CHAPTER 2

アプリケーション インスペクションの特別なアクション（インスペクション ポリシー マップ）

モジュラ ポリシー フレームワークでは、多くのアプリケーション インスペクションで実行される特別なアクションを設定できます。レイヤ 3/4 ポリシー マップでインスペクション エンジンをイネーブルにする場合は、*インスペクション ポリシー マップ*で定義されるアクションを必要に応じてイネーブルにすることもできます。インスペクション ポリシー マップが、インスペクション アクションを定義したレイヤ 3/4 クラス マップ内のトラフィックと一致すると、トラフィックのそのサブセットが指定したとおりに動作します（たとえば、ドロップやレート制限など）。

インスペクション ポリシー マップに関する情報

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

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

- トラフィック照合コマンド：インスペクション ポリシー マップで直接トラフィック照合コマンドを定義して、アプリケーションのトラフィックを、URL 文字列などのアプリケーションに固有の基準と照合できます。一致した場合にはアクションをイネーブルにします。
  - 一部のトラフィック照合コマンドでは、正規表現を指定してパケット内部のテキストを照合できます。ポリシー マップを設定する前に、正規表現クラス マップ内で、正規表現を単独またはグループで作成およびテストしておいてください。

- インスペクション クラス マップ：インスペクション クラス マップには、複数のトラフィック照合コマンドが含まれます。その後、ポリシー マップでクラス マップを指定し、クラス マップのアクションを全体としてイネーブルにします。クラス マップを作成することと、インスペクション ポリシー マップ内で直接トラフィック照合を定義することの違いは、より複雑な一致基準を作成できる点と、クラス マップを再使用できる点です。ただし、異なる照合基準に対して異なるアクションを設定することはできません。**注：**すべてのアプリケーションがインスペクション クラス マップをサポートするわけではありません。
- パラメータ：パラメータは、インスペクション エンジンの動作に影響します。

ガイドラインと制限事項

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

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

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

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

- 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。

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

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

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

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

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

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

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

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

デフォルトのインスペクション ポリシー マップ

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 コマンドを参照してください。

```
policy-map type inspect dns preset_dns_map
 parameters
    message-length maximum client auto
    message-length maximum 512
    dns-guard
    protocol-enforcement
    nat-rewrite
```

（注） デフォルトのインスペクション ポリシー マップは、**_default_esmtp_map** など、ほかにもあります。たとえば、**inspect esmtp** はポリシー マップ「_default_esmtp_map」を暗黙的に使用します。すべてのデフォルト ポリシー マップは、**show running-config all policy-map** コマンドを使用して表示できます。

インスペクション ポリシー マップのアクションの定義

レイヤ 3/4 ポリシー マップでインスペクション エンジンをイネーブルにする場合は、インスペクション ポリシー マップで定義されるアクションを必要に応じてイネーブルにすることもできます。

手順の詳細

	コマンド	目的
ステップ 1	（任意） インスペクション クラス マップを作成します。	「インスペクション クラス マップ内のトラフィックの特定」（P.2-6）を参照してください。 または、ポリシー マップ内でトラフィックを直接特定できます。
ステップ 2	（任意） 正規表現を作成します。	正規表現をサポートするポリシー マップ タイプについては、一般的な操作のコンフィギュレーション ガイドを参照してください。

	コマンド	目的
ステップ 3	**policy-map type inspect** *application policy_map_name* **例：** `hostname(config)# policy-map type inspect http http_policy`	インスペクション ポリシー マップを作成します。インスペクション ポリシー マップをサポートするアプリケーションのリストについては、「アプリケーション レイヤ プロトコル インスペクションの設定」（P.7-11）を参照してください。 *policy_map_name* 引数は、最大 40 文字のポリシー マップ名です。すべてのタイプのポリシー マップで同じ名前スペースが使用されるため、別のタイプのポリシー マップですでに使用されている名前は再度使用できません。CLI はポリシー マップ コンフィギュレーション モードに入ります。
ステップ 4	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。	

	class *class_map_name*
	
	例：
	`hostname(config-pmap)# class http_traffic`
	`hostname(config-pmap-c)#`
		「インスペクション クラス マップ内のトラフィックの特定」（P.2-6）で作成したインスペクション クラス マップを指定します。
		
		すべてのアプリケーションがインスペクション クラス マップをサポートするわけではありません。

	インスペクションの章でアプリケーションごとに説明されている **match** コマンドの 1 つを使用して、ポリシー マップで直接トラフィックを指定します。
	
	例：
	`hostname(config-pmap)# match req-resp content-type mismatch`
	`hostname(config-pmap-c)#`
		match not コマンドを使用すると、**match not** コマンドの基準に一致するすべてのトラフィックにアクションは適用されません。
		
