するため、正当な DNS 応答が ASA を通過できるのは、限られた期間内だけであり、リソースの継続使用はできません。ただし、**show conn** コマンドを入力した場合、新しい DNS セッションによってリセットされている DNS 接続のアイドル タイマーが表示されます。これは共有 DNS 接続の性質によるものであり、仕様です。

DNS リライトの動作

DNS インスペクションがイネーブルであるとき、DNS リライトは、任意のインターフェイスから送信された DNS メッセージの NAT を完全にサポートします。

内部のネットワーク上のクライアントが、外部インターフェイス上の DNS サーバから送信される内部アドレスの DNS 解決を要求した場合、DNS A レコードは正しく変換されます。DNS インスペクション エンジンがディセーブルである場合、A レコードは変換されません。

DNS インスペクションがイネーブルのままである間、**alias**、**static**、または **nat** コマンドを使用して DNS リライトを設定できます。

必要なコンフィギュレーションの詳細については、「DNS リライトの設定」（P.47-3）を参照してください。

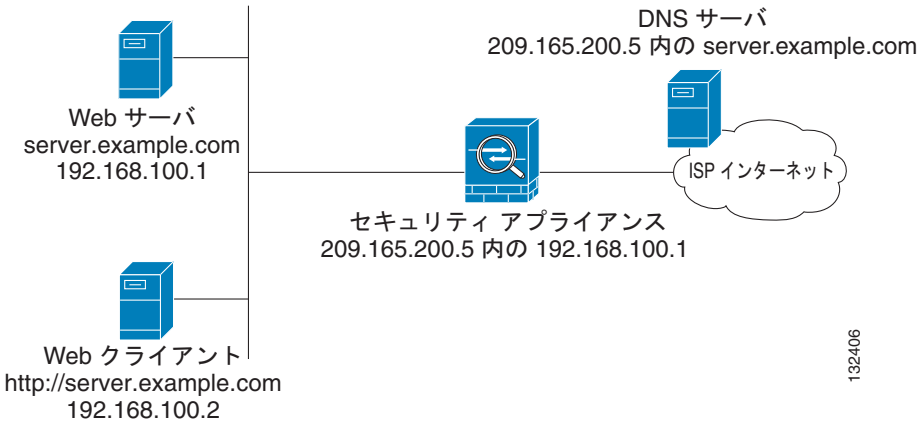
DNS リライトは次の 2 つの機能を実行します。

- DNS クライアントがプライベート インターフェイスにある場合、*DNS 応答*のパブリック アドレス（ルーティング可能なアドレスまたは「マッピング」アドレス）をプライベート アドレス（「実際の」アドレス）に変換します。
- DNS クライアントがパブリック インターフェイスにある場合、プライベート アドレスをパブリック アドレスに変換します。

図 47-1 では、DNS サーバは外部（ISP）ネットワークにあります。サーバの実際のアドレス（192.168.100.1）は、**static** コマンドで ISP が割り当てたアドレス（209.165.200.5）にマッピングされています。内部インターフェイスの Web クライアントが http://server.example.com という URL の Web サーバにアクセスしようとすると、Web クライアントが動作するホストが、Web サーバの IP アドレスの解決を求める DNS 要求を DNS サーバに送信します。ASA は、IP ヘッダーに含まれるルーティング不可の送信元アドレスを変換し、外部インターフェイスの ISP ネットワークに要求を転送します。DNS 応答が返されると、ASA はアドレス変換を宛先アドレスだけではなく、DNS 応答の A レコードに含まれる、埋め込まれた Web サーバの IP アドレスにも適用します。結果として、内部ネットワーク上の Web クライアントは、内部ネットワーク上の Web サーバとの接続に使用する正しいアドレスを取得します。

このような事例の設定手順については、「2 つの NAT ゾーンを持つ DNS リライトの設定」（P.47-4）を参照してください。

図 47-1　　DNS 応答に含まれるアドレスの変換（DNS リライト）



DNS リライトは、DNS 要求を作成するクライアントが DMZ ネットワークにあり、DNS サーバが内部インターフェイスにある場合にも機能します。この事例の詳細と設定手順については、「3 つの NAT ゾーンを持つ DNS リライトの概要」（P.47-5）を参照してください。

DNS リライトの設定

NAT コンフィギュレーションを使用して DNS リライトを設定します。

この項は、次の内容で構成されています。

2 つの NAT ゾーンを持つ DNS リライトの設定

図 47-1 の DNS リライトと類似の事例を実装するには、次の手順を実行します。

ステップ 1　**dns** オプションを使用して Web サーバのスタティック変換を作成します。第 34 章「ネットワーク オブジェクト NAT の設定」を参照してください。

ステップ 2　Web サーバが HTTP 要求を受信するポートへのトラフィックを許可するアクセス リストを作成します。

```
hostname(config)# access-list acl-name extended permit tcp any host mapped-address eq port
```

引数は次のとおりです。

acl-name：アクセス リストに付けた名前。

mapped-address：Web サーバの変換後の IP アドレス。

port：Web サーバが HTTP 要求を受信する TCP ポート。

ステップ 3　ステップ 2 で作成したアクセス リストをマッピング インターフェイスに適用します。これを行うには、**access-group** コマンドを次のように使用します。

```
hostname(config)# access-group acl-name in interface mapped_ifc
```

ステップ 4　DNS インスペクションがディセーブルの場合、または最大 DNS パケット長を変更する場合は、DNS インスペクションを設定します。デフォルトで、DNS アプリケーション インスペクションは、最大 DNS パケット長を 512 バイトとしてイネーブルになっています。設定手順については、「インスペクション制御を追加するための DNS インスペクション ポリシー マップの設定」（P.47-7）を参照してください。

ステップ 5　パブリック DNS サーバで、次のように Web サーバの A レコードを追加します。

```
domain-qualified-hostname.IN A mapped-address
```

domain-qualified-hostname は、server.example.com のようにドメイン サフィックスを付けたホスト名です。ホスト名の後ろのピリオドは重要です。*mapped-address* は、Web サーバの変換後の IP アドレスです。

次の例では、図 47-1 の事例の ASA を設定しています。DNS インスペクションはすでにイネーブルになっていることが前提です。

```
hostname(config)# object network obj-192.168.100.1-01
hostname(config-network-object)# host 192.168.100.1
hostname(config-network-object)# nat (inside,outside) static 209.165.200.225 dns
hostname(config)# access-list 101 permit tcp any host 209.165.200.225 eq www
hostname(config)# access-group 101 in interface outside
```

このコンフィギュレーションには、DNS サーバの次の A レコードが必要です。

```
server.example.com.IN A 209.165.200.225
```

3 つの NAT ゾーンを持つ DNS リライトの概要

図 47-2 では、DNS インスペクションによってどのようにして NAT が最小コンフィギュレーションの DNS サーバと透過的に連携動作するかを示す、より複雑な事例を示します。このような事例の設定手順については、「3 つの NAT ゾーンを持つ DNS リライトの設定」（P.47-6）を参照してください。

図 47-2　　3 つの NAT ゾーンを持つ DNS リライト

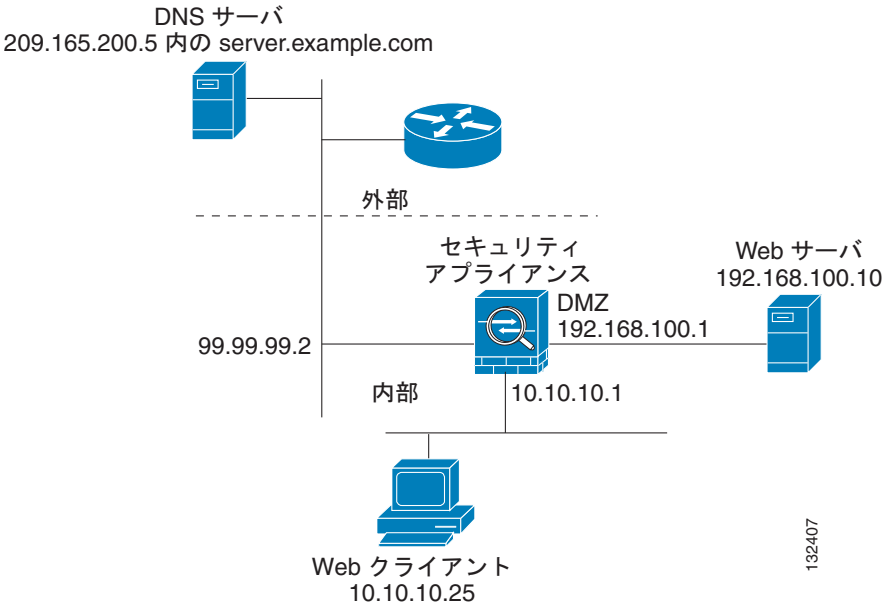


図 47-2 で、Web サーバ server.example.com の実際のアドレスは、ASA の DMZ インターフェイスの 192.168.100.10 です。IP アドレス 10.10.10.25 の Web クライアントが内部インターフェイスにあり、パブリック DNS サーバが外部インターフェイスにあります。サイト NAT ポリシーは次のとおりです。

- 外部 DNS サーバは server.example.com の信頼できるアドレス レコードを保持しています。
- 外部ネットワークのホストは、ドメイン名が server.example.com の Web サーバに、外部 DNS サーバまたは IP アドレス 209.165.200.5 を使用して接続できます。
- 内部ネットワークのクライアントは、ドメイン名が server.example.com の Web サーバに、外部 DNS サーバまたは IP アドレス 192.168.100.10 を使用してアクセスできます。

いずれかのインターフェイスのホストまたはクライアントは、DMZ Web サーバにアクセスするときに、パブリック DNS サーバに対して server.example.com の A レコードを問い合わせます。DNS サーバは、server.example.com がアドレス 209.165.200.5 にバインドされていることを示す A レコードを返します。

*外部*ネットワークの Web クライアントが http://server.example.com にアクセスを試みたときのイベント シーケンスは次のとおりです。

1. Web クライアントを実行しているホストが DNS サーバに、server.example.com の IP アドレスを求める要求を送信します。
2. DNS サーバが応答で IP アドレス 209.165.200.225 を示します。
3. Web クライアントが HTTP 要求を 209.165.200.225 に送信します。
4. 外部ホストからのパケットが ASA の外部インターフェイスに到達します。

5. スタティック ルールによってアドレス 209.165.200.225 が 192.168.100.10 に変換され、ASA がパケットを DMZ の Web サーバに誘導します。

*内部*ネットワークの Web クライアントが http://server.example.com にアクセスを試みたときのイベント シーケンスは次のとおりです。

1. Web クライアントを実行しているホストが DNS サーバに、server.example.com の IP アドレスを求める要求を送信します。
2. DNS サーバが応答で IP アドレス 209.165.200.225 を示します。
3. ASA が DNS 応答を受信し、その応答を DNS アプリケーション インスペクション エンジンに送信します。
4. DNS アプリケーション インスペクション エンジンは、次の処理を行います。
   a. 埋め込まれた A レコード アドレス「[outside]:209.165.200.5」の変換を元に戻す NAT ルールを検索します。この例では、次のスタティック コンフィギュレーションが検索されます。

```
object network obj-192.168.100.10-01
  host 192.168.100.10
  nat (dmz,outside) static 209.165.200.5 dns
```

   b. **dns** オプションが含まれているため、次のように A レコードをリライトするスタティック ルールを使用します。

```
[outside]:209.165.200.225 --> [dmz]:192.168.100.10
```

