基本インターネット プロトコルのインスペクション

ここでは、基本インターネット プロトコルのアプリケーション インスペクションについて説明します。特定のプロトコルに関してインスペクションを使用する必要がある理由、およびインスペクションを適用する全体的な方法については、「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準備」（P.7-1）を参照してください。

DNS インスペクション

ここでは、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。

DNS インスペクションのアクション

DNS 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。DNS インスペクションをカスタマイズして多くのタスクを実行できます。

- DNS レコードを NAT の設定に基づいて変換します。詳細については、「DNS および NAT」（P.4-33）を参照してください。
- メッセージの長さ、ドメイン名の長さ、ラベルの長さを適用します。
- 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存在するかどうかを確認します。
- DNS のヘッダー、タイプ、クラス、その他に基づいてパケットを検査します。

DNS インスペクションのデフォルト

DNS インスペクションは、次のような preset_dns_map インスペクション クラス マップを使用して、デフォルトでイネーブルになっています。

- 最大 DNS メッセージ長は、512 バイトです。
- 最大クライアント DNS メッセージ長は、リソース レコードに一致するように自動的に設定されます。
- DNS ガードはイネーブルになり、ASA によって 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ている 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。
- NAT の設定に基づく DNS レコードの変換はイネーブルです。
- プロトコルの強制はイネーブルであり、DNS メッセージ形式チェックが行われます。ドメイン名の長さが 255 文字以下、ラベルの長さが 63 文字、圧縮、ループ ポインタのチェックなどです。

次のデフォルトの DNS インスペクション コマンドを参照してください。

```
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
! ...
service-policy global_policy global
```

DNS インスペクションの設定

DNS 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。デフォルト以外の処理が必要な場合にのみ設定する必要があります。DNS インスペクションをカスタマイズする場合は、次のプロセスを使用します。

手順

ステップ 1　「DNS インスペクション ポリシー マップの設定」（P.8-3）。

ステップ 2　「DNS インスペクション サービス ポリシーの設定」（P.8-7）。

DNS インスペクション ポリシー マップの設定

デフォルトのインスペクション動作がネットワークにとって十分でない場合、DNS インスペクション ポリシー マップを作成して DNS インスペクション アクションをカスタマイズできます。

トラフィックの一致基準を定義するときに、クラス マップを作成するか、またはポリシー マップに match ステートメントを直接含めることができます。次の手順では、両方の方法について説明します。

はじめる前に

一部のトラフィック照合オプションでは、照合のために正規表現を使用します。これらのテクニックの 1 つを使用する場合は、最初に正規表現または正規表現のクラス マップを作成します。

手順

ステップ 1　（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照合をグループ化します。代わりに、ポリシー マップで **match** コマンドを直接指定できます。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 マップで指定するトラフィックに対しては、インスペクション ポリシー マップで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。

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 1 つの **match** ステートメントと一致したらクラス 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. 次のいずれかの **match** コマンドを使用して、アクションを実行するトラフィックを指定します。**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。

- **match** [**not**] **header-flag** [**eq**] {*f_name* [*f_name...*] | *f_value*}：DNS フラグと一致します。*f_name* 引数は DNS フラグ名であり、**AA**（権限応答）、**QR**（クエリー）、**RA**（使用できる再帰）、**RD**（必要な再帰）、**TC**（切り捨て）のいずれかです。*f_value* 引数は、0x で始まる 16 ビットの 16 進値です（0x0 ～ 0xffff）。**eq** キーワードは完全一致を指定します（すべて一致）。**eq** キーワードを指定しないと、パケットは指定されているヘッダーの 1 つと一致するだけで十分です（いずれかと一致）。例：**match header-flag AA QR**。
- **match** [**not**] **dns-type** {**eq** {*t_name* | *t_value*} | **range** *t_value1 t_value2*}：DNS タイプと一致します。*t_name* 引数は DNS タイプ名であり、次のいずれかです。**A**（IPv4 アドレス）、**AXFR**（フル ゾーン転送）、**CNAME**（正規の名前）、**IXFR**（増分ゾーン転送）、**NS**（権限ネーム サーバ）、**SOA**（権限ゾーンの開始）、**TSIG**（トランザクション署名）です。*t_value* 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。例：**match dns-type eq A**。
- **match** [**not**] **dns-class** {**eq** {**in** | *c_value*} | **range** *c_value1 c_value2*}：DNS クラスと一致します。クラスは **in**（インターネットの場合）または *c_value*（DNS クラス フィールドの 0 ～ 65535 の任意の値）です。**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。例：**match dns-class eq in**。
- **match** [**not**] {**question** | **resource-record** {**answer** | **authority** | **additional**}}：DNS の質問またはリソース レコードと一致します。**question** キーワードは、DNS メッセージの問い合わせ部分を指定します。**resource-record** キーワードは、リソース レコードのセクション **answer**、**authority**、**additional** のいずれかを指定します。例：**match resource-record answer**。
- **match** [**not**] **domain-name regex** {*regex_name* | **class** *class_name*}：DNS メッセージのドメイン名のリストを、指定された正規表現または正規表現クラスに対して照合します。

d. クラス マップ コンフィギュレーション モードを終了するには、「**exit**」と入力します。

ステップ 2　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 3　（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 4　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。

- DNS クラス マップを作成した場合は、次のコマンドを入力してそれを指定します。

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

- DNS クラス マップで説明されている **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 [log] | drop-connection [log]|
enforce-tsig {[drop] [log]} | mask [log] | log}
```

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。

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。このアクションは、ヘッダー フラグの照合だけで利用可能です。

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。

enforce-tsig {**[drop] [log]**} キーワードは、メッセージに TSIG リソース レコードが存在することを強制します。TSIG リソース レコードがないパケットをドロップ、ログ記録、またはドロップしてログ記録できます。ヘッダー フラグ一致の場合、このオプションをマスク アクションと組み合わせて使用できます。それ以外の場合、このアクションと他のアクションを同時に指定することはできません。

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.2-4）を参照してください。

次に例を示します。

```
hostname(config)# policy-map type inspect dns dns-map
hostname(config-pmap)# class dns-class-map
hostname(config-pmap-c)# drop
hostname(config-pmap-c)# match header-flag eq aa
hostname(config-pmap-c)# drop log
```

ステップ 5　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. 1 つまたは複数のパラメータを設定します。次のオプションを設定できます。オプションをディセーブルにするには、コマンドの **no** 形式を使用してください。

- **dns-guard**：DNS ガードをイネーブルにします。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。

- **id-mismatch count** *number* **duration** *seconds* **action log**：DNS ID の過剰な不一致のロギングをイネーブルにします。**count** *number* **duration** *seconds* 引数は、システム メッセージ ログが送信されるようになる 1 秒間の不一致インスタンスの最大数を指定します。
- **id-randomization**：DNS クエリーの DNS 識別子をランダム化します。
- **message-length maximum** {*length* | **client** {*length* | **auto**} | **server** {*length* | **auto**}}：DNS メッセージの最大長を設定します（512 ～ 65535 バイト）。クライアント メッセージまたはサーバ メッセージの最大長も設定できます。**auto** キーワードは、リソース レコードの値に最大長を設定します。
- **nat-rewrite**：DNS レコードを NAT の設定に基づいて変換します。
- **protocol-enforcement**：DNS メッセージ形式のチェックをイネーブルにします。ドメイン名の長さが 255 文字以下、ラベルの長さが 63 文字、圧縮、ループ ポインタのチェックなどです。
- **tsig enforced action** {[**drop**] [**log**]}：TSIG リソース レコードの存在を要求します。準拠していないパケットをドロップしたり（**drop**）、パケットをログに記録したり（**log**）できます。両方指定することもできます。

次に例を示します。