		正規表現をサポートするポリシー マップ タイプについては、一般的な操作のコンフィギュレーション ガイドを参照してください。

ステップ 5	*action*
	
	例：
	`hostname(config-pmap-c)#`
	`drop-connection log`
		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。アクションは、インスペクションおよび一致タイプよって異なります。一般的なアクションは、**drop**、**log**、および **drop-connection** です。各一致で使用できるアクションについては、該当するインスペクションの章を参照してください。

ステップ 6	**parameters**
	
	例：
	`hostname(config-pmap)# parameters`
	`hostname(config-pmap-p)#`
		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。CLI はパラメータ コンフィギュレーション モードに移行します。各アプリケーションで設定可能なパラメータについては、該当するインスペクションの章を参照してください。

例

次の例では、HTTP インスペクション ポリシー マップとその関連クラス マップを示します。このポリシー マップは、サービス ポリシーがイネーブルにするレイヤ 3/4 ポリシー マップによってアクティブになります。

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

hostname(config-cmap)# class-map type inspect http match-all http-traffic
hostname(config-cmap)# match req-resp content-type mismatch
hostname(config-cmap)# match request body length gt 1000
hostname(config-cmap)# match not request uri regex class URLs

hostname(config-cmap)# policy-map type inspect http http-map1
hostname(config-pmap)# class http-traffic
hostname(config-pmap-c)# drop-connection log
hostname(config-pmap-c)# match req-resp content-type mismatch
hostname(config-pmap-c)# reset log
hostname(config-pmap-c)# parameters
hostname(config-pmap-p)# protocol-violation action log

hostname(config-pmap-p)# policy-map test
hostname(config-pmap)# class test（レイヤ 3/4 クラス マップは表示されません）
hostname(config-pmap-c)# inspect http http-map1

hostname(config-pmap-c)# service-policy test interface outside
```

インスペクション クラス マップ内のトラフィックの特定

このタイプのクラス マップを使用して、アプリケーション固有の基準と照合できます。たとえば DNS トラフィックの場合は、DNS クエリー内のドメイン名と照合可能です。

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（match-all クラス マップ）。あるいはクラス マップで、照合リストのいずれかを照合できます（match-any クラス マップ）。クラス マップを作成することと、インスペクション ポリシー マップ内で直接トラフィック照合を定義することの違いは、クラス マップを使用して複数の match コマンドをグループ化できる点と、クラス マップを再使用できる点です。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、接続のドロップ、リセット、またはロギングなどのアクションを指定できます。タイプの異なるトラフィックで異なるアクションを実行する場合は、ポリシー マップで直接トラフィックを指定してください。

制約事項

すべてのアプリケーションがインスペクション クラス マップをサポートするわけではありません。サポートされるアプリケーションのリストについては、**class-map type inspect** の CLI ヘルプを参照してください。

手順の詳細

	コマンド	目的
ステップ 1	（任意） 正規表現を作成します。	一般的な操作のコンフィギュレーション ガイドを参照してください。

ステップ 2	**class-map type inspect** *application* [**match-all** | **match-any**] *class_map_name*
	
	例：
	`hostname(config)# class-map type inspect http http_traffic`
	`hostname(config-cmap)#`
		インスペクション クラス マップを作成します。*application* は検査するアプリケーションです。サポートされるアプリケーションのリストについては、CLI ヘルプまたは第 7 章「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」を参照してください。
		
		class_map_name 引数は、最大 40 文字のクラス マップ名です。
		
		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。

ステップ 3	（任意） **description** *string* **例：** `hostname(config-cmap)# description All UDP traffic`	クラス マップに説明を追加します。
ステップ 4	アプリケーションで使用可能な 1 つ以上の **match** コマンドを入力して、クラスに含めるトラフィックを定義します。	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。 各アプリケーションで使用可能な **match** コマンドについては、該当するインスペクションの章を参照してください。

例

次の例では、すべての基準に一致する必要がある HTTP クラス マップを作成します。

```
hostname(config-cmap)# class-map type inspect http match-all http-traffic
hostname(config-cmap)# match req-resp content-type mismatch
hostname(config-cmap)# match request body length gt 1000
hostname(config-cmap)# match not request uri regex class URLs
```

次の例では、基準のいずれかに一致する必要がある HTTP クラス マップを作成します。

```
hostname(config-cmap)# class-map type inspect http match-any monitor-http
hostname(config-cmap)# match request method get
hostname(config-cmap)# match request method put
hostname(config-cmap)# match request method post
```

次の作業

インスペクション ポリシーを使用するには、第 1 章「モジュラ ポリシー フレームワークを使用したサービス ポリシー」を参照してください。

インスペクション ポリシー マップの機能履歴

表 2-1 に、この機能のリリース履歴を示します。

表 2-1　　サービス ポリシーの機能履歴

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