   （注）　**nat** コマンドに **dns** オプションが含まれていない場合、DNS リライトは実行されず、他のパケット処理が継続されます。

   c. 内部 Web クライアントと通信するときに、Web サーバ アドレス [dmz]:192.168.100.10 を変換する NAT が検索されます。

      適用可能な NAT ルールがない場合、アプリケーション インスペクションは終了します。

      NAT ルール（nat または static）が適用可能な場合は、**dns** オプションも指定されている必要があります。**dns** オプションが指定されていなかった場合、ステップ b の A レコード リライトは取り消され、他のパケット処理が継続されます。
5. ASA が DMZ インターフェイスの server.example.com に HTTP 要求を送信します。

3 つの NAT ゾーンを持つ DNS リライトの設定

図 47-2 の事例の NAT ポリシーをイネーブルにするには、次の手順を実行します。

ステップ 1　**dns** オプションを使用して、DMZ ネットワークの Web サーバのスタティック変換を作成します。第 34 章「ネットワーク オブジェクト NAT の設定」を参照してください。

ステップ 2　Web サーバが HTTP 要求を受信するポートへのトラフィックを許可するアクセス リストを作成します。

```
hostname(config)# access-list acl-name extended permit tcp any host mapped-address eq port
```

引数は次のとおりです。

acl-name：アクセス リストに付けた名前。

mapped-address：Web サーバの変換後の IP アドレス。

port：Web サーバが HTTP 要求を受信する TCP ポート。

ステップ 3　ステップ 2 で作成したアクセス リストを外部インターフェイスに適用します。これを行うには、**access-group** コマンドを次のように使用します。

```
hostname(config)# access-group acl-name in interface outside
```

ステップ 4　DNS インスペクションがディセーブルの場合、または最大 DNS パケット長を変更する場合は、DNS インスペクションを設定します。デフォルトで、DNS アプリケーション インスペクションは、最大 DNS パケット長を 512 バイトとしてイネーブルになっています。設定手順については、「インスペクション制御を追加するための DNS インスペクション ポリシー マップの設定」（P.47-7）を参照してください。

ステップ 5　パブリック DNS サーバで、次のように Web サーバの A レコードを追加します。

```
domain-qualified-hostname.IN A mapped-address
```

domain-qualified-hostname は、server.example.com のようにドメイン サフィックスを付けたホスト名です。ホスト名の後ろのピリオドは重要です。*mapped-address* は、Web サーバの変換後の IP アドレスです。

次の例では、図 47-2 の事例の ASA を設定しています。DNS インスペクションはすでにイネーブルになっていることが前提です。

```
hostname(config)# object network obj-192.168.100.10-01
hostname(config-network-object)# host 192.168.100.10
hostname(config-network-object)# nat (dmz,outside) static 209.165.200.225 dns
hostname(config)# access-list 101 permit tcp any host 209.165.200.225 eq www
hostname(config)# access-group 101 in interface outside
```

このコンフィギュレーションには、DNS サーバの次の A レコードが必要です。

```
server.example.com.IN A 209.165.200.225
```

インスペクション制御を追加するための DNS インスペクション ポリシー マップの設定

DNS アプリケーション インスペクションは、DNS スプーフィングとキャッシュ ポイズニングを防ぐための DNS メッセージ制御をサポートしています。ユーザが設定可能なルールによって、DNS ヘッダー、ドメイン名、リソース レコードのタイプ、およびクラスに基づいてフィルタリングを行うことができます。たとえば、ゾーン転送をこの機能のあるサーバ間だけに制限できます。

公開サーバが特定の内部ゾーンだけをサポートしている場合に、DNS ヘッダーにある Recursion Desired フラグと Recursion Available フラグをマスクして、サーバを攻撃から守ることができます。また、DNS のランダム化をイネーブルにすると、ランダム化をサポートしていないサーバや強度の低い疑似乱数ジェネレータを使用するサーバのスプーフィングやキャッシュ ポイズニングを回避できます。照会できるドメイン名を制限することにより、公開サーバの保護がさらに確実になります。

不一致の DNS 応答数が過度に増えた場合（キャッシュ ポイズニング攻撃を示している可能性がある）、DNS 不一致のアラートを設定して通知することができます。さらに、すべての DNS メッセージにトランザクション署名（TSIG）を付けるようにチェックする設定も行うことができます。

メッセージがパラメータに違反したときのアクションを指定するには、DNS インスペクション ポリシー マップを作成します。作成したインスペクション ポリシー マップは、DNS インスペクションをイネーブルにすると適用できます。

DNS インスペクション ポリシー マップを作成するには、次の手順を実行します。

ステップ 1　（任意）「正規表現の作成」（P.18-15）に従って、1 つ以上の正規表現をトラフィック照合コマンドに追加して使用できるようにします。ステップ 3 に記載されている **match** コマンドで照合できるテキストのタイプを参照してください。

ステップ 2　（任意）「正規表現クラス マップの作成」（P.18-17）に従って、1 つ以上の正規表現のクラス マップを作成して正規表現をグループ化します。

ステップ 3　（任意）次の手順に従って、DNS インスペクションのクラス マップを作成します。

クラス マップは複数のトラフィックとの照合をグループ化します。クラス マップと一致するには、トラフィックは、すべての **match** コマンドと一致する必要があります。または、**match** コマンドを直接ポリシー マップに指定できます。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 マップでトラフィックとの照合を直接定義することの違いは、クラス マップでは複雑な照合基準を作成でき、クラス マップを再利用できるということです。

クラス マップと照合しないトラフィックを指定するには、**match not** コマンドを使用します。たとえば、**match not** コマンドで文字列「example.com」を指定すると、「example.com」が含まれるすべてのトラフィックはクラス マップと照合されません。

このクラス マップで特定するトラフィックに対して、ドロップ、接続のドロップ、リセット、マスク、レート制限の設定、接続のロギングなどのアクションをインスペクション ポリシー マップに指定できます。

match コマンドごとに異なるアクションを実行する場合、ポリシー マップに直接トラフィックを特定する必要があります。

a. 次のコマンドを入力して、クラス マップを作成します。

```
hostname(config)# class-map type inspect dns [match-all | match-any] class_map_name
hostname(config-cmap)#
```

class_map_name には、クラス マップの名前を指定します。**match-all** キーワードはデフォルトです。トラフィックがクラス マップと一致するには、すべての基準と一致する必要があることを指定します。**match-any** キーワードは、トラフィックが少なくとも基準の 1 つに一致したらクラス マップと一致することを指定します。CLI がクラスマップ コンフィギュレーション モードに入り、1 つ以上の **match** コマンドを入力できます。

b. （任意）クラス マップに説明を追加するには、次のコマンドを使用します。

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

c. （任意）DNS ヘッダーに設定されている特定のフラグを照合するには、次のコマンドを入力します。

```
hostname(config-cmap)# match [not] header-flag [eq] {f_well_known | f_value}
```

f_well_known 引数には、DNS のフラグ ビットを指定します。*f_value* 引数には、16 ビットの値を 16 進数で指定します。**eq** キーワードは完全一致を指定します。

d. （任意）DNS タイプ（クエリー タイプや RR タイプなど）を照合するには、次のコマンドを入力します。

```
hostname(config-cmap)# match [not] dns-type {eq t_well_known | t_val} {range t_val1
t_val2}
```

t_well_known 引数には、DNS のフラグ ビットを指定します。*t_val* 引数には、DNS タイプ フィールドの任意の値（0 ～ 65535）を指定します。**range** キーワードは範囲を指定し、**eq** キーワードは完全一致を指定します。

e. （任意）DNS クラスを照合するには、次のコマンドを入力します。

```
hostname(config-cmap)# match [not] dns-class {eq c_well_known | c_val} {range c_val1
c_val2}
```

c_well_known 引数には、DNS クラスを指定します。*c_val* 引数には、DNS クラス フィールドの任意の値を指定します。**range** キーワードは範囲を指定し、**eq** キーワードは完全一致を指定します。

f. （任意）DNS の問い合わせまたはリソース レコードを照合するには、次のコマンドを入力します。

```
hostname(config-cmap)# match {question | {resource-record answer | authority | any}}
```

question キーワードを指定すると、DNS メッセージの問い合わせ部分を照合します。**resource-record** キーワードを指定すると、DNS メッセージのリソース レコード部分を照合します。***answer*** キーワードを指定すると、Answer RR セクションを照合します。**authority** キーワードを指定すると、Authority RR セクションを照合します。***additional*** キーワードを指定すると、Additional RR セクションを照合します。

g. （任意）DNS メッセージのドメイン名リストを照合するには、次のコマンドを入力します。

```
hostname(config-cmap)# match [not] domain-name {regex regex_id | regex class class_id]
```

regex *regex_name* 引数には、ステップ 1 で作成した正規表現を指定します。**class** *regex_class_name* には、ステップ 2 で作成した正規表現のクラス マップを指定します。

ステップ 4　DNS インスペクション ポリシー マップを作成するには、次のコマンドを入力します。

```
hostname(config)# policy-map type inspect dns policy_map_name
hostname(config-pmap)#
```

policy_map_name には、ポリシー マップの名前を指定します。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。

ステップ 5　（任意）このポリシー マップに説明を追加するには、次のコマンドを使用します。

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

ステップ 6　一致したトラフィックにアクションを適用するには、次の手順を実行します。

a. 次のいずれかの方法を使用して、アクションを実行するトラフィックを指定します。

- 次のコマンドを入力して、ステップ 3 で作成した DNS クラス マップを指定します。

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

- ステップ 3 で説明されている **match** コマンドのいずれかを使用して、ポリシー マップに直接トラフィックを指定します。**match not** コマンドを使用すると、**match not** コマンドの基準に一致しないすべてのトラフィックにアクションが適用されます。

b. 次のコマンドを入力して、一致したトラフィックに対して実行するアクションを指定します。

```
hostname(config-pmap-c)# {[drop [send-protocol-error] |
drop-connection [send-protocol-error]| mask | reset] [log] | rate-limit message_rate}
```

各 **match** コマンドまたは **class** コマンドですべてのオプションを使用できるわけではありません。使用できる正確なオプションについては、CLI ヘルプまたはコマンド リファレンスを参照してください。

drop キーワードを指定すると、一致するすべてのパケットをドロップします。

send-protocol-error キーワードを指定すると、プロトコル エラー メッセージを送信します。

drop-connection キーワードを指定すると、パケットをドロップし、接続を閉じます。

mask キーワードを指定すると、パケットの一致部分をマスクします。

reset キーワードを指定すると、パケットをドロップして接続を閉じ、サーバとクライアントの両方またはいずれかに TCP リセットを送信します。

log キーワードを指定すると、システム ログ メッセージを送信します。このキーワードは単独で、または他のキーワードのいずれかと一緒に使用できます。

rate-limit *message_rate* 引数では、メッセージのレートを制限します。

ポリシー マップには、複数の **class** コマンドまたは **match** コマンドを指定できます。**class** コマンドと **match** コマンドの順序については、「インスペクション ポリシー マップのアクションの定義」（P.37-3）を参照してください。

ステップ 7　インスペクション エンジンに影響のあるパラメータを設定するには、次の手順を実行します。

a. パラメータ コンフィギュレーション モードに入るには、次のコマンドを入力します。

```
hostname(config-pmap)# parameters
hostname(config-pmap-p)#
```

b. DNS クエリーの DNS 識別子をランダム化するには、次のコマンドを入力します。

```
hostname(config-pmap-p)# id-randomization
```

c. 過度な DNS ID の不一致をロギングするには、次のコマンドを入力します。

```
hostname(config-pmap-p)# id-mismatch [count number duration seconds] action log
```

count *string* 引数には、不一致インスタンスの最大数を指定します。一致しないインスタンスがこの数を超えたら、システム ログ メッセージを送信します。**duration** ***seconds*** には、モニタする期間を秒単位で指定します。

d. TSIG リソース レコードが存在することを必須とするには、次のコマンドを入力します。

```
hostname(config-pmap-p)# tsig enforced action {drop [log] | [log}
```

count *string* 引数には、不一致インスタンスの最大数を指定します。一致しないインスタンスがこの数を超えたら、システム ログ メッセージを送信します。**duration** ***seconds*** には、モニタする期間を秒単位で指定します。