```
hostname(config-pmap)# parameters
hostname(config-pmap-p)# dns-guard
hostname(config-pmap-p)# message-length maximum 1024
hostname(config-pmap-p)# nat-rewrite
hostname(config-pmap-p)# protocol-enforcement
```

例

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。

```
regex domain_example "example\.com"
regex domain_foo "foo\.com"

! define the domain names that the server serves
class-map type inspect regex match-any my_domains
    match regex domain_example
    match regex domain_foo

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

policy-map type inspect dns new_dns_map
    class pub_server_map
        drop log
    match header-flag RD
        mask log
    parameters
        message-length maximum client auto
        message-length maximum 512
        dns-guard
        protocol-enforcement
        nat-rewrite
```

DNS インスペクション サービス ポリシーの設定

ASA のデフォルトの設定には、すべてのインターフェイスにグローバルに適用されるデフォルト ポートの DNS インスペクションが含まれます。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。

手順

ステップ 1　必要な場合は、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**）。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。

照合ステートメントについては、「トラフィックの特定（レイヤ 3/4 クラス マップ）」（P.1-14）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　DNS 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。

ステップ 4　DNS インスペクションを設定します。

inspect dns [*dns_policy_map*] [**dynamic-filter-snoop**]

それぞれの説明は次のとおりです。

- *dns_policy_map* は、オプションの DNS インスペクション ポリシー マップです。デフォルト以外のインスペクション処理が必要な場合にのみマップが必要です。DNS インスペクション ポリシー マップの作成の詳細については、「DNS インスペクション ポリシー マップの設定」（P.8-3）を参照してください。
- **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。

例：

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

（注） デフォルトのグローバル ポリシー（または使用中の任意のポリシー）を編集して、異なる DNS インスペクション ポリシー マップを使用する場合は（たとえば、デフォルトの preset_dns_map を置き換える）、**no inspect dns** コマンドで DNS インスペクションを除去した後、新しい DNS インスペクション ポリシー マップ名を指定して再度追加します。

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 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。

例

次の例では、グローバル デフォルト設定で新しいインスペクション ポリシー マップを使用する方法を示します。

```
policy-map global_policy
 class inspection_default
  no inspect dns preset_dns_map
  inspect dns new_dns_map
service-policy global_policy global
```

DNS インスペクションのモニタリング

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

ここでは、FTP インスペクション エンジンについて説明します。

- 「FTP インスペクションの概要」（P.8-9）
- 「厳密な FTP」（P.8-10）
- 「FTP インスペクションの設定」（P.8-11）
- 「FTP インスペクションの確認とモニタリング」（P.8-15）

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。

厳密な FTP

厳密な FTP 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くなるため、保護されたネットワークのセキュリティが強化されます。厳密な FTP をイネーブルにするには、**inspect ftp** コマンドに **strict** オプションを含めます。

厳密な FTP を使用するときは、オプションで FTP インスペクション ポリシー マップを指定して、ASA を通過することが許可されない FTP コマンドを指定できます。

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。

FTP インスペクションの設定

FTP 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。デフォルト以外の処理が必要な場合にのみ設定する必要があります。FTP インスペクションをカスタマイズする場合は、次のプロセスを使用します。

手順

ステップ 1　「FTP インスペクション ポリシー マップの設定」（P.8-11）。

ステップ 2　「FTP インスペクション サービス ポリシーの設定」（P.8-14）。

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 インスペクション ポリシー マップを作成および設定します。作成したマップは、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。

はじめる前に

一部のトラフィック照合オプションでは、照合のために正規表現を使用します。これらのテクニックの 1 つを使用する場合は、最初に正規表現または正規表現のクラス マップを作成します。

手順

ステップ 1　（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照合をグループ化します。代わりに、ポリシー マップで **match** コマンドを直接指定できます。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 マップで指定するトラフィックに対しては、インスペクション ポリシー マップで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。

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 1 つの **match** ステートメントと一致したらクラス 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. 次のいずれかの **match** コマンドを使用して、アクションを実行するトラフィックを指定します。**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。

- **match** [**not**] **filename regex** {*regex_name* | **class** *class_name*}：FTP 転送のファイル名を、指定された正規表現または正規表現クラスに対して照合します。
- **match** [**not**] **filetype regex** {*regex_name* | **class** *class_name*}：FTP 転送のファイル タイプを、指定された正規表現または正規表現クラスに対して照合します。
- **match** [**not**] **request-command** *ftp_command* [*ftp_command...*]：FTP コマンドを照合します。以下の 1 つ以上です。

  APPE：ファイルに追加します。

  CDUP：現在の作業ディレクトリの親ディレクトリに変更します。

  DELE：サーバのファイルを削除します。

  GET：サーバからファイルを取得します。

  HELP：ヘルプ情報を提供します。

  MKD：サーバにディレクトリを作成します。

  PUT：ファイルをサーバに送信します。

  MKD：サーバのディレクトリを削除します。

  RNFR：「変更前の」ファイル名を指定します。

  RNTO：「変更後の」ファイル名を指定します。

  SITE：サーバ固有のコマンドの指定に使用されます。通常、これはリモート管理に使用されます。

  STOU：一義的なファイル名を使用してファイルを保存します。

- **match** [**not**] **server regex** {*regex_name* | **class** *class_name*}：FTP サーバ名を、指定された正規表現または正規表現クラスに対して照合します。
- **match** [**not**] **username regex** {*regex_name* | **class** *class_name*}：FTP ユーザ名を、指定された正規表現または正規表現クラスに対して照合します。

d. クラス マップ コンフィギュレーション モードを終了するには、「**exit**」と入力します。

ステップ 2　FTP インスペクション ポリシー マップを作成します。

```
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 3　（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 4　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。

- FTP クラス マップを作成した場合は、次のコマンドを入力してそれを指定します。

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

- FTP クラス マップで説明されている **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)# reset [log]
```

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、サーバまたはクライアントに TCP リセットを送信します。システム ログ メッセージを送信するには、**log** キーワードを追加します。

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.2-4）を参照してください。

ステップ 5　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. 1 つまたは複数のパラメータを設定します。次のオプションを設定できます。オプションをディセーブルにするには、コマンドの **no** 形式を使用してください。

- **mask-banner**：FTP サーバから接続時バナーをマスクします。
- **mask-syst-reply**：**syst** コマンドに対する応答をマスクします。

例

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 インスペクション サービス ポリシーの設定

ASA のデフォルトの設定には、すべてのインターフェイスにグローバルに適用されるデフォルト ポートの FTP インスペクションが含まれます。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。

手順

ステップ 1　必要な場合は、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 ftp_class_map
hostname(config-cmap)# match access-list ftp
```