次の例は、DNS インスペクション ポリシー マップを定義する方法を示しています。

```
hostname(config)# regex domain_example "example\.com"
hostname(config)# regex domain_foo "foo\.com"

hostname(config)# !define the domain names that the server serves
hostname(config)# class-map type inspect regex match-any my_domains
hostname(config-cmap)# match regex domain_example
hostname(config-cmap)# match regex domain_foo

hostname(config)# !Define a DNS map for query only
hostname(config)# class-map type inspect dns match-all pub_server_map
hostname(config-cmap)# match not header-flag QR
hostname(config-cmap)# match question
hostname(config-cmap)# match not domain-name regex class my_domains

hostname(config)# policy-map type inspect dns serv_prot
hostname(config-pmap)# class pub_server_map
hostname(config-pmap-c)# drop log
hostname(config-pmap-c)# match header-flag RD
hostname(config-pmap-c)# mask log

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

```
hostname(config)# policy-map pub_policy
hostname(config-pmap)# class dns_serv_map
hostname(config-pmap-c)# inspect dns serv_prot

hostname(config)# service-policy pub_policy interface dmz
```

DNS インスペクションの確認とモニタリング

現在の DNS 接続に関する情報を表示するには、次のコマンドを入力します。

```
hostname# show conn
```

DNS サーバを使用する接続の場合、show conn コマンドの出力で、接続の送信元ポートが DNS サーバの IP アドレスに置き換えられることがあります。

複数の DNS セッションが同じ 2 つのホスト間で発生し、それらのセッションの 5 つのタプル（送信元/宛先 IP アドレス、送信元/宛先ポート、およびプロトコル）が同じものである場合、それらのセッションに対しては接続が 1 つだけ作成されます。DNS ID は app_id で追跡され、各 app_id のアイドル タイマーは独立して実行されます。

app_id の有効期限はそれぞれ独立して満了するため、正当な DNS 応答がセキュリティ アプライアンスを通過できるのは、限られた期間内だけであり、リソースの継続使用はできません。ただし、**show conn** コマンドを入力した場合、新しい DNS セッションによってリセットされている DNS 接続のアイドル タイマーが表示されます。これは共有 DNS 接続の性質によるものであり、仕様です。

DNS アプリケーション インスペクションの統計情報を表示するには、**show service-policy** コマンドを入力します。次に、**show service-policy** コマンドの出力例を示します。

```
hostname# show service-policy
Interface outside:
Service-policy: sample_policy
    Class-map: dns_port
      Inspect: dns maximum-length 1500, packet 0, drop 0, reset-drop 0
```

FTP インスペクション

この項では、FTP インスペクション エンジンについて説明します。この項は、次の内容で構成されています。

- 「FTP インスペクションの概要」（P.47-11）
- 「strict オプションの使用」（P.47-12）
- 「インスペクション制御を追加するための FTP インスペクション ポリシー マップの設定」（P.47-13）
- 「FTP 検査の確認とモニタリング」（P.47-16）

FTP インスペクションの概要

FTP アプリケーション インスペクションは、FTP セッションを検査し、次の 4 つのタスクを実行します。

- ダイナミックな二次的データ接続の準備

- FTP コマンド応答シーケンスの追跡
- 監査証跡の生成
- 埋め込み IP アドレスの変換

FTP アプリケーション インスペクションによって、FTP データ転送用にセカンダリ チャネルが用意されます。これらのチャネルのポートは、PORT コマンドまたは PASV コマンドを使用してネゴシエートされます。セカンダリ チャネルは、ファイル アップロード、ファイル ダウンロード、またはディレクトリ リスト イベントへの応答で割り当てられます。

（注）　**no inspect ftp** コマンドを使用して、FTP インスペクション エンジンをディセーブルにすると、発信ユーザはパッシブ モードだけで接続を開始でき、着信 FTP はすべてディセーブルになります。

strict オプションの使用

inspect ftp コマンドに **strict** オプションを使用すると、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が向上します。

（注）　ASA の通過を禁止する FTP コマンドを指定するには、「インスペクション制御を追加するための FTP インスペクション ポリシー マップの設定」（P.47-13）に従って FTP マップを作成します。

インターフェイスに対して **strict** オプションをオンにすると、FTP インスペクションによって次の動作が適用されます。

- FTP コマンドが確認応答されてからでないと、ASA は新しいコマンドを許可しません。
- ASA は、埋め込みコマンドを送信する接続をドロップします。
- 227 コマンドと PORT コマンドが、エラー文字列に表示されないように確認されます。

注意　**strict** オプションを使用すると、FTP RFC に厳密に準拠していない FTP クライアントは失敗することがあります。

strict オプションがイネーブルの場合、各 FTP コマンドと応答シーケンスが追跡され、次の異常なアクティビティがないか確認されます。

- 切り捨てされたコマンド：PORT コマンドおよび PASV 応答コマンドのカンマの数が 5 であるかどうかが確認されます。カンマの数が 5 でない場合は、PORT コマンドが切り捨てられていると見なされ、TCP 接続は閉じられます。
- 不正なコマンド：FTP コマンドが、RFC の要求どおりに <CR><LF> 文字で終了しているかどうか確認されます。終了していない場合は、接続が閉じられます。
- RETR コマンドと STOR コマンドのサイズ：これらが、固定の定数と比較チェックされます。サイズが定数より大きい場合は、エラー メッセージがロギングされ、接続が閉じられます。
- コマンド スプーフィング：PORT コマンドは、常にクライアントから送信されます。PORT コマンドがサーバから送信される場合、TCP 接続は拒否されます。
- 応答スプーフィング：PASV 応答コマンド（227）は、常にサーバから送信されます。PASV 応答コマンドがクライアントから送信される場合、TCP 接続は拒否されます。これにより、ユーザが「227 xxxxx a1, a2, a3, a4, p1, p2.」を実行する場合のセキュリティ ホールが予防できます。
- TCP ストリーム編集：ASA は、TCP ストリーム編集を検出した場合に接続が閉じられます。

- 無効ポート ネゴシエーション：ネゴシエートされたダイナミック ポート値が、1024 未満であるかどうかが調べられます。1 ～ 1024 の範囲のポート番号は、予約済み接続用に指定されているため、ネゴシエートされたポートがこの範囲内であった場合、TCP 接続は解放されます。
- コマンド パイプライン：PORT コマンドと PASV 応答コマンド内のポート番号の後に続く文字数が、定数の 8 と比べられます。8 より大きい場合は、TCP 接続が閉じられます。
- ASA は、SYST コマンドに対する FTP サーバ応答を X の連続に置き換えることで、FTP クライアントがサーバのシステム タイプを取得できないようにします。このデフォルトの動作を無効にするには、FTP マップで、**no mask-syst-reply** コマンドを使用します。

インスペクション制御を追加するための FTP インスペクション ポリシー マップの設定

厳密な FTP インスペクションには、セキュリティと制御を向上させるためのコマンド フィルタリングとセキュリティ チェック機能が用意されています。プロトコルとの適合性のインスペクションには、パケットの長さのチェック、デリミタとパケットの形式のチェック、コマンドのターミネータのチェック、およびコマンドの検証が含まれます。

また、ユーザの値に基づいて FTP 接続をブロックできるので、FTP サイトにダウンロード用のファイルを置き、アクセスを特定のユーザだけに制限できます。ファイルのタイプ、サーバ名、および他の属性に基づいて、FTP 接続をブロックできます。インスペクション時に FTP 接続が拒否されると、システム メッセージのログが作成されます。

FTP インスペクションで FTP サーバがそのシステム タイプを FTP クライアントに公開することを許可し、許可する FTP コマンドを制限する場合、FTP マップを作成および設定します。作成した FTP マップは、FTP インスペクションをイネーブルにすると適用できます。

FTP マップを作成するには、次の手順を実行します。

ステップ 1　（任意）「正規表現の作成」（P.18-15）に従って、1 つ以上の正規表現をトラフィック照合コマンドに追加して使用できるようにします。ステップ 3 に記載されている **match** コマンドで照合できるテキストのタイプを参照してください。

ステップ 2　（任意）「正規表現クラス マップの作成」（P.18-17）に従って、1 つ以上の正規表現のクラス マップを作成して正規表現をグループ化します。

ステップ 3　（任意）次の手順に従って、FTP インスペクションのクラス マップを作成します。

クラス マップは複数のトラフィックとの照合をグループ化します。クラス マップと一致するには、トラフィックは、すべての **match** コマンドと一致する必要があります。または、**match** コマンドを直接ポリシー マップに指定できます。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 マップでトラフィックとの照合を直接定義することの違いは、クラス マップでは複雑な照合基準を作成でき、クラス マップを再利用できるということです。

クラス マップと照合しないトラフィックを指定するには、**match not** コマンドを使用します。たとえば、**match not** コマンドで文字列「example.com」を指定すると、「example.com」が含まれるすべてのトラフィックはクラス マップと照合されません。

このクラス マップで特定するトラフィックに対して、ドロップ、接続のドロップ、リセット、マスク、レート制限の設定、接続のロギングなどのアクションをインスペクション ポリシー マップに指定できます。

match コマンドごとに異なるアクションを実行する場合、ポリシー マップに直接トラフィックを特定する必要があります。

a.　次のコマンドを入力して、クラス マップを作成します。

```
hostname(config)# class-map type inspect ftp [match-all | match-any] class_map_name
hostname(config-cmap)#
```

class_map_name には、クラス マップの名前を指定します。**match-all** キーワードはデフォルトです。トラフィックがクラス マップと一致するには、すべての基準と一致する必要があることを指定します。**match-any** キーワードは、トラフィックが少なくとも基準の 1 つに一致したらクラス マップと一致することを指定します。CLI がクラスマップ コンフィギュレーション モードに入り、1 つ以上の **match** コマンドを入力できます。

b. （任意）クラス マップに説明を追加するには、次のコマンドを使用します。

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

c. （任意）FTP 転送のファイル名を照合するには、次のコマンドを入力します。

```
hostname(config-cmap)# match [not] filename regex [regex_name |
class regex_class_name]
```

regex_name には、ステップ 1 で作成した正規表現を指定します。**class** *regex_class_name* には、ステップ 2 で作成した正規表現のクラス マップを指定します。

d. （任意）FTP 転送のファイル タイプを照合するには、次のコマンドを入力します。

```
hostname(config-cmap)# match [not] filetype regex [regex_name |
class regex_class_name]
```

regex_name には、ステップ 1 で作成した正規表現を指定します。**class** *regex_class_name* には、ステップ 2 で作成した正規表現のクラス マップを指定します。

e. （任意）特定の FTP コマンドを禁止するには、次のコマンドを入力します。

```
hostname(config-cmap)# match [not] request-command ftp_command [ftp_command...]
```

ftp_command には、制限する 1 つ以上の FTP コマンドを指定します。制限できる FTP コマンドのリストについては、表 47-1 を参照してください。

表 47-1　FTP マップの request-command deny オプション

request-command deny オプション	目的
appe	ファイルへの追加を行うコマンドを拒否します。
cdup	現在の作業ディレクトリの親ディレクトリに移動するコマンドを拒否します。
dele	サーバのファイルを削除するコマンドを拒否します。
get	サーバからファイルを取得するクライアント コマンドを拒否します。
help	ヘルプ情報を提供するコマンドを拒否します。
mkd	サーバ上にディレクトリを作成するコマンドを拒否します。
put	サーバにファイルを送信するクライアント コマンドを拒否します。
rmd	サーバ上のディレクトリを削除するコマンドを拒否します。
rnfr	変更元ファイル名を指定するコマンドを拒否します。
rnto	変更先ファイル名を指定するコマンドを拒否します。

表 47-1　FTP マップの request-command deny オプション （続き）

request-command deny オプション	目的
site	サーバ システム固有のコマンドを拒否します。通常、リモート管理に使用します。
stou	固有のファイル名を使用してファイルを保存するコマンドを拒否します。

f. （任意）FTP サーバを照合するには、次のコマンドを入力します。

```
hostname(config-cmap)# match [not] server regex [regex_name | class regex_class_name]
```

regex_name には、ステップ 1 で作成した正規表現を指定します。**class** *regex_class_name* には、ステップ 2 で作成した正規表現のクラス マップを指定します。

g. （任意）FTP ユーザ名を照合するには、次のコマンドを入力します。

```
hostname(config-cmap)# match [not] username regex [regex_name |
class regex_class_name]
```

regex_name には、ステップ 1 で作成した正規表現を指定します。**class** *regex_class_name* には、ステップ 2 で作成した正規表現のクラス マップを指定します。

h. （任意）アクティブな FTP トラフィック コマンド PORT および EPRT を照合するには、次のコマンドを入力します。

```
hostname(config-cmap)# match [not] active-ftp
```

i. （任意）パッシブな FTP トラフィック コマンド PASV および EPSV を照合するには、次のコマンドを入力します。

```
hostname(config-cmap)# match [not] passive-ftp
```

ステップ 4　FTP インスペクション ポリシー マップを作成するには、次のコマンドを入力します。

```
hostname(config)# policy-map type inspect ftp policy_map_name
hostname(config-pmap)#
```

policy_map_name には、ポリシー マップの名前を指定します。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。

ステップ 5　（任意）このポリシー マップに説明を追加するには、次のコマンドを使用します。

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

ステップ 6　一致したトラフィックにアクションを適用するには、次の手順を実行します。

a. 次のいずれかの方法を使用して、アクションを実行するトラフィックを指定します。

- 次のコマンドを入力して、ステップ 3 で作成した FTP クラス マップを指定します。

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

- ステップ 3 で説明されている **match** コマンドのいずれかを使用して、ポリシー マップに直接トラフィックを指定します。**match not** コマンドを使用すると、**match not** コマンドの基準に一致しないすべてのトラフィックにアクションが適用されます。

b. 次のコマンドを入力して、一致したトラフィックに対して実行するアクションを指定します。

```
hostname(config-pmap-c)# {[drop [send-protocol-error] |
drop-connection [send-protocol-error]| mask | reset] [log] | rate-limit message_rate}
```

各 **match** コマンドまたは **class** コマンドですべてのオプションを使用できるわけではありません。使用できる正確なオプションについては、CLI ヘルプまたはコマンド リファレンスを参照してください。

drop キーワードを指定すると、一致するすべてのパケットをドロップします。

send-protocol-error キーワードを指定すると、プロトコル エラー メッセージを送信します。

drop-connection キーワードを指定すると、パケットをドロップし、接続を閉じます。

mask キーワードを指定すると、パケットの一致部分をマスクします。

reset キーワードを指定すると、パケットをドロップして接続を閉じ、サーバとクライアントの両方またはいずれかに TCP リセットを送信します。

log キーワードを指定すると、システム ログ メッセージを送信します。このキーワードは単独で、または他のキーワードのいずれかと一緒に使用できます。

rate-limit *message_rate* 引数では、メッセージのレートを制限します。

ポリシー マップには、複数の **class** コマンドまたは **match** コマンドを指定できます。**class** コマンドと **match** コマンドの順序については、「インスペクション ポリシー マップのアクションの定義」（P.37-3）を参照してください。

ステップ 7　インスペクション エンジンに影響のあるパラメータを設定するには、次の手順を実行します。

a. パラメータ コンフィギュレーション モードに入るには、次のコマンドを入力します。

```
hostname(config-pmap)# parameters
hostname(config-pmap-p)#
```

b. FTP サーバとの接続時に表示されるバナーをマスクするには、次のコマンドを入力します。

```
hostname(config-pmap-p)# mask-banner
```

c. **syst** コマンドへの応答をマスクするには、次のコマンドを入力します。

```
hostname(config-pmap-p)# mask-syst-reply
```

ユーザ名とパスワードを送信する前に、すべての FTP ユーザに接続時バナーが表示されます。デフォルトでは、このバナーには、ハッカーがシステムの弱点を特定するのに役立つバージョン情報が含まれます。このバナーをマスクする方法を次に示します。

```
hostname(config)# policy-map type inspect ftp mymap
hostname(config-pmap)# parameters
hostname(config-pmap-p)# mask-banner

hostname(config)# class-map match-all ftp-traffic
hostname(config-cmap)# match port tcp eq ftp

hostname(config)# policy-map ftp-policy
hostname(config-pmap)# class ftp-traffic
hostname(config-pmap-c)# inspect ftp strict mymap

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

FTP 検査の確認とモニタリング

FTP アプリケーション インスペクションでは、次のログ メッセージが生成されます。

- An Audit record 303002 is generated for each file that is retrieved or uploaded.

- The FTP command is checked to see if it is RETR or STOR and the retrieve and store commands are logged.
- The username is obtained by looking up a table providing the IP address.
- The username, source IP address, destination IP address, NAT address, and the file operation are logged.
- Audit record 201005 is generated if the secondary dynamic channel preparation failed due to memory shortage.

NAT と連携することにより、FTP アプリケーション インスペクションでは、アプリケーション ペイロード内の IP アドレスが変換されます。これは、RFC 959 に詳細に記述されています。

HTTP インスペクション

この項では、HTTP インスペクション エンジンについて説明します。この項は、次の内容で構成されています。

- 「HTTP インスペクションの概要」（P.47-17）
- 「インスペクション制御を追加するための HTTP インスペクション ポリシー マップの設定」（P.47-18）

HTTP インスペクションの概要

HTTP インスペクション エンジンを使用して、特定の攻撃、および HTTP トラフィックに関係するその他の脅威から保護します。HTTP インスペクションは、次のようないくつかの機能を実行します。

- 拡張 HTTP インスペクション
- N2H2 または Websense を使用する URL のスクリーニング

  詳細については、「URL フィルタリングに関する情報」（P.64-7）を参照してください。
- Java と ActiveX のフィルタリング

後の 2 つの機能は、**filter** コマンドとともに設定します。フィルタリングの詳細については、第 64 章「フィルタリング サービスの設定」を参照してください。

拡張 HTTP インスペクション機能はアプリケーション ファイアウォールとも呼ばれ、HTTP マップを設定するときに使用できます（「インスペクション制御を追加するための HTTP インスペクション ポリシー マップの設定」を参照）。これによって、攻撃者がネットワーク セキュリティ ポリシーに従わない HTTP メッセージを使用できないようにします。この機能は、すべての HTTP メッセージについて次のことを確認します。

- RFC 2616 への準拠
- RFC で定義された方式だけを使用していること
- 追加の基準への準拠

インスペクション制御を追加するための HTTP インスペクション ポリシー マップの設定

メッセージがパラメータに違反したときのアクションを指定するには、HTTP インスペクション ポリシー マップを作成します。作成したインスペクション ポリシー マップは、HTTP インスペクションをイネーブルにすると適用できます。

（注）　HTTP インスペクション ポリシー マップを使用して HTTP インスペクションをイネーブルにすると、デフォルトでは、アクション reset および log を使用した厳密な HTTP インスペクションがイネーブルになります。インスペクションに合格しない場合に実行されるアクションは変更できますが、インスペクション ポリシー マップがイネーブルのままである限り、厳密なインスペクションをディセーブルにすることはできません。

HTTP インスペクション ポリシー マップを作成するには、次の手順を実行します。

ステップ 1　（任意）「正規表現の作成」（P.18-15）に従って、1 つ以上の正規表現をトラフィック照合コマンドに追加して使用できるようにします。ステップ 3 に記載されている **match** コマンドで照合できるテキストのタイプを参照してください。

ステップ 2　（任意）「正規表現クラス マップの作成」（P.18-17）に従って、1 つ以上の正規表現のクラス マップを作成して正規表現をグループ化します。

ステップ 3　（任意）次の手順に従って、HTTP インスペクションのクラス マップを作成します。

クラス マップは複数のトラフィックとの照合をグループ化します。クラス マップと一致するには、トラフィックは、すべての **match** コマンドと一致する必要があります。または、**match** コマンドを直接ポリシー マップに指定できます。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 マップでトラフィックとの照合を直接定義することの違いは、クラス マップでは複雑な照合基準を作成でき、クラス マップを再利用できるということです。

クラス マップと照合しないトラフィックを指定するには、**match not** コマンドを使用します。たとえば、**match not** コマンドで文字列「example.com」を指定すると、「example.com」が含まれるすべてのトラフィックはクラス マップと照合されません。

このクラス マップで特定するトラフィックに対して、ドロップ、接続のドロップ、リセット、マスク、レート制限の設定、接続のロギングなどのアクションをインスペクション ポリシー マップに指定できます。

match コマンドごとに異なるアクションを実行する場合、ポリシー マップに直接トラフィックを特定する必要があります。

a.　次のコマンドを入力して、クラス マップを作成します。

```
hostname(config)# class-map type inspect http [match-all | match-any] class_map_name
hostname(config-cmap)#
```

class_map_name には、クラス マップの名前を指定します。**match-all** キーワードはデフォルトです。トラフィックがクラス マップと一致するには、すべての基準と一致する必要があることを指定します。**match-any** キーワードは、トラフィックが少なくとも基準の 1 つに一致したらクラス マップと一致することを指定します。CLI がクラスマップ コンフィギュレーション モードに入り、1 つ以上の **match** コマンドを入力できます。

b.　（任意）クラス マップに説明を追加するには、次のコマンドを使用します。

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

c. （任意）HTTP 応答の content-type フィールドが、対応する HTTP 要求メッセージの accept フィールドと一致しないトラフィックを照合するには、次のコマンドを入力します。

```
hostname(config-cmap)# match [not] req-resp content-type mismatch
```

d. （任意）HTTP 要求メッセージの引数に含まれるテキストを照合するには、次のコマンドを入力します。

```
hostname(config-cmap)# match [not] request args regex [regex_name | class
regex_class_name]
```

regex_name には、ステップ 1 で作成した正規表現を指定します。**class** *regex_class_name* には、ステップ 2 で作成した正規表現のクラス マップを指定します。

e. （任意）HTTP 要求メッセージ本文に含まれるテキストを照合するか、または HTTP 要求メッセージ本文の最大長を超えるトラフィックを照合するには、次のコマンドを入力します。

```
hostname(config-cmap)# match [not] request body {regex [regex_name | class
regex_class_name] | length gt max_bytes}
```

regex *regex_name* 引数には、ステップ 1 で作成した正規表現を指定します。**class** *regex_class_name* には、ステップ 2 で作成した正規表現のクラス マップを指定します。**length gt** *max_bytes* には、メッセージ本文の最大長をバイト単位で指定します。