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**）。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。

照合ステートメントについては、「トラフィックの特定（レイヤ 3/4 クラス マップ）」（P.1-14）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　FTP 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。

ステップ 4　FTP インスペクションを設定します。

```
inspect ftp [strict [ftp_policy_map]]
```

それぞれの説明は次のとおりです。

- **strict** は、厳密な FTP を実装します。FTP インスペクション ポリシー マップを指定するには、厳密な FTP を使用する必要があります。
- *ftp_policy_map* は、オプションの FTP インスペクション ポリシー マップです。デフォルト以外のインスペクション処理が必要な場合にのみマップが必要です。FTP インスペクション ポリシー マップの作成の詳細については、「FTP インスペクション ポリシー マップの設定」（P.8-11）を参照してください。

例：

```
hostname(config-class)# no inspect ftp
hostname(config-class)# inspect ftp strict ftp-map
```

（注）　デフォルトのグローバル ポリシー（または使用中の任意のポリシー）を編集して、異なる FTP インスペクション ポリシー マップを使用する場合は、**no inspect ftp** コマンドで FTP インスペクションを除去した後、新しい FTP インスペクション ポリシー マップ名を指定して再度追加します。

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 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。

FTP インスペクション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。

- 取得またはアップロードされたファイルごとに監査レコード 303002 が生成されます。
- FTP コマンドが RETR または STOR であるかがチェックされ、取得コマンドおよび保存コマンドがログに記録されます。
- IP アドレスを提供するテーブルを検索してユーザ名が取得されます。

- ユーザ名、接続元の IP アドレス、接続先の IP アドレス、NAT アドレス、およびファイル操作がログに記録されます。
- メモリ不足によって動的なセカンダリ チャネルの準備に失敗した場合は、監査レコード 201005 が生成されます。

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

ここでは、HTTP インスペクション エンジンについて説明します。

HTTP インスペクションの概要

ヒント　アプリケーションおよび URL のフィルタリングを実行するサービス モジュールをインストールできます。これには、ASA CX や ASA FirePOWER などの HTTP インスペクションが含まれます。ASA 上で実行される HTTP インスペクションは、これらのモジュールと互換性がありません。HTTP インスペクション ポリシー マップを使用して ASA 上で手作業による設定を試みるより、専用のモジュールを使用してアプリケーション フィルタリングを設定する方がはるかに簡単であることに注意してください。

HTTP インスペクション エンジンを使用して、HTTP トラフィックに関係する特定の攻撃やその他の脅威から保護します。

HTTP アプリケーション インスペクションで HTTP のヘッダーと本文をスキャンし、さまざまなデータ チェックができます。これらのチェックで、HTTP 構築、コンテンツ タイプ、トンネル プロトコル、メッセージ プロトコルなどがセキュリティ アプライアンスを通過することを防止します。

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 インスペクション ポリシー マップを設定するときに使用できます。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。

HTTP アプリケーション インスペクションでトンネル アプリケーションと ASCII 以外の文字を含む HTTP 要求や応答をブロックして、悪意のあるコンテンツが Web サーバに到達することを防ぎます。HTTP 要求や応答ヘッダーのさまざまな要素のサイズ制限、URL のブロッキング、HTTP サーバ ヘッダー タイプのスプーフィングもサポートされています。

拡張 HTTP インスペクションは、すべての HTTP メッセージについて次の点を確認します。

- 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

HTTP インスペクションの設定

HTTP インスペクションはデフォルトではイネーブルになりません。ASA CX や ASA FirePOWER などの HTTP インスペクションおよびアプリケーション フィルタリングに専用のモジュールを使用していない場合、以下の方法を使用して、ASA に HTTP インスペクションを手動で設定できます。

ヒント　サービス モジュールと ASA の両方で HTTP インスペクションを設定しないでください。インスペクションの互換性はありません。

手順

ステップ 1　「HTTP インスペクション ポリシー マップの設定」（P.8-17）。

ステップ 2　「HTTP インスペクション サービス ポリシーの設定」（P.8-21）。

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。

はじめる前に

一部のトラフィック照合オプションでは、照合のために正規表現を使用します。これらのテクニックの 1 つを使用する場合は、最初に正規表現または正規表現のクラス マップを作成します。

手順

ステップ 1　（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照合をグループ化します。代わりに、ポリシー マップで **match** コマンドを直接指定できます。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 マップで指定するトラフィックに対しては、インスペクション ポリシー マップで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。

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 1 つの **match** ステートメントと一致したらクラス 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. 次のいずれかの **match** コマンドを使用して、アクションを実行するトラフィックを指定します。**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。

- **match** [**not**] **req-resp content-type mismatch**：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します。
- **match** [**not**] **request args regex** {*regex_name* | **class** *class_name*}：HTTP 要求メッセージの引数で見つかったテキストを、指定した正規表現または正規表現クラスと照合します。
- **match** [**not**] **request body** {**regex** {*regex_name* | **class** *class_name*} | **length gt** *bytes*}：HTTP 要求メッセージの本文で見つかったテキストを、指定した正規表現または正規表現クラスと照合します。または、要求の本文が指定した長さより長いメッセージを照合します。
- **match** [**not**] **request header** {*field* | **regex** *regex_name*} **regex** {*regex_name* | **class** *class_name*}：HTTP 要求メッセージ ヘッダーのフィールドの内容を、指定した正規表現または正規表現クラスと照合します。フィールド名を明示的に指定することも、フィールド名を正規表現と一致させることもできます。フィールド名は次のとおりです。accept、accept-charset、accept-encoding、accept-language、allow、authorization、cache-control、connection、content-encoding、content-language、content-length、content-location、content-md5、content-range、content-type、cookie、date、expect、expires、from、host、if-match、if-modified-since、if-none-match、if-range、if-unmodified-since、last-modified、max-forwards、pragma、proxy-authorization、range、referer、te、trailer、transfer-encoding、upgrade、user-agent、via、warning。
- **match** [**not**] **request header** {*field* | **regex** {*regex_name* | **class** *class_name*}} {**length gt** *bytes* | **count gt** *number*}：HTTP 要求メッセージ ヘッダーの指定したフィールドの長さ、またはヘッダーのフィールドの総数を照合します。フィールド名を明示的に指定することも、フィールド名を正規表現または正規表現クラスと一致させることもできます。フィールド名は、前の項目の一覧と同じです。
- **match** [**not**] **request header** {**length gt** *bytes* | **count gt** *number* | **non-ascii**}：HTTP 要求メッセージ ヘッダーの全体の長さ、ヘッダーのフィールドの総数、または ASCII 以外の文字を含むヘッダーを照合します。