f. （任意）HTTP 要求メッセージ ヘッダーに含まれるテキストを照合するか、ヘッダーのカウントまたは長さを制限するには、次のコマンドを入力します。

```
hostname(config-cmap)# match [not] request header {[field]
[regex [regex_name | class regex_class_name]] |
[length gt max_length_bytes | count gt max_count_bytes]}
```

field には、事前定義済みのメッセージ ヘッダーのキーワードを指定します。**regex** *regex_name* 引数には、ステップ 1 で作成した正規表現を指定します。**class** *regex_class_name* には、ステップ 2 で作成した正規表現のクラス マップを指定します。**length gt** *max_bytes* には、メッセージ本文の最大長をバイト単位で指定します。**count gt** *max_count* には、ヘッダー フィールドの最大数を指定します。

g. （任意）HTTP 要求メッセージ方式に含まれるテキストを照合するには、次のコマンドを入力します。

```
hostname(config-cmap)# match [not] request method {[method] |
[regex [regex_name | class regex_class_name]]
```

method には、事前定義済みのメッセージ方式のキーワードを指定します。**regex** *regex_name* 引数には、ステップ 1 で作成した正規表現を指定します。**class** *regex_class_name* には、ステップ 2 で作成した正規表現のクラス マップを指定します。

h. （任意）HTTP 要求メッセージの URI に含まれるテキストを照合するには、次のコマンドを入力します。

```
hostname(config-cmap)# match [not] request uri {regex [regex_name | class
regex_class_name] | length gt max_bytes}
```

regex *regex_name* 引数には、ステップ 1 で作成した正規表現を指定します。**class** *regex_class_name* には、ステップ 2 で作成した正規表現のクラス マップを指定します。**length gt** *max_bytes* には、メッセージ本文の最大長をバイト単位で指定します。

i. （任意）HTTP 応答メッセージ本文に含まれるテキストを照合するか、Java アプレットと Active X オブジェクトのタグをコメントアウトしてフィルタリングするには、次のコマンドを入力します。

```
hostname(config-cmap)# match [not] response body {[active-x] | [java-applet] |
[regex [regex_name | class regex_class_name]] | length gt max_bytes}
```

regex *regex_name* 引数には、ステップ 1 で作成した正規表現を指定します。**class** *regex_class_name* には、ステップ 2 で作成した正規表現のクラス マップを指定します。**length gt** *max_bytes* には、メッセージ本文の最大長をバイト単位で指定します。

j. （任意）HTTP 応答メッセージ ヘッダーに含まれるテキストを照合するか、ヘッダーのカウントまたは長さを制限するには、次のコマンドを入力します。

```
hostname(config-cmap)# match [not] response header {[field]
[regex [regex_name | class regex_class_name]] |
[length gt max_length_bytes | count gt max_count]}
```

field には、事前定義済みのメッセージ ヘッダーのキーワードを指定します。**regex** *regex_name* 引数には、ステップ 1 で作成した正規表現を指定します。**class** *regex_class_name* には、ステップ 2 で作成した正規表現のクラス マップを指定します。**length gt** *max_bytes* には、メッセージ本文の最大長をバイト単位で指定します。**count gt** *max_count* には、ヘッダー フィールドの最大数を指定します。

k. （任意）HTTP 応答メッセージのステータス行に含まれるテキストを照合するには、次のコマンドを入力します。

```
hostname(config-cmap)# match [not] response status-line {regex [regex_name | class
regex_class_name]}
```

regex *regex_name* 引数には、ステップ 1 で作成した正規表現を指定します。**class** *regex_class_name* には、ステップ 2 で作成した正規表現のクラス マップを指定します。

ステップ 4　HTTP インスペクション ポリシー マップを作成するには、次のコマンドを入力します。

```
hostname(config)# policy-map type inspect http policy_map_name
hostname(config-pmap)#
```

policy_map_name には、ポリシー マップの名前を指定します。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。

ステップ 5　（任意）このポリシー マップに説明を追加するには、次のコマンドを使用します。

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

ステップ 6　一致したトラフィックにアクションを適用するには、次の手順を実行します。

a. 次のいずれかの方法を使用して、アクションを実行するトラフィックを指定します。

- 次のコマンドを入力して、ステップ 3 で作成した HTTP クラス マップを指定します。

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

- ステップ 3 で説明されている **match** コマンドのいずれかを使用して、ポリシー マップに直接トラフィックを指定します。**match not** コマンドを使用すると、**match not** コマンドの基準に一致しないすべてのトラフィックにアクションが適用されます。

b. 次のコマンドを入力して、一致したトラフィックに対して実行するアクションを指定します。

```
hostname(config-pmap-c)# {[drop [send-protocol-error] |
drop-connection [send-protocol-error]| mask | reset] [log] | rate-limit message_rate}
```

各 **match** コマンドまたは **class** コマンドですべてのオプションを使用できるわけではありません。使用できる正確なオプションについては、CLI ヘルプまたはコマンド リファレンスを参照してください。

drop キーワードを指定すると、一致するすべてのパケットをドロップします。

send-protocol-error キーワードを指定すると、プロトコル エラー メッセージを送信します。

drop-connection キーワードを指定すると、パケットをドロップし、接続を閉じます。

mask キーワードを指定すると、パケットの一致部分をマスクします。

reset キーワードを指定すると、パケットをドロップして接続を閉じ、サーバとクライアントの両方またはいずれかに TCP リセットを送信します。

log キーワードを指定すると、システム ログ メッセージを送信します。このキーワードは単独で、または他のキーワードのいずれかと一緒に使用できます。

rate-limit *message_rate* 引数では、メッセージのレートを制限します。

ポリシー マップには、複数の **class** コマンドまたは **match** コマンドを指定できます。**class** コマンドと **match** コマンドの順序については、「インスペクション ポリシー マップのアクションの定義」（P.37-3）を参照してください。

ステップ 7　インスペクション エンジンに影響のあるパラメータを設定するには、次の手順を実行します。

a. パラメータ コンフィギュレーション モードに入るには、次のコマンドを入力します。

```
hostname(config-pmap)# parameters
hostname(config-pmap-p)#
```

b. HTTP のプロトコル違反があるかどうかチェックするには、次のコマンドを入力します。

```
hostname(config-pmap-p)# protocol-violation [action [drop-connection | reset | log]]
```

drop-connection アクションを指定すると、接続を閉じます。**reset** アクションを指定すると、接続を閉じ、クライアントに TCP リセットを送信します。**log** アクションを指定すると、ポリシー マップがトラフィックに一致したときにシステム ログ メッセージを送信します。

c. ヘッダーのサーバ フィールドの文字列を置き換えるには、次のコマンドを入力します。

```
hostname(config-pmap-p)# spoof-server string
```

string 引数には、ヘッダーのサーバ フィールドと置き換える文字列を指定します。注：WebVPN のストリームは **spoof-server** コマンドの対象外です。

次の例は、HTTP インスペクション ポリシー マップを定義する方法を示します。この例では、「GET」または「PUT」メソッドで「www\.xyz.com/.*\.asp」または「www\.xyz[0-9][0-9]\.com」にアクセスを試みる HTTP 接続を許可し、ログに記録しています。その他の URL とメソッドの組み合わせはすべて許可され、ログに記録されません。

```
hostname(config)# regex url1 "www\.xyz.com/.*\.asp"
hostname(config)# regex url2 "www\.xyz[0-9][0-9]\.com"
hostname(config)# regex get "GET"
hostname(config)# regex put "PUT"

hostname(config)# class-map type regex match-any url_to_log
hostname(config-cmap)# match regex url1
hostname(config-cmap)# match regex url2
hostname(config-cmap)# exit

hostname(config)# class-map type regex match-any methods_to_log
hostname(config-cmap)# match regex get
hostname(config-cmap)# match regex put
hostname(config-cmap)# exit

hostname(config)# class-map type inspect http http_url_policy
hostname(config-cmap)# match request uri regex class url_to_log
hostname(config-cmap)# match request method regex class methods_to_log
hostname(config-cmap)# exit

hostname(config)# policy-map type inspect http http_policy
hostname(config-pmap)# class http_url_policy
```

```
hostname(config-pmap-c)# log
```

ICMP インスペクション

ICMP インスペクション エンジンを使用すると、ICMP トラフィックが「セッション」を持つようになるため、TCP トラフィックや UDP トラフィックのように検査することが可能になります。ICMP インスペクション エンジンを使用しない場合、アクセス リストで ICMP に ASA の通過を許可しないことをお勧めします。ステートフル インスペクションを実行しないと、ICMP がネットワーク攻撃に利用される可能性があります。ICMP インスペクション エンジンは、要求ごとに応答が 1 つだけであること、シーケンス番号が正しいことを確認します。

ICMP エラー インスペクション

この機能がイネーブルの場合、ASA は、NAT コンフィギュレーションに基づいて ICMP エラー メッセージを送信する中間ホップ用の変換セッションを作成します。ASA は、変換後の IP アドレスでパケットを上書きします。

ディセーブルの場合、ASA は、ICMP エラー メッセージを生成する中間ノード用の変換セッションを作成しません。内部ホストと ASA の間にある中間ノードによって生成された ICMP エラー メッセージは、NAT リソースをそれ以上消費することなく、外部ホストに到達します。外部ホストが traceroute コマンドを使用して ASA の内部にある宛先までのホップをトレースする場合、これは適切ではありません。ASA が中間ホップを変換しない場合、すべての中間ホップは、マッピングされた宛先 IP アドレスとともに表示されます。

ICMP ペイロードがスキャンされて、元のパケットから 5 つのタプルが取得されます。取得した 5 つのタプルを使用してルックアップを実行し、クライアントの元のアドレスを判別します。ICMP エラー インスペクション エンジンは、ICMP パケットに対して次の変更を加えます。

- IP ヘッダー内のマッピング IP を実際の IP（宛先アドレス）に変更し、IP チェックサムを修正する。
- ICMP パケットに変更を加えたため、ICMP ヘッダー内の ICMP チェックサムを修正する。
- ペイロードに次の変更を加える。
  - 元のパケットのマッピング IP を実際の IP に変更する。
  - 元のパケットのマッピング ポートを実際のポートに変更する。
  - 元のパケットの IP チェックサムを再計算する。

インスタント メッセージ インスペクション

この項では、IM インスペクション エンジンについて説明します。この項は、次の内容で構成されています。

- 「インスタント メッセージ インスペクションの概要」（P.47-23）
- 「インスペクション制御を追加するためのインスタント メッセージ インスペクション ポリシーマップの設定」（P.47-23）

インスタント メッセージ インスペクションの概要

インスタント メッセージ（IM）インスペクション エンジンを使用すると、IM アプリケーションの制御を細かく調整して、ネットワークの使用を制御し、機密情報の漏洩やワームの繁殖などの企業のネットワークへの脅威を阻止できます。

インスペクション制御を追加するためのインスタント メッセージ インスペクション ポリシー マップの設定

メッセージがパラメータに違反したときのアクションを指定するには、IM インスペクション ポリシー マップを作成します。作成したインスペクション ポリシー マップは、IM インスペクションをイネーブルにすると適用できます。

IM インスペクション ポリシー マップを作成するには、次の手順を実行します。

ステップ 1　（任意）「正規表現の作成」（P.18-15）に従って、1 つ以上の正規表現をトラフィック照合コマンドに追加して使用できるようにします。ステップ 3 に記載されている **match** コマンドで照合できるテキストのタイプを参照してください。