- **match** [**not**] **request method** {*method* | **regex** {*regex_name* | **class** *class_name*}}：HTTP 要求のメソッドを照合します。メソッドを明示的に指定することも、メソッドを正規表現または正規表現クラスと一致させることもできます。メソッドは次のとおりです。bcopy、bdelete、bmove、bpropfind、bproppatch、connect、copy、delete、edit、get、getattribute、getattributenames、getproperties、head、index、lock、mkcol、mkdir、move、notify、options、poll、post、propfind、proppatch、put、revadd、revlabel、revlog、revnum、save、search、setattribute、startrev、stoprev、subscribe、trace、unedit、unlock、unsubscribe。
- **match** [**not**] **request uri** {**regex** {*regex_name* | **class** *class_name*} | **length gt** *bytes*}：HTTP 要求メッセージの URI で見つかったテキストを、指定した正規表現または正規表現クラスと照合します。または、要求の URI が指定した長さより長いメッセージを照合します。
- **match** [**not**] **response body** {**active-x** | **java-applet** | **regex** {*regex_name* | **class** *class_name*}}：HTTP 応答メッセージの本文で見つかったテキストを、指定した正規表現または正規表現クラスと照合します。または、Java アプレットおよび Active X オブジェクトをフィルタ処理のためにコメント化します。
- **match** [**not**] **response body length gt** *bytes*：本文が指定した長さより大きい HTTP 応答メッセージを照合します。
- **match** [**not**] **response header** {*field* | **regex** *regex_name*} **regex** {*regex_name* | **class** *class_name*}：HTTP 応答メッセージ ヘッダーのフィールドの内容を、指定した正規表現または正規表現クラスと照合します。フィールド名を明示的に指定することも、フィールド名を正規表現と一致させることもできます。フィールド名は次のとおりです。accept-ranges、age、allow、cache-control、connection、content-encoding、content-language、content-length、content-location、content-md5、content-range、content-type、date、etag、expires、last-modified、location、pragma、proxy-authenticate、retry-after、server、set-cookie、trailer、transfer-encoding、upgrade、vary、via、warning、www-authenticate。
- **match** [**not**] **response header** {*field* | **regex** {*regex_name* | **class** *class_name*}} {**length gt** *bytes* | **count gt** *number*}：HTTP 応答メッセージ ヘッダーの指定したフィールドの長さ、またはヘッダーのフィールドの総数を照合します。フィールド名を明示的に指定することも、フィールド名を正規表現または正規表現クラスと一致させることもできます。フィールド名は、前の項目の一覧と同じです。
- **match** [**not**] **response header** {**length gt** *bytes* | **count gt** *number* | **non-ascii**}：HTTP 応答メッセージ ヘッダーの全体の長さ、ヘッダーのフィールドの総数、または ASCII 以外の文字を含むヘッダーを照合します。
- **match** [**not**] **response status-line regex** {*regex_name* | **class** *class_name*}：HTTP 応答メッセージのステータス行で見つかったテキストを、指定した正規表現または正規表現クラスと照合します。

d. クラス マップ コンフィギュレーション モードを終了するには、「**exit**」と入力します。

ステップ 2　HTTP インスペクション ポリシー マップを作成します。

```
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 3　（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 4　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。

- HTTP クラス マップを作成した場合は、次のコマンドを入力してそれを指定します。

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

- HTTP クラス マップで説明されている **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-connection [log]| reset [log] | log}
```

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。

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、サーバまたはクライアントに 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。

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.2-4）を参照してください。

ステップ 5　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. 1 つまたは複数のパラメータを設定します。次のオプションを設定できます。オプションをディセーブルにするには、コマンドの **no** 形式を使用してください。

- **body-match-maximum** *number*：HTTP メッセージの本文照合時に検索する本文の最大文字数を設定します。デフォルトは 200 バイトです。大きな値を指定すると、パフォーマンスに大きな影響を与えます。
- **protocol-violation action {drop-connection [log] | reset [log] | log}**：HTTP メッセージの本文照合時に検索する本文の最大文字数を設定します。デフォルトは 200 バイトです。大きな値を指定すると、パフォーマンスに大きな影響を与えます。HTTP プロトコル違反を検査します。違反に対して実行するアクション（切断、リセット、ログ記録）、およびロギングをイネーブルまたはディセーブルにするかどうかも選択する必要があります。
- **spoof-server** *string*：サーバ ヘッダー フィールドの文字列を置き換えます。WebVPN ストリームは **spoof-server** コマンドの対象ではありません。

例

次に、「GET」メソッドまたは「PUT」メソッドで「www\.xyz.com/.*\.asp」または「www\.xyz[0-9][0-9]\.com」にアクセスしようとしている HTTP 接続を許可し、ロギングする HTTP インスペクション ポリシー マップを定義する例を示します。それ以外の URL/メソッドの組み合わせは、サイレントに許可されます。

```
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
```

HTTP インスペクション サービス ポリシーの設定

HTTP インスペクションはデフォルトのインスペクション ポリシーではイネーブルにされないため、このインスペクションが必要な場合はイネーブルにする必要があります。ただし、デフォルトの inspect クラスにはデフォルトの HTTP ポートが含まれているので、デフォルトのグローバル インスペクション ポリシーを編集するだけで HTTP インスペクションを追加できます。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。

手順

ステップ 1　必要な場合は、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 http_class_map
hostname(config-cmap)# match access-list http
```

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**）。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。

照合ステートメントについては、「トラフィックの特定（レイヤ 3/4 クラス マップ）」（P.1-14）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　HTTP 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。

ステップ 4　HTTP インスペクションを設定します。

inspect http [*http_policy_map*]

http_policy_map は、オプションの HTTP インスペクション ポリシー マップです。デフォルト以外のインスペクション処理が必要な場合にのみマップが必要です。HTTP インスペクション ポリシー マップの作成の詳細については、「HTTP インスペクション ポリシー マップの設定」（P.8-17）を参照してください。

例：

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

（注）　デフォルトのグローバル ポリシー（または使用中の任意のポリシー）を編集して、異なる HTTP インスペクション ポリシー マップを使用する場合は、**no inspect http** コマンドで HTTP インスペクションを除去した後、新しい HTTP インスペクション ポリシー マップ名を指定して再度追加します。

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 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。

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は、ACL で ICMP が ASA を通過するのを禁止することを推奨します。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。

ただし、ASA インターフェイスに送られる 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。

ICMP インスペクションをイネーブルにする方法については、「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。

ICMP エラー インスペクション

ICMP エラー インスペクションをイネーブルにすると、ASA は NAT の設定に基づいて、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。

ICMP エラー インスペクションをイネーブルにする方法については、「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。

インスタント メッセージ インスペクション

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 のネットワーク使用を制御し、機密情報の漏洩、ワームの送信、および企業ネットワークへのその他の脅威を停止できます。

IM インスペクションはデフォルトではイネーブルになりません。IM インスペクションが必要な場合は設定する必要があります。

手順

ステップ 1　「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）。

ステップ 2　「IM インスペクション サービス ポリシーの設定」（P.8-27）。

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。

はじめる前に

一部のトラフィック照合オプションでは、照合のために正規表現を使用します。これらのテクニックの 1 つを使用する場合は、最初に正規表現または正規表現のクラス マップを作成します。

手順

ステップ 1　（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照合をグループ化します。代わりに、ポリシー マップで **match** コマンドを直接指定できます。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 マップで指定するトラフィックに対しては、インスペクション ポリシー マップで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。

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 1 つの **match** ステートメントと一致したらクラス 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. 次のいずれかの **match** コマンドを使用して、アクションを実行するトラフィックを指定します。**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。

- **match** [**not**] **protocol {im-yahoo | im-msn}**：特定の IM プロトコル（Yahoo または MSN）を照合します。
- **match** [**not**] **service {chat | file-transfer | webcam | voice-chat | conference | games}**：特定の IM サービスを照合します。
- **match** [**not**] **login-name regex** {*regex_name* | **class** *class_name*}：IM メッセージの送信元クライアント ログイン名を、指定された正規表現または正規表現クラスに対して照合します。
- **match** [**not**] **peer-login-name regex** {*regex_name* | **class** *class_name*}：IM メッセージの宛先ピア ログイン名を、指定された正規表現または正規表現クラスに対して照合します。
- **match** [**not**] **ip-address** *ip_address mask*}：IM メッセージの送信元 IP アドレスとマスクを照合します。
- **match** [**not**] **peer-ip-address** *ip_address mask*}：IM メッセージの宛先 IP アドレスとマスクを照合します。
- **match** [**not**] **version regex** {*regex_name* | **class** *class_name*}：IM メッセージのバージョンを、指定された正規表現または正規表現クラスに対して照合します。
- **match** [**not**] **filename regex** {*regex_name* | **class** *class_name*}:IM メッセージのファイル名を、指定された正規表現または正規表現クラスに対して照合します。この照合は MSN IM プロトコルに対してはサポートされません。

d. クラス マップ コンフィギュレーション モードを終了するには、「**exit**」と入力します。

ステップ 2　IM インスペクション ポリシー マップを作成します。

```
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 3　（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 4　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。

- IM クラス マップを作成した場合は、次のコマンドを入力してそれを指定します。

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

- IM クラス マップで説明されている **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-connection [log]| reset [log] | log}
```

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。

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、サーバまたはクライアントに 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。

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.2-4）を参照してください。

例

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
```

IM インスペクション サービス ポリシーの設定

IM インスペクションはデフォルトのインスペクション ポリシーではイネーブルにされないため、このインスペクションが必要な場合はイネーブルにする必要があります。ただし、デフォルトの inspect クラスにはデフォルトの IM ポートが含まれているので、デフォルトのグローバル インスペクション ポリシーを編集するだけで IM インスペクションを追加できます。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。

手順

ステップ 1　必要な場合は、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 im_class_map
hostname(config-cmap)# match access-list im
```

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**）。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。

照合ステートメントについては、「トラフィックの特定（レイヤ 3/4 クラス マップ）」（P.1-14）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　IM 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。

ステップ 4　IM インスペクションを設定します。

inspect im [*im_policy_map*]

im_policy_map は、オプションの IM インスペクション ポリシー マップです。デフォルト以外のインスペクション処理が必要な場合にのみマップが必要です。IM インスペクション ポリシー マップの作成の詳細については、「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）を参照してください。

例：

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

（注）　デフォルトのグローバル ポリシー（または使用中の任意のポリシー）を編集して、異なる IM インスペクション ポリシー マップを使用する場合は、**no inspect im** コマンドで IM インスペクションを除去した後、新しい IM インスペクション ポリシー マップ名を指定して再度追加します。

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 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。

IP オプション インスペクション

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 オプションをクリアしたうえでのパケットの転送許可を ASA に指示します。

ここでは、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。

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 オプションおよび関連する RFC の参照のリストについては、IANA のページ（http://www.iana.org/assignments/ip-parameters/ip-parameters.xhtml）を参照してください。

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 オプションをクリアしたうえでのパケットの転送許可を ASA に指示します。

オプションをクリアしたときの結果

IP オプション インスペクション ポリシー マップを設定する場合、各オプション タイプを許可またはクリアするかどうかを指定できます。オプション タイプを指定しないと、そのオプションを含むパケットはドロップされます。

オプションを単に許可すると、そのオプションを含むパケットは未変更で渡されます。

IP ヘッダーからオプションをクリアするように指定すると、IP ヘッダーは次のように変更されます。

- オプションがヘッダーから除去されます。
- 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。
- パケット内のインターネット ヘッダー長（IHL）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。

インスペクションでサポートされる IP オプション

IP オプション インスペクションでは、パケット内の次の IP オプションをチェックできます。IP ヘッダーにこれら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。

- 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。すべてのオプションのビット数が 32 ビットの倍数でない場合、NOP オプションは、オプションを 32 ビット境界上に揃えるために、「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、パケットの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。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 オプション インスペクションのデフォルト

IP オプション インスペクションは、_default_ip_options_map インスペクション ポリシー マップを使用して、デフォルトで有効になります。

- Router Alert オプションは許可されます。
- その他のオプションを含むパケットはドロップされます。これには、サポートされていないオプションを含むパケットが含まれます。

ポリシー マップのコンフィギュレーションは次のとおりです。

```
policy-map type inspect ip-options _default_ip_options_map
 description Default IP-OPTIONS policy-map
 parameters
  router-alert action allow
```

IP オプション インスペクションの設定

IP オプション 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。デフォルト マップで許可されているもの以外の追加オプションが必要な場合にのみ、設定する必要があります。

手順

ステップ 1　「IP オプション インスペクション ポリシー マップの設定」（P.8-31）。

ステップ 2　「IP オプション インスペクション サービス ポリシーの設定」（P.8-31）。

IP オプション インスペクション ポリシー マップの設定

デフォルト以外の IP オプション インスペクションを実行する場合は、IP オプション インスペクション ポリシー マップを作成して、サポートされる各オプション タイプの処理方法を指定します。

手順

ステップ 1　IP オプション インスペクション ポリシー マップを設定します。

```
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.　1 つまたは複数のパラメータを設定します。次のオプションを設定できます。オプションをディセーブルにするには、コマンドの **no** 形式を使用してください。いずれの場合も、**allow** アクションはそのオプションを含むパケットを変更なしで許可し、**clear** アクションはパケットを許可しますがヘッダーからそのオプションを除去します。マップで指定されていないオプションを含むパケットはすべてドロップされます。オプションの詳細については、「インスペクションでサポートされる IP オプション」（P.8-30）を参照してください。

- **eool action** {**allow** | **clear**}：End of Options List オプションを許可またはクリアします。
- **nop action** {**allow** | **clear**}：No Operation オプションを許可またはクリアします。
- **router-alert action** {**allow** | **clear**}：Router Alert（RTRALT）オプションを許可またはクリアします。

IP オプション インスペクション サービス ポリシーの設定

ASA のデフォルトのコンフィギュレーションには、すべてのインターフェイスにグローバルに適用される IP オプション インスペクションが含まれます。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。