ステップ 2　（任意）「正規表現クラス マップの作成」（P.18-17）に従って、1 つ以上の正規表現のクラス マップを作成して正規表現をグループ化します。

ステップ 3　（任意）次の手順に従って、IM インスペクションのクラス マップを作成します。

クラス マップは複数のトラフィックとの照合をグループ化します。クラス マップと一致するには、トラフィックは、すべての **match** コマンドと一致する必要があります。または、**match** コマンドを直接ポリシー マップに指定できます。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 マップでトラフィックとの照合を直接定義することの違いは、クラス マップでは複雑な照合基準を作成でき、クラス マップを再利用できるということです。

クラス マップと照合しないトラフィックを指定するには、**match not** コマンドを使用します。たとえば、**match not** コマンドで文字列「example.com」を指定すると、「example.com」が含まれるすべてのトラフィックはクラス マップと照合されません。

このクラス マップで特定するトラフィックに対して、接続のドロップ、リセット、接続のロギングなどのアクションをインスペクション ポリシー マップに指定できます。

match コマンドごとに異なるアクションを実行する場合、ポリシー マップに直接トラフィックを特定する必要があります。

a.　次のコマンドを入力して、クラス マップを作成します。

```
hostname(config)# class-map type inspect im [match-all | match-any] class_map_name
hostname(config-cmap)#
```

class_map_name には、クラス マップの名前を指定します。**match-all** キーワードはデフォルトです。トラフィックがクラス マップと一致するには、すべての基準と一致する必要があることを指定します。**match-any** キーワードは、トラフィックが少なくとも基準の 1 つに一致したらクラス マップと一致することを指定します。CLI がクラスマップ コンフィギュレーション モードに入り、1 つ以上の **match** コマンドを入力できます。

b.　（任意）クラス マップに説明を追加するには、次のコマンドを使用します。

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

string には、クラス マップの説明を 200 文字以内で指定します。

c. （任意）特定の IM プロトコル（Yahoo や MSN など）のトラフィックを照合するには、次のコマンドを入力します。

```
hostname(config-cmap)# match [not] protocol {im-yahoo | im-msn}
```

d. （任意）特定の IM サービス（チャット、ファイル転送、Web カメラ、音声チャット、会議、ゲームなど）を照合するには、次のコマンドを入力します。

```
hostname(config-cmap)# match [not] service {chat | file-transfer | webcam | voice-chat
| conference | games}
```

e. （任意）IM メッセージの送信元のログイン名を照合するには、次のコマンドを入力します。

```
hostname(config-cmap)# match [not] login-name regex {class class_name | regex_name}
```

regex *regex_name* 引数には、ステップ 1 で作成した正規表現を指定します。**class** *regex_class_name* には、ステップ 2 で作成した正規表現のクラス マップを指定します。

f. （任意）IM メッセージの宛先のログイン名を照合するには、次のコマンドを入力します。

```
hostname(config-cmap)# match [not] peer-login-name regex {class class_name |
regex_name}
```

regex *regex_name* 引数には、ステップ 1 で作成した正規表現を指定します。**class** *regex_class_name* には、ステップ 2 で作成した正規表現のクラス マップを指定します。

g. （任意）IM メッセージの送信元の IP アドレスを照合するには、次のコマンドを入力します。

```
hostname(config-cmap)# match [not] ip-address ip_address ip_address_mask
```

ip_address と *ip_address_mask* には、メッセージの送信元の IP アドレスとネットマスクを指定します。

h. （任意）IM メッセージの宛先の IP アドレスを照合するには、次のコマンドを入力します。

```
hostname(config-cmap)# match [not] peer-ip-address ip_address ip_address_mask
```

ip_address と *ip_address_mask* には、メッセージの宛先の IP アドレスとネットマスクを指定します。

i. （任意）IM メッセージのバージョンを照合するには、次のコマンドを入力します。

```
hostname(config-cmap)# match [not] version regex {class class_name | regex_name}
```

regex *regex_name* 引数には、ステップ 1 で作成した正規表現を指定します。**class** *regex_class_name* には、ステップ 2 で作成した正規表現のクラス マップを指定します。

j. （任意）IM メッセージのファイル名を照合するには、次のコマンドを入力します。

```
hostname(config-cmap)# match [not] filename regex {class class_name | regex_name}
```

regex *regex_name* 引数には、ステップ 1 で作成した正規表現を指定します。**class** *regex_class_name* には、ステップ 2 で作成した正規表現のクラス マップを指定します。

（注） MSN IM プロトコルではサポートされていません。

ステップ 4　IM インスペクション ポリシー マップを作成するには、次のコマンドを入力します。

```
hostname(config)# policy-map type inspect im policy_map_name
hostname(config-pmap)#
```

policy_map_name には、ポリシー マップの名前を指定します。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。

ステップ 5　（任意）このポリシー マップに説明を追加するには、次のコマンドを使用します。

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

ステップ 6　次のいずれかの方法を使用して、アクションを実行するトラフィックを指定します。

- 次のコマンドを入力して、ステップ 3 で作成した IM クラス マップを指定します。

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

- ステップ 3 で説明されている **match** コマンドのいずれかを使用して、ポリシー マップに直接トラフィックを指定します。**match not** コマンドを使用すると、**match not** コマンドの基準に一致しないすべてのトラフィックにアクションが適用されます。

ポリシー マップには、複数の **class** コマンドまたは **match** コマンドを指定できます。**class** コマンドと **match** コマンドの順序については、「インスペクション ポリシー マップのアクションの定義」（P.37-3）を参照してください。

ステップ 7　次のコマンドを入力して、一致したトラフィックに対して実行するアクションを指定します。

```
hostname(config-pmap-c)# {drop-connection | reset | log}
```

drop-connection アクションを指定すると、接続を閉じます。**reset** アクションを指定すると、接続を閉じ、クライアントに TCP リセットを送信します。**log** アクションを指定すると、ポリシー マップがトラフィックに一致したときにシステム ログ メッセージを送信します。

次の例は、IM インスペクション ポリシー マップを定義する方法を示しています。

```
hostname(config)# regex loginname1 "ying\@yahoo.com"
hostname(config)# regex loginname2 "Kevin\@yahoo.com"
hostname(config)# regex loginname3 "rahul\@yahoo.com"
hostname(config)# regex loginname4 "darshant\@yahoo.com"
hostname(config)# regex yahoo_version_regex "1\.0"
hostname(config)# regex gif_files ".*\.gif"
hostname(config)# regex exe_files ".*\.exe"

hostname(config)# class-map type regex match-any yahoo_src_login_name_regex
hostname(config-cmap)# match regex loginname1
hostname(config-cmap)# match regex loginname2

hostname(config)# class-map type regex match-any yahoo_dst_login_name_regex
hostname(config-cmap)# match regex loginname3
hostname(config-cmap)# match regex loginname4

hostname(config)# class-map type inspect im match-any yahoo_file_block_list
hostname(config-cmap)# match filename regex gif_files
hostname(config-cmap)# match filename regex exe_files

hostname(config)# class-map type inspect im match-all yahoo_im_policy
hostname(config-cmap)# match login-name regex class yahoo_src_login_name_regex
hostname(config-cmap)# match peer-login-name regex class yahoo_dst_login_name_regex

hostname(config)# class-map type inspect im match-all yahoo_im_policy2
hostname(config-cmap)# match version regex yahoo_version_regex

hostname(config)# class-map im_inspect_class_map
hostname(config-cmap)# match default-inspection-traffic

hostname(config)# policy-map type inspect im im_policy_all
hostname(config-pmap)# class yahoo_file_block_list
hostname(config-pmap-c)# match service file-transfer
```

```
hostname(config-pmap)# class yahoo_im_policy
hostname(config-pmap-c)# drop-connection
hostname(config-pmap)# class yahoo_im_policy2
hostname(config-pmap-c)# reset
hostname(config)# policy-map global_policy_name
hostname(config-pmap)# class im_inspect_class_map
hostname(config-pmap-c)# inspect im im_policy_all
```

IP オプション インスペクション

この項では、IP オプション インスペクション エンジンについて説明します。この項は、次の内容で構成されています。

IP オプション インスペクションの概要

各 IP パケットには、Options フィールドのある IP ヘッダーが含まれています。Options フィールドは、通常は IP オプションと呼ばれ、制御機能を提供します。特定の状況で必要になりますが、一般的な通信では必要ありません。具体的には、IP オプションにはタイムスタンプ、セキュリティ、および特殊なルーティングの規定が含まれています。IP オプションの使用は任意であり、このフィールドにはオプションを 0 個、1 個、またはそれ以上含めることができます。

IP オプション インスペクションを設定して、どの IP パケットが所定の IP オプションを持ち、ASA を通過できるかを制御できます。このインスペクションを設定することで、パケットの転送許可や、指定した IP オプションのクリアが ASA に指示され、パケットの転送が可能になります。

IP オプション インスペクションでは、パケット内の次の 3 つの IP オプションをチェックできます。

- End of Options List（EOOL）または IP Option 0：このオプションにはゼロ バイトが 1 つだけ含まれており、オプションのリストの終わりを示すために、すべてのオプションの末尾に表示されます。これは、ヘッダー長に基づくヘッダーの末尾とは一致しない場合があります。
- No Operation（NOP）または IP Option 1：IP ヘッダーの Options フィールドには、オプションを 0 個、1 個、またはそれ以上含めることができ、これがフィールド変数全体の長さになります。ただし、IP ヘッダーは 32 ビットの倍数である必要があります。全オプションのビット数が 32 ビットの倍数でない場合、32 ビット境界上のオプションと位置合わせするために、NOP オプションは「内部パディング」として使用されます。
- Router Alert（RTRALT）または IP Option 20：このオプションは、中継ルータに対し、パケットの宛先がそのルータでない場合でも、パケットのコンテンツを検査するよう通知します。このインスペクションは、RSVP を実装している場合に役に立ちます。同様のプロトコルは、パケット配信パス上にあるルータでの比較的複雑な処理を必要とします。

（注）　IP オプション インスペクションは、グローバル インスペクション ポリシーにデフォルトで含まれています。したがって、ASA がルーテッド モードの場合、その ASA はパケットに Router Alert オプション（option 20）が含まれた RSVP トラフィックを許可します。

Router Alert オプションが含まれた RSVP パケットをドロップすると、VoIP の実装で問題が生じることがあります。

IP ヘッダーから Router Alert オプションをクリアするように ASA を設定すると、その IP ヘッダーは次のように変更されます。

- Options フィールドは、32 ビット境界で終了するようにパディングされます。
- Internet header length（IHL; インターネット ヘッダーの長さ）が変更されます。
- パケット全体の長さが変更されます。
- チェックサムが再計算されます。

IP ヘッダーに EOOL、NOP、または RTRALT 以外のオプションがさらに含まれている場合、これらのオプションを許可するように ASA が設定されているかどうかに関係なく、ASA はそのパケットをドロップします。

インスペクション制御を追加するための IP オプション インスペクション ポリシー マップの設定

ステップ 1　IP オプションインスペクション ポリシー マップを作成するには、次のコマンドを入力します。

```
hostname(config)# policy-map type inspect ip-options policy_map_name
hostname(config-pmap)#
```

policy_map_name には、ポリシー マップの名前を指定します。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。

ステップ 2　（任意）このポリシー マップに説明を追加するには、次のコマンドを使用します。

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

ステップ 3　インスペクション エンジンに影響のあるパラメータを設定するには、次の手順を実行します。

a. パラメータ コンフィギュレーション モードに入るには、次のコマンドを入力します。

```
hostname(config-pmap)# parameters
hostname(config-pmap-p)#
```

b. End of Options List（EOOL）オプションが含まれたパケットを許可またはクリアするには、次のコマンドを入力します。

```
hostname(config-pmap-p)# eool action {allow | clear}
```

ゼロ バイトが 1 つだけ含まれたこのオプションは、オプションのリストの終わりを示すために、すべてのオプションの末尾に表示されます。これは、ヘッダー長に基づくヘッダーの末尾とは一致しない場合があります。

c. No Operation（NOP）オプションが含まれたパケットを許可またはクリアするには、次のコマンドを入力します。

```
hostname(config-pmap-p)# nop action {allow | clear}
```

IP ヘッダーの Options フィールドには、オプションを 0 個、1 個、またはそれ以上含めることができ、これがフィールド変数全体の長さになります。ただし、IP ヘッダーは 32 ビットの倍数である必要があります。全オプションのビット数が 32 ビットの倍数でない場合、32 ビット境界上のオプションと位置合わせするために、NOP オプションは「内部パディング」として使用されます。

d. Router Alert（RTRALT）オプションが含まれたパケットを許可またはクリアするには、次のコマンドを入力します。

```
hostname(config-pmap-p)# router-alert action {allow | clear}
```

このオプションは、中継ルータに対し、パケットの宛先がそのルータでない場合でも、パケットのコンテンツを検査するよう通知します。このインスペクションは、RSVP を実装している場合に役に立ちます。同様のプロトコルは、パケット配信パス上にあるルータでの比較的複雑な処理を必要とします。

（注）　ASA でパケットを許可する前に、パケットから IP オプションをクリアするには、**clear** コマンドを入力します。

IPsec パススルー インスペクション

この項では、IPSec パススルー インスペクション エンジンについて説明します。この項は、次の内容で構成されています。

IPsec パススルー インスペクションの概要

Internet Protocol Security（IPsec）は、データ ストリームの各 IP パケットを認証および暗号化することによって、IP 通信をセキュリティで保護するためのプロトコル スイートです。IPSec には、セッションの開始時、およびセッション中に使用される暗号キーのネゴシエーションの開始時に、エージェント間の相互認証を確立するためのプロトコルも含まれています。IPSec を使用して、ホスト（コンピュータ ユーザまたはサーバなど）のペア間、セキュリティ ゲートウェイ（ルータやファイアウォールなど）のペア間、またはセキュリティ ゲートウェイとホスト間のデータ フローを保護できます。

IPsec パススルー アプリケーション インスペクションは、IKE UDP ポート 500 接続に関連付けられた ESP（IP プロトコル 50）および AH（IP プロトコル 51）トラフィックを簡単に横断できます。このインスペクションは、冗長なアクセス リスト コンフィギュレーションを回避して ESP および AH トラフィックを許可し、タイムアウトと最大接続数を使用してセキュリティも確保します。

インスペクション用パラメータの定義に使用する特定のマップを識別するには、IPsec パススルー インスペクション パラメータを指定します。所定の IPSec パススルー検査のポリシーマップを設定し、パラメータ コンフィギュレーションにアクセスします。このコンフィギュレーションでは、ESP または AH トラフィックの制限を指定できます。パラメータ コンフィギュレーションでは、クライアントあたりの最大接続数と、アイドル タイムアウトを設定できます。

NAT および非 NAT トラフィックは許可されます。ただし、PAT はサポートされません。

IPSec パススルー パラメータ マップの定義例

次に、アクセス リストを使用して IKE トラフィックを識別し、IPsec Pass Thru パラメータ マップを定義して、ポリシーを定義し、外部インターフェイスにポリシーを適用する例を示します。

```
hostname(config)# access-list ipsecpassthruacl permit udp any any eq 500
hostname(config)# class-map ipsecpassthru-traffic
hostname(config-cmap)# match access-list ipsecpassthruacl
hostname(config)# policy-map type inspect ipsec-pass-thru iptmap
hostname(config-pmap)# parameters
hostname(config-pmap-p)# esp per-client-max 10 timeout 0:11:00
```

```
hostname(config-pmap-p)# ah per-client-max 5 timeout 0:06:00
hostname(config)# policy-map inspection_policy
hostname(config-pmap)# class ipsecpassthru-traffic
hostname(config-pmap-c)# inspect ipsec-pass-thru iptmap
hostname(config)# service-policy inspection_policy interface outside
```

IPv6 インスペクション

MPF ルールを使用して IPv6 インスペクションを設定し、拡張ヘッダーに基づいて IPv6 トラフィックを選択的にブロックできます。IPv6 パケットに対しては、早期セキュリティ チェックが実行されます。ASA は、ルータ ヘッダーとノー ネクスト ヘッダーをブロックする一方で、常に、ホップバイホップと宛先オプション タイプの拡張ヘッダーを通過させます。

デフォルトの IPv6 インスペクションをイネーブルにする、または IPv6 インスペクションを定義することができます。IPv6 インスペクションの MPF ポリシー マップを定義することで、IPv6 パケットにある拡張ヘッダーに含まれる、次に示すタイプに基づいて、選択的に IPv6 パケットをドロップするように ASA を設定できます。

- ホップバイホップ オプション
- ルーティング（タイプ 0）
- フラグメント
- 宛先オプション
- 認証
- 暗号ペイロード

デフォルト IPv6 インスペクションでは、IPv6 パケット内の拡張ヘッダーのタイプと順序が RFC 2460 に準拠しているかどうかについてのチェックも実行されます。

- IPv6 ヘッダー
- Hop-by-Hop オプション ヘッダー（0）
- 宛先オプション ヘッダー（60）
- ルーティング ヘッダー（43）
- フラグメント ヘッダー（44）
- 認証（51）
- カプセル化セキュリティ ペイロード ヘッダー（50）
- 宛先オプション ヘッダー（60）
- ノー ネクスト ヘッダー（59）

ポリシー マップが IPv6 インスペクション用に設定されていないか、または設定済みのポリシー マップがインターフェイスに関連付けられていない場合、ASA は、任意のモビリティ タイプで、ルーティング タイプの IPv6 拡張ヘッダーのある、インターフェイスに到着したパケットをドロップします。

IPv6 インスペクション ポリシー マップの作成時に、ASA は、0 ～ 255 の範囲のヘッダー ルーティング タイプに一致するパケットをドロップする設定を自動的に生成します。

IPv6 インスペクション ポリシー マップの設定

IPv6 拡張ヘッダーを処理する IPv6 インスペクション ポリシー マップを設定できます。IPv6 ポリシー マップは、指定された方向の分類された IPv6 パケットごとに適用されます。現在、IPv6 トラフィックのみを検査できます。

NETBIOS インスペクション

この項では、IM インスペクション エンジンについて説明します。この項は、次の内容で構成されています。

NETBIOS インスペクションの概要

NETBIOS インスペクションはデフォルトでイネーブルになっています。NetBios インスペクション エンジンは、ASA の NAT コンフィギュレーションに基づいて、NetBios Name Service（NBNS; NetBios ネーム サービス）パケット内の IP アドレスを変換します。

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

メッセージがパラメータに違反したときのアクションを指定するには、NetBIOS インスペクション ポリシー マップを作成します。作成したインスペクション ポリシー マップは、NETBIOS インスペクションをイネーブルにすると適用できます。

NetBIOS インスペクション ポリシー マップを作成するには、次の手順を実行します。

ステップ 1　（任意）「正規表現の作成」（P.18-15）に従って、1 つ以上の正規表現をトラフィック照合コマンドに追加して使用できるようにします。ステップ 3 に記載されている **match** コマンドで照合できるテキストのタイプを参照してください。

ステップ 2　（任意）「正規表現クラス マップの作成」（P.18-17）に従って、1 つ以上の正規表現のクラス マップを作成して正規表現をグループ化します。

ステップ 3　NetBIOS インスペクション ポリシー マップを作成するには、次のコマンドを入力します。

```
hostname(config)# policy-map type inspect netbios policy_map_name
hostname(config-pmap)#
```

policy_map_name には、ポリシー マップの名前を指定します。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。

ステップ 4　（任意）このポリシー マップに説明を追加するには、次のコマンドを使用します。

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

ステップ 5　一致したトラフィックにアクションを適用するには、次の手順を実行します。

a. 次のいずれかの方法を使用して、アクションを実行するトラフィックを指定します。

- 次のコマンドを入力して、ステップ 3 で作成した NetBIOS クラス マップを指定します。

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

- ステップ 3 で説明されている **match** コマンドのいずれかを使用して、ポリシー マップに直接トラフィックを指定します。**match not** コマンドを使用すると、**match not** コマンドの基準に一致しないすべてのトラフィックにアクションが適用されます。

b. 次のコマンドを入力して、一致したトラフィックに対して実行するアクションを指定します。

```
hostname(config-pmap-c)# {[drop [send-protocol-error] |
drop-connection [send-protocol-error]| mask | reset] [log] | rate-limit message_rate}
```

各 **match** コマンドまたは **class** コマンドですべてのオプションを使用できるわけではありません。使用できる正確なオプションについては、CLI ヘルプまたはコマンド リファレンスを参照してください。

drop キーワードを指定すると、一致するすべてのパケットをドロップします。

send-protocol-error キーワードを指定すると、プロトコル エラー メッセージを送信します。

drop-connection キーワードを指定すると、パケットをドロップし、接続を閉じます。

mask キーワードを指定すると、パケットの一致部分をマスクします。

reset キーワードを指定すると、パケットをドロップして接続を閉じ、サーバとクライアントの両方またはいずれかに TCP リセットを送信します。

log キーワードを指定すると、システム ログ メッセージを送信します。このキーワードは単独で、または他のキーワードのいずれかと一緒に使用できます。

rate-limit *message_rate* 引数では、メッセージのレートを制限します。

ポリシー マップには、複数の **class** コマンドまたは **match** コマンドを指定できます。**class** コマンドと **match** コマンドの順序については、「インスペクション ポリシー マップのアクションの定義」（P.37-3）を参照してください。

ステップ 6　インスペクション エンジンに影響のあるパラメータを設定するには、次の手順を実行します。

a. パラメータ コンフィギュレーション モードに入るには、次のコマンドを入力します。

```
hostname(config-pmap)# parameters
hostname(config-pmap-p)#
```

b. NetBIOS のプロトコル違反があるかどうかチェックするには、次のコマンドを入力します。

```
hostname(config-pmap-p)# protocol-violation [action [drop-connection | reset | log]]
```

drop-connection アクションを指定すると、接続を閉じます。**reset** アクションを指定すると、接続を閉じ、クライアントに TCP リセットを送信します。**log** アクションを指定すると、ポリシー マップがトラフィックに一致したときにシステム ログ メッセージを送信します。

次の例は、NetBIOS インスペクション ポリシー マップを定義する方法を示しています。

```
hostname(config)# policy-map type inspect netbios netbios_map
hostname(config-pmap)# protocol-violation drop log

hostname(config)# policy-map netbios_policy
hostname(config-pmap)# class inspection_default
hostname(config-pmap-c)# inspect netbios netbios_map
```

PPTP インスペクション

PPTP は、PPP トラフィックのトンネリングに使用されるプロトコルです。PPTP セッションは、1 つの TCP チャネルと通常 2 つの PPTP GRE トンネルで構成されます。TCP チャネルは、PPTP GRE トンネルのネゴシエートと管理に使用される制御チャネルです。GRE トンネルは、2 つのホスト間の PPP セッションを伝送します。

PPTP アプリケーション インスペクションは、イネーブルになると、PPTP プロトコル パケットを検査し、PPTP トラフィックを許可するために必要な GRE 接続と xlate をダイナミックに作成します。RFC 2637 で定義されているバージョン 1 だけがサポートされます。

PAT は、PPTP TCP 制御チャネル上で修正バージョンの GRE（RFC 2637）がネゴシエートされたときに、その GRE に対してだけ実行されます。PAT は、未修正バージョンの GRE（RFC 1701、RFC 1702）には実行*されません*。