手順

ステップ 1　必要な場合は、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 ip_options_class_map
hostname(config-cmap)# match access-list ipoptions
```

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**）。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。

照合ステートメントについては、「トラフィックの特定（レイヤ 3/4 クラス マップ）」（P.1-14）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　IP オプション 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。

ステップ 4　IP オプション インスペクションを設定します。

inspect ip-options [*ip_options_policy_map*]

ip_options_policy_map は、オプションの IP オプション インスペクション ポリシー マップです。デフォルト以外のインスペクション処理が必要な場合にのみマップが必要です。IP オプション インスペクション ポリシー マップの作成の詳細については、「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）を参照してください。

例：

```
hostname(config-class)# no inspect ip-options
hostname(config-class)# inspect ip-options ip-options-map
```

（注）　デフォルトのグローバル ポリシー（または使用中の任意のポリシー）を編集して、異なる IP オプション インスペクション ポリシー マップを使用する場合は、**no inspect ip-options** コマンドで IP オプション インスペクションを除去した後、新しい IP オプション インスペクション ポリシー マップ名を指定して再度追加します。

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 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。

IP オプション インスペクションのモニタリング

以下の方法を使用して IP オプション インスペクションの結果をモニタリングできます。

- インスペクションによってパケットがドロップされるたびに、syslog 106012 が発行されます。メッセージではドロップの原因になったオプションが示されます。
- **show service-policy inspect ip-options** コマンドを使用して、各オプションの統計情報を表示します。

IPsec パススルー インスペクション

ここでは、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。

- 「IPsec パススルー インスペクションの概要」（P.8-33）
- 「IPsec パススルー インスペクションの設定」（P.8-34）

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検査できます。このインスペクションは、冗長な ACL 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。

ESP または AH トラフィックの制限を指定するには、IPsec パススルーのポリシー マップを設定します。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 インスペクションの設定

IPsec パススルー インスペクションはデフォルトではイネーブルになりません。IPsec パススルー インスペクションが必要な場合は設定する必要があります。

手順

ステップ 1　「IPsec パススルー インスペクション ポリシー マップの設定」（P.8-34）。

ステップ 2　「IPsec パススルー インスペクション サービス ポリシーの設定」（P.8-35）。

IPsec パススルー インスペクション ポリシー マップの設定

IPsec パススルー マップでは、IPsec パススルー アプリケーション インスペクションのデフォルト設定値を変更できます。IPsec パススルー マップを使用すると、アクセス リストを使用しなくても、特定のフローを許可できます。

コンフィギュレーションに含まれるデフォルト マップ _default_ipsec_passthru_map では、ESP 接続に対するクライアントごとの最大数は制限なしに設定され、ESP アイドル タイムアウトは 10 分に設定されます。異なる値が必要な場合、または AH 値を設定する必要がある場合にのみ、インスペクション ポリシー マップを設定する必要があります。

手順

ステップ 1　IPsec パススルー インスペクション ポリシー マップを作成します。

```
hostname(config)# policy-map type inspect ipsec-pass-thru 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.　1 つまたは複数のパラメータを設定します。次のオプションを設定できます。オプションをディセーブルにするには、コマンドの **no** 形式を使用してください。

- **esp per-client-max** *number* **timeout** *time* : ESP トンネルを許可し、クライアントごとに許可される最大接続数およびアイドル タイムアウト（hh:mm:ss の形式）を設定します。接続の数を無制限に設定するには、値を 0 に指定します。
- **ah per-client-max** *number* **timeout** *time* : AH トンネルを許可します。パラメータの意味は esp コマンドと同じです。

例

次に、ACL 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
```

IPsec パススルー インスペクション サービス ポリシーの設定

IPsec パススルー インスペクションはデフォルトのインスペクション ポリシーではイネーブルにされないため、このインスペクションが必要な場合はイネーブルにする必要があります。ただし、デフォルトの inspect クラスにはデフォルトの IPsec ポートが含まれているので、デフォルトのグローバル インスペクション ポリシーを編集するだけで IPsec インスペクションを追加できます。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。

手順

ステップ 1　必要な場合は、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 ipsec_class_map
hostname(config-cmap)# match access-list ipsec
```

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**）。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。

照合ステートメントについては、「トラフィックの特定（レイヤ 3/4 クラス マップ）」（P.1-14）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　IPsec パススルー オプション 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。

ステップ 4　IPsec パススルー インスペクションを設定します。

inspect ipsec-pass-thru [*ipsec_policy_map*]

ipsec_policy_map は、オプションの IPsec パススルー インスペクション ポリシー マップです。デフォルト以外のインスペクション処理が必要な場合にのみマップが必要です。インスペクション ポリシー マップの作成の詳細については、「IPsec パススルー インスペクション ポリシー マップの設定」（P.8-34）を参照してください。

例：

```
hostname(config-class)# no inspect ipsec-pass-thru
hostname(config-class)# inspect ipsec-pass-thru ipsec-map
```

（注）　デフォルトのグローバル ポリシー（または使用中の任意のポリシー）を編集して、異なる IPsec パススルー インスペクション ポリシー マップを使用する場合は、**no inspect ipsec-pass-thru** コマンドで IPsec パススルー インスペクションを除去した後、新しい IPsec パススルー インスペクション ポリシー マップ名を指定して再度追加します。

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 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。

IPv6 インスペクション

IPv6 インスペクションを使用すると、拡張ヘッダーに基づいて IPv6 トラフィックを選択的にログに記録したりドロップしたりできます。さらに、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も確認できます。

- 「IPv6 インスペクションのデフォルト」（P.8-37）
- 「IPv6 インスペクションの設定」（P.8-37）

IPv6 インスペクションのデフォルト

IPv6 インスペクションをイネーブルにし、インスペクション ポリシー マップを指定しないと、デフォルトの IPv6 インスペクション ポリシー マップが使用され、次のアクションが実行されます。

- 既知の IPv6 拡張ヘッダーのみを許可します。準拠しないパケットはドロップされ、ログに記録されます。
- RFC 2460 仕様で定義されている IPv6 拡張ヘッダーの順序を適用します。準拠しないパケットはドロップされ、ログに記録されます。
- ルーティング タイプ ヘッダーを含むパケットをドロップします。

ポリシー マップのコンフィギュレーションは次のとおりです。

```
policy-map type inspect ipv6 _default_ipv6_map
 description Default IPV6 policy-map
 parameters
  verify-header type
  verify-header order
 match header routing-type range 0 255
  drop log
```

IPv6 インスペクションの設定

IPv6 インスペクションはデフォルトではイネーブルになりません。IPv6 インスペクションが必要な場合は設定する必要があります。

手順

ステップ 1　「IPv6 インスペクション ポリシー マップの設定」（P.8-37）。

ステップ 2　「IPv6 インスペクション サービス ポリシーの設定」（P.8-39）。

IPv6 インスペクション ポリシー マップの設定

ドロップまたはロギングする拡張ヘッダーを指定するには、またはパケットの検証をディセーブルにするには、サービス ポリシーで使用される IPv6 インスペクション ポリシー マップを作成します。

手順

ステップ 1　IPv6 インスペクション ポリシー マップを作成します。

```
hostname(config)# policy-map type inspect ipv6 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）説明をポリシー マップに追加します。

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

ステップ 3　（任意）IPv6 メッセージのヘッダーに基づいてトラフィックをドロップまたはロギングします。

a. IPv6 ヘッダーに基づいてトラフィックを識別します。