具体的には、ASA は、PPTP のバージョン通知と発信コールの要求/応答シーケンスを検査します。RFC 2637 で定義されている PPTP バージョン 1 だけが検査されます。どちらかの側から通知されたバージョンがバージョン 1 でない場合、TCP 制御チャネルでのそれ以降のインスペクションはディセーブルになります。また、発信コールの要求と応答のシーケンスは追跡されます。接続と xlate は、後続のセカンダリ GRE データ トラフィックを許可するために、必要に応じてダイナミックに割り当てられます。

PPTP インスペクション エンジンは、PPTP トラフィックを PAT で変換できるように、イネーブルにする必要があります。また、PAT は、PPTP TCP 制御チャネルで修正バージョンの GRE（RFC 2637）がネゴシエートされた場合に限り、その GRE に対してだけ実行されます。PAT は、未修正バージョンの GRE（RFC 1701、RFC 1702）には実行されません。

RFC 2637 で定義されているように、PPTP プロトコルは、主に、モデム バンク PPTP Access Concentrator（PAC; PPTP アクセス コンセントレータ）から開始されたヘッドエンド PPTP Network Server（PNS; PPTP ネットワーク サーバ）への PPP セッションのトンネリングに使用されます。このように使用された場合、PAC がリモート クライアントで PNS がサーバです。

ただし、Windows によって VPN で使用された場合、この関係は逆になります。PNS は、中央のネットワークにアクセスするためにヘッドエンド PAC への接続を開始するリモートのシングル ユーザ PC です。

SMTP および拡張 SMTP インスペクション

この項では、IM インスペクション エンジンについて説明します。この項は、次の内容で構成されています。

SMTP および拡張 SMTP（ESMTP）のインスペクションの概要

ESMTP アプリケーション インスペクションを使用すると、ASA を通過できる SMTP コマンドの種類を制限し、モニタ機能を追加することによって、SMTP ベースの攻撃からより強固に保護できます。

ESMTP は SMTP プロトコルの拡張で、ほとんどの観点で SMTP に似ています。便宜上、このマニュアルでは、SMTP という用語を SMTP と ESMTP の両方に使用します。拡張 SMTP に対するアプリケーション インスペクション処理は、SMTP アプリケーション インスペクションに似ており、SMTP セッションのサポートが含まれています。拡張 SMTP セッションで使用するほとんどのコマンドは、SMTP セッションで使用するコマンドと同じですが、ESMTP セッションの方が大幅に高速で、配信ステータス通知など信頼性およびセキュリティに関するオプションが増えています。

拡張 SMTP アプリケーション インスペクションでは、AUTH、EHLO、ETRN、HELP、SAML、SEND、SOML、STARTTLS、および VRFY を含む拡張 SMTP コマンドに対するサポートが追加されています。ASA は、7 つの RFC 821 コマンド（DATA、HELO、MAIL、NOOP、QUIT、RCPT、RSET）をサポートするとともに、合計 15 の SMTP コマンドをサポートします。

その他の拡張 SMTP コマンド（ATRN、ONEX、VERB、CHUNKING など）、およびプライベート拡張はサポートされません。サポートされないコマンドは、内部サーバにより拒否される X に変換されます。この結果は、「500 Command unknown: 'XXX'」のようなメッセージで表示されます。不完全なコマンドは、廃棄されます。

ESMTP インスペクション エンジンは、文字「2」、「0」、「0」を除くサーバの SMTP バナーの文字をアスタリスクに変更します。Carriage Return（CR; 復帰）、および Linefeed（LF; 改行）は無視されます。

SMTP インスペクションをイネーブルにする場合、次のルールに従わないと、対話型の SMTP に使用する Telnet セッションが停止することがあります。SMTP コマンドの長さは 4 文字以上にする必要があります。復帰と改行で終了する必要があります。次の応答を発行する前に現在の応答を待機する必要があります。

SMTP サーバは、数値の応答コード、およびオプションの可読文字列でクライアント要求に応答します。SMTP アプリケーション インスペクションは、ユーザが使用できるコマンドとサーバが返送するメッセージを制御し、その数を減らします。SMTP インスペクションは、次の 3 つの主要なタスクを実行します。

- SMTP 要求を 7 つの基本 SMTP コマンドと 8 つの拡張コマンドに制限します。
- SMTP コマンド応答シーケンスをモニタします。
- 監査証跡の生成：メール アドレス内に埋め込まれている無効な文字が置き換えられたときに、監査レコード 108002 を生成します。詳細については、RFC 821 を参照してください。

SMTP インスペクションでは、次の異常な署名がないかどうか、コマンドと応答のシーケンスをモニタします。

- 切り捨てられたコマンド
- 不正なコマンド終端（<CR><LR> で終了していない）
- MAIL コマンドと RCPT コマンドでは、メールの送信者と受信者が指定されます。異常な文字がないか、メール アドレスがスキャンされます。パイプ（|）が削除（空白スペースに変更）され、「<」および「>」については、メール アドレスの定義に使用される場合だけ許可されます（「<」の後には、必ず「>」が使用されている必要があります）。
- SMTP サーバによる不意の移行
- 未知のコマンドに対しては、ASA はパケット内のすべての文字を X に変更します。この場合、サーバがクライアントに対してエラー コードを生成します。パケット内が変更されているため、TCP チェックサムは、再計算または調節する必要があります。
- TCP ストリーム編集
- コマンド パイプライン

インスペクション制御を追加するための ESMTP インスペクション ポリシー マップの設定

ESMTP インスペクションは、スパム、フィッシング、不正な形式のメッセージによる攻撃、バッファ オーバーフロー / アンダーフロー攻撃を検出します。また、アプリケーション セキュリティとプロトコル準拠により、正常な ESMTP メッセージだけを通し、各種の攻撃の検出や送受信者およびメール中継のブロックも行います。

メッセージがパラメータに違反したときのアクションを指定するには、ESMTP インスペクション ポリシー マップを作成します。作成したインスペクション ポリシー マップは、ESMTP インスペクションをイネーブルにすると適用できます。

ESMTP インスペクション ポリシー マップを作成するには、次の手順を実行します。

ステップ 1　（任意）「正規表現の作成」（P.18-15）に従って、1 つ以上の正規表現をトラフィック照合コマンドに追加して使用できるようにします。ステップ 3 に記載されている **match** コマンドで照合できるテキストのタイプを参照してください。

ステップ 2　（任意）「正規表現クラス マップの作成」（P.18-17）に従って、1 つ以上の正規表現のクラス マップを作成して正規表現をグループ化します。

ステップ 3　ESMTP インスペクション ポリシー マップを作成するには、次のコマンドを入力します。

```
hostname(config)# policy-map type inspect esmtp policy_map_name
hostname(config-pmap)#
```

policy_map_name には、ポリシー マップの名前を指定します。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。

ステップ 4　（任意）このポリシー マップに説明を追加するには、次のコマンドを使用します。

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

ステップ 5　一致したトラフィックにアクションを適用するには、次の手順を実行します。

a. 次のいずれかの方法を使用して、アクションを実行するトラフィックを指定します。

- 次のコマンドを入力して、ステップ 3 で作成した ESMTP クラス マップを指定します。

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

- ステップ 3 で説明されている **match** コマンドのいずれかを使用して、ポリシー マップに直接トラフィックを指定します。**match not** コマンドを使用すると、**match not** コマンドの基準に一致しないすべてのトラフィックにアクションが適用されます。

b. 次のコマンドを入力して、一致したトラフィックに対して実行するアクションを指定します。

```
hostname(config-pmap-c)# {[drop [send-protocol-error] |
drop-connection [send-protocol-error]| mask | reset] [log] | rate-limit message_rate}
```

各 **match** コマンドまたは **class** コマンドですべてのオプションを使用できるわけではありません。使用できる正確なオプションについては、CLI ヘルプまたはコマンド リファレンスを参照してください。

drop キーワードを指定すると、一致するすべてのパケットをドロップします。

send-protocol-error キーワードを指定すると、プロトコル エラー メッセージを送信します。

drop-connection キーワードを指定すると、パケットをドロップし、接続を閉じます。

mask キーワードを指定すると、パケットの一致部分をマスクします。

reset キーワードを指定すると、パケットをドロップして接続を閉じ、サーバとクライアントの両方またはいずれかに TCP リセットを送信します。

log キーワードを指定すると、システム ログ メッセージを送信します。このキーワードは単独で、または他のキーワードのいずれかと一緒に使用できます。

rate-limit *message_rate* 引数では、メッセージのレートを制限します。

ポリシー マップには、複数の **class** コマンドまたは **match** コマンドを指定できます。**class** コマンドと **match** コマンドの順序については、「インスペクション ポリシー マップのアクションの定義」（P.37-3）を参照してください。

ステップ 6　インスペクション エンジンに影響のあるパラメータを設定するには、次の手順を実行します。

a. パラメータ コンフィギュレーション モードに入るには、次のコマンドを入力します。

```
hostname(config-pmap)# parameters
hostname(config-pmap-p)#
```

b. ローカル ドメイン名を設定するには、次のコマンドを入力します。

```
hostname(config-pmap-p)# mail-relay domain-name action [drop-connection | log]]
```

drop-connection アクションを指定すると、接続を閉じます。**log** アクションを指定すると、ポリシー マップがトラフィックに一致したときにシステム ログ メッセージを送信します。

c. バナーを難読化するには、次のコマンドを入力します。

```
hostname(config-pmap-p)# mask-banner
```

次の例は、ESMTP インスペクション ポリシー マップを定義する方法を示しています。

```
hostname(config)# regex user1 "user1@cisco.com"
hostname(config)# regex user2 "user2@cisco.com"
hostname(config)# regex user3 "user3@cisco.com"
hostname(config)# class-map type regex senders_black_list
hostname(config-cmap)# description egular expressions to filter out undesired senders"
hostname(config-cmap)# match regex user1
hostname(config-cmap)# match regex user2
hostname(config-cmap)# match regex user3

hostname(config)# policy-map type inspect esmtp advanced_esmtp_map
hostname(config-pmap)# match sender-address regex class senders_black_list
hostname(config-pmap-c)# drop-connection log

hostname(config)# policy-map outside_policy
hostname(config-pmap)# class inspection_default
hostname(config-pmap-c)# inspect esmtp advanced_esmtp_map

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

TFTP インスペクション

TFTP インスペクションはデフォルトでイネーブルになっています。

TFTP は、RFC 1350 に記述されているように、TFTP サーバとクライアントの間のファイルの読み書きを行うための簡易プロトコルです。

ASA は、TFTP トラフィックを検査し、必要に応じてダイナミックに接続と変換を作成し、TFTP クライアントとサーバの間のファイル転送を許可します。具体的には、インスペクション エンジンは TFTP Read Request（RRQ; 読み取り要求）、Write Request（WRQ; 書き込み要求）、およびエラー通知（ERROR）を検査します。

有効な読み取り要求（RRQ）または書き込み要求（WRQ）を受信すると、必要に応じて、ダイナミックなセカンダリ チャネルと PAT 変換が割り当てられます。このセカンダリ チャネルは、これ以降 TFTP によってファイル転送またはエラー通知用に使用されます。

TFTP サーバだけがセカンダリ チャネル経由のトラフィックを開始できます。また、TFTP クライアントとサーバの間に存在できる不完全なセカンダリ チャネルは 1 つまでです。サーバからのエラー通知があると、セカンダリ チャネルは閉じます。

TFTP トラフィックのリダイレクトにスタティック PAT が使用されている場合は、TFTP インスペクションをイネーブルにする必要があります。