```
hostname(config-pmap)# match header type
```

type は次のいずれかです。

- **ah**：IPv6 認証拡張ヘッダーと一致します。
- **count gt** *number*：IPv6 拡張ヘッダーの最大数を指定します（0 ～ 255）。
- **destination-option**：IPv6 の宛先オプション拡張ヘッダーと一致します。
- **esp**：IPv6 のカプセル化セキュリティ ペイロード（ESP）拡張ヘッダーと一致します。
- **fragment**：IPv6 のフラグメント拡張ヘッダーと一致します。
- **hop-by-hop**：IPv6 のホップバイホップ拡張ヘッダーと一致します。
- **routing-address count gt** *number*：IPv6 ルーティング ヘッダー タイプ 0 アドレスの最大数を設定します（0 ～ 255）。
- **routing-type** {**eq** | **range**} *number*：IPv6 ルーティング ヘッダー タイプと一致します（0 ～ 255）。範囲を指定するには、値をスペースで区切ります（例：**30 40**）。

b. 一致したパケットに対して実行するアクションを指定します。パケットをドロップし、必要に応じてログに記録できます。または、ログへの記録だけを行うこともできます。アクションを入力しない場合、パケットがログに記録されます。

```
hostname(config-pmap)# {drop [log] | log}
```

c. ドロップまたはロギングするすべてのヘッダーを識別するまで、プロセスを繰り返します。

ステップ 4　インスペクション エンジンに影響するパラメータを設定します。

a. パラメータ コンフィギュレーション モードを開始します。

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

b. 1 つまたは複数のパラメータを設定します。次のオプションを設定できます。オプションをディセーブルにするには、コマンドの **no** 形式を使用してください。

- **verify-header type**：既知の IPv6 拡張ヘッダーだけを許可します。
- **verify-header order**：RFC 2460 で定義されている IPv6 拡張ヘッダーの順序を適用します。

例

次の例では、ホップバイホップ、宛先オプション、ルーティング アドレス、およびルーティング タイプ 0 の各ヘッダーを含むすべての IPv6 パケットをドロップし、ログに記録するインスペクション ポリシー マップを作成します。また、ヘッダーの順序とタイプを適用します。

```
policy-map type inspect ipv6 ipv6-pm
 parameters
  verify-header type
  verify-header order
 match header hop-by-hop
  drop log
 match header destination-option
  drop log
 match header routing-address count gt 0
  drop log
```

```
 match header routing-type eq 0
  drop log

policy-map global_policy
 class class-default
  inspect ipv6 ipv6-pm
!
service-policy global_policy global
```

IPv6 インスペクション サービス ポリシーの設定

IPv6 インスペクションはデフォルトのインスペクション ポリシーではイネーブルにされないため、このインスペクションが必要な場合はイネーブルにする必要があります。デフォルトのグローバル インスペクション ポリシーを編集して IPv6 インスペクションを追加できます。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。

手順

ステップ 1　必要な場合は、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 ipv6_class_map
hostname(config-cmap)# match access-list ipv6
```

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**）。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。

照合ステートメントについては、「トラフィックの特定（レイヤ 3/4 クラス マップ）」（P.1-14）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　IPv6 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。

ステップ 4　IPv6 インスペクションを設定します。

inspect ipv6 [*ipv6_policy_map*]

ipv6_policy_map は、オプションの IPv6 インスペクション ポリシー マップです。デフォルト以外のインスペクション処理が必要な場合にのみマップが必要です。インスペクション ポリシー マップの作成の詳細については、「IPv6 インスペクション ポリシー マップの設定」（P.8-37）を参照してください。

例：

```
hostname(config-class)# no inspect ipv6
hostname(config-class)# inspect ipv6 ipv6-map
```

（注）　デフォルトのグローバル ポリシー（または使用中の任意のポリシー）を編集して、異なる IPv6 インスペクション ポリシー マップを使用する場合は、**no inspect ipv6** コマンドで IPv6 インスペクションを除去した後、新しい IPv6 インスペクション ポリシー マップ名を指定して再度追加します。

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 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。

NetBIOS インスペクション

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 ネーム サービス（NBNS）パケット内の IP アドレスを変換します。必要に応じて、NetBIOS プロトコル違反をドロップまたはログに記録するポリシー マップを作成できます。

手順

ステップ 1　「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）。

ステップ 2　「NetBIOS インスペクション サービス ポリシーの設定」（P.8-41）。

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

プロトコル違反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。

手順

ステップ 1　NetBIOS インスペクション ポリシー マップを作成します。

```
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 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　パラメータ コンフィギュレーション モードを開始します。

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

ステップ 4　NetBIOS プロトコル違反に対して実行するアクションを指定します。

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

drop アクションはパケットをドロップします。**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。

例

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

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

NetBIOS インスペクション サービス ポリシーの設定

NetBIOS アプリケーション インスペクションでは、NetBIOS ネーム サービス パケットおよび NetBIOS データグラム サービス パケットに埋め込まれている IP アドレスで NAT を実行します。また、プロトコル準拠チェックを行って、さまざまなフィールドの数や長さの整合性を確認します。

ASA のデフォルトの設定には、すべてのインターフェイスにグローバルに適用されるデフォルト ポートの NetBIOS インスペクションが含まれます。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。

手順

ステップ 1　必要な場合は、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 netbios_class_map
hostname(config-cmap)# match access-list netbios
```

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**）。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。

照合ステートメントについては、「トラフィックの特定（レイヤ 3/4 クラス マップ）」（P.1-14）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　NetBIOS 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。

ステップ 4　NetBIOS インスペクションを設定します。

```
inspect netbios [netbios_policy_map]
```

netbios_policy_map は、オプションの NetBIOS インスペクション ポリシー マップです。デフォルト以外のインスペクション処理が必要な場合にのみマップが必要です。NetBIOS インスペクション ポリシー マップの作成の詳細については、「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）を参照してください。

例：

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

（注）　デフォルトのグローバル ポリシー（または使用中の任意のポリシー）を編集して、異なる NetBIOS インスペクション ポリシー マップを使用する場合は、**no inspect skinny** コマンドで NetBIOS インスペクションを除去した後、新しい NetBIOS インスペクション ポリシー マップ名を指定して再度追加します。

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 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。

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。

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。

PPTP インスペクションをイネーブルにする方法については、「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。

SMTP および拡張 SMTP インスペクション

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。

ここでは、ESMTP インスペクション エンジンについて説明します。

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。復帰（CR）、および改行（LF）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、次の異常なシグニチャがないかどうか、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。縦棒（|）は削除され（ブランクに変更されます）、「<」および「>」はメール アドレスを定義する場合にのみ許可されます（「>」より前に「<」がある必要があります）。
- 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るため、TCP チェックサムの再計算または調整が必要になります。
- TCP ストリーム編集
- コマンド パイプライン

ESMTP インスペクションのデフォルト

ESMTP インスペクションは、_default_esmtp_map インスペクション ポリシー マップを使用して、デフォルトで有効になります。

- サーバ バナーはマスクされます。
- 暗号化されたトラフィックが検査されます。
- 送信側と受信側のアドレスの特殊文字は認識されず、アクションは実行されません。
- コマンド行の長さが 512 より大きい接続は、ドロップされてログに記録されます。
- 受信者が 100 より多い接続は、ドロップされてログに記録されます。
- 本文の長さが 998 バイトより大きいメッセージはログに記録されます。
- ヘッダー行の長さが 998 より大きい接続は、ドロップされてログに記録されます。
- MIME ファイル名が 255 文字より長いメッセージは、ドロップされてログに記録されます。
- 「others」に一致する EHLO 応答パラメータはマスクされます。

ポリシー マップのコンフィギュレーションは次のとおりです。

```
policy-map type inspect esmtp _default_esmtp_map
 description Default ESMTP policy-map
 parameters
  mask-banner
  no mail-relay
  no special-character
  no allow-tls
 match cmd line length gt 512
  drop-connection log
 match cmd RCPT count gt 100
  drop-connection log
 match body line length gt 998
  log
 match header line length gt 998
  drop-connection log
 match sender-address length gt 320
  drop-connection log
```

```
match MIME filename length gt 255
 drop-connection log
match ehlo-reply-parameter others
 mask
```

ESMTP インスペクションの設定

ESMTP 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。デフォルト インスペクション マップとは異なるプロセスが必要な場合にのみ、設定する必要があります。

手順

ステップ 1　「ESMTP インスペクション ポリシー マップの設定」（P.8-46）。

ステップ 2　「ESMTP インスペクション サービス ポリシーの設定」（P.8-48）。

ESMTP 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、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。

はじめる前に

一部のトラフィック照合オプションでは、照合のために正規表現を使用します。これらのテクニックの 1 つを使用する場合は、最初に正規表現または正規表現のクラス マップを作成します。

手順

ステップ 1　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 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　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 **match** コマンドを使用して、アクションを実行するトラフィックを指定します。**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。

- **match** [**not**] **body** {**length** | **line length**} **gt** *bytes*：ESMTP 本文メッセージの長さまたは行の長さが指定したバイト数より大きいメッセージと一致します。
- **match** [**not**] **cmd verb** *verb1* [*verb2...*]：メッセージ内のコマンド動詞と一致します。次のコマンドの 1 つまたは複数を指定できます。auth、data、ehlo、etrn、helo、help、mail、noop、quit、rcpt、rset、saml、soml、vrfy。

- **match** [**not**] **cmd line length gt** *bytes*：コマンド動詞の行の長さが指定したバイト数より大きいメッセージと一致します。
- **match** [**not**] **cmd rcpt count gt** *count*：受信者の数が指定した値より大きいメッセージと一致します。
- **match** [**not**] **ehlo-reply-parameter** *parameter* [*parameter2...*]：ESMTP EHLO 応答パラメータと一致します。次のパラメータの 1 つまたは複数を指定できます。8bitmime、auth、binaryname、checkpoint、dsn、etrn、others、pipelining、size、vrfy。
- **match** [**not**] **header** {**length** | **line length**} **gt** *bytes*：ESMTP ヘッダーの長さまたは行の長さが指定したバイト数より大きいメッセージと一致します。
- **match** [**not**] **header to-fields count gt** *count*：ヘッダーの To フィールドの数が指定した値より大きいメッセージと一致します。
- **match** [**not**] **invalid-recipients count gt** *number*：無効な受信者の数が指定した値より大きいメッセージと一致します。
- **match** [**not**] **mime filetype regex** {*regex_name* | **class** *class_name*}：MIME またはメディア ファイル タイプを、指定した正規表現または正規表現クラスと照合します。
- **match** [**not**] **mime filename length gt** *bytes*：ファイル名が指定したバイト数より大きいメッセージと一致します。
- **match** [**not**] **mime encoding** *type* [*type2...*]：MIME エンコーディング タイプと一致します。次のタイプの 1 つまたは複数を指定できます。7bit、8bit、base64、binary、others、quoted-printable。
- **match** [**not**] **sender-address regex** {*regex_name* | **class** *class_name*}：送信者の電子メール アドレスを、指定した正規表現または正規表現クラスと照合します。
- **match** [**not**] **sender-address length gt** *bytes*：送信者のアドレスが指定したバイト数より大きいメッセージと一致します。

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-connection [log]| mask [log] | reset [log] | log |
rate-limit message_rate}
```

各 **match** 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-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。このアクションは、**ehlo-reply-parameter** および **cmd verb** に対してのみ使用できます。
- **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。このオプションは、**cmd verb** のみで使用できます。唯一のアクションとして使用することも、**mask** アクションと組み合わせて使用することもできます。

ポリシー マップでは、複数の **match** コマンドを指定できます。**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.2-4）を参照してください。

ステップ 4　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. 1 つまたは複数のパラメータを設定します。次のオプションを設定できます。オプションをディセーブルにするには、コマンドの **no** 形式を使用してください。

- **mail-relay** *domain-name* **action** {**drop-connection** [**log**] | **log**}：メール中継のドメイン名を指定します。接続をドロップし、必要に応じてログに記録できます。または、ログへの記録だけを行うこともできます。
- **mask-banner**：ESMTP サーバからのバナーをマスクします。
- **special-character action** {**drop-connection** [**log**] | **log**}：電子メールの送信者または受信者アドレスに特殊文字パイプ（|）、バック クォート、NUL が含まれるメッセージに対して実行するアクションを指定します。接続をドロップし、必要に応じてログに記録できます。または、ログへの記録だけを行うこともできます。
- **allow-tls** [**action log**]：インスペクションなしで ESMTP over TLS（暗号化された接続）を許可するかどうか。必要に応じて、暗号化された接続をログに記録できます。

例

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 "Regular 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
```

ESMTP インスペクション サービス ポリシーの設定

ASA のデフォルトのコンフィギュレーションには、すべてのインターフェイスにグローバルに適用される ESMTP インスペクションが含まれます。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。

手順

ステップ 1　必要な場合は、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 esmtp_class_map
hostname(config-cmap)# match access-list esmtp
```

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**）。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。

照合ステートメントについては、「トラフィックの特定（レイヤ 3/4 クラス マップ）」（P.1-14）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　IP オプション 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。

ステップ 4　ESMTP インスペクションを設定します。

inspect esmtp [*esmtp_policy_map*]

esmtp_policy_map は、オプションの ESMTP インスペクション ポリシー マップです。デフォルト以外のインスペクション処理が必要な場合にのみマップが必要です。ESMTP インスペクション ポリシー マップの作成の詳細については、「ESMTP インスペクション サービス ポリシーの設定」（P.8-48）を参照してください。

例：

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

（注）　デフォルトのグローバル ポリシー（または使用中の任意のポリシー）を編集して、異なるインスペクション ポリシー マップを使用する場合は、**no inspect esmtp** コマンドで ESMTP インスペクションを除去した後、新しいインスペクション ポリシー マップ名を指定して再度追加します。

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 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。

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 読み取り要求（RRQ）、書き込み要求（WRQ）、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。

TFTP インスペクションをイネーブルにする方法については、「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。