

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を参照してください。

- DCERPC インスペクション （2 ページ）
- DNS インスペクション （5 ページ）
- FTP インスペクション （11 ページ）
- HTTP インスペクション （17 ページ）
- ICMP インスペクション （22 ページ）
- ICMP エラー インスペクション （22 ページ）
- ILS インスペクション （23 ページ）
- インスタント メッセージ インスペクション （24 ページ）
- IP オプション インスペクション （27 ページ）
- IPsec パススルー インスペクション （30 ページ）
- IPv6 インスペクション （32 ページ）
- NetBIOS インスペクション （35 ページ）
- PPTP インスペクション （36 ページ）
- RSH インスペクション （36 ページ）
- SMTP および拡張 SMTP インスペクション （37 ページ）
- SNMP インスペクション （42 ページ）
- SQL*Net インスペクション （43 ページ）
- Sun RPC インスペクション （43 ページ）
- TFTP インスペクション （45 ページ）
- XDMCP インスペクション （46 ページ）
- VXLAN インスペクション （46 ページ）
- 基本的なインターネット プロトコル インスペクションの履歴 （47 ページ）

DCERPC インスペクション

デフォルトのインスペクション ポリシーでは、DCERPC インスペクションがイネーブルにされていないため、この検査が必要な場合はイネーブルにします。デフォルトのグローバルインスペクション ポリシーを編集するだけで、DCERPC 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。

次の項では、DCERPC インスペクション エンジンについて説明します。

DCERPC の概要

DCERPC に基づく Microsoft リモート プロシージャ コール（MSRPC）は、Microsoft 分散クライアントおよびサーバー アプリケーションで広く使用されているプロトコルであり、ソフトウェア クライアントがサーバー上のプログラムをリモートで実行できるようにします。

通常、このプロトコルの接続では、クライアントが予約済みポート番号で接続を受け入れるエンドポイント マッパーというサーバーに、必要なサービスについてダイナミックに割り当てられるネットワーク情報を問い合わせます。次に、クライアントは、サービスを提供しているサーバーのインスタンスへのセカンダリ接続をセットアップします。セキュリティ アプライアンスは、適切なポート番号とネットワーク アドレスへのセカンダリ接続を許可し、必要に応じて NAT を適用します。

DCERPC インスペクション エンジンは、EPM とウェルノウン TCP ポート 135 上のクライアントとの間のネイティブ TCP 通信を検査します。クライアント用に EPM のマッピングとルックアップがサポートされています。クライアントとサーバーは、どのセキュリティ ゾーンにあってもかまいません。埋め込まれたサーバーの IP アドレスとポート番号は、EPM からの応答メッセージで受け取ります。クライアントが EPM から返されたサーバーのポートに対して複数の接続を試みる可能性があるので、ピンホールが複数使用でき、ユーザーがそのタイムアウトを設定できるようになっています。

DCE インスペクションは、次の汎用一意識別子（UUID）とメッセージをサポートします。

- エンドポイント マッパー（EPM）UUID。すべての EPM メッセージがサポートされます。
- ISystemMapper UUID（非 EPM）。サポートされるメッセージ タイプは次のとおりです。
  - RemoteCreateInstance opnum4
  - RemoteGetClassObject opnum3
- OxidResolver UUID（非EPM）。サポートされるメッセージは次のとおりです。
  - ServerAlive2 opnum5
- IP アドレスまたはポート情報を含まない任意のメッセージ（これらのメッセージでは検査の必要がないため）。

DCERPC インスペクション ポリシー マップの設定

DCERPC インスペクションの追加のパラメータを指定するには、DCERPC 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、DCERPC インスペクションをイネーブルにすると適用できます。

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。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、クラスマップを再使用できる点です。

手順

ステップ 1 （任意）DCERPC インスペクション クラス マップを作成します。

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) クラス マップを作成します。**class-map type inspect dcerpc** [**match-all** | **match-any**] *class_map_name*

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 はクラス マップ コンフィギュレーション モードに移行します。

b) 次の **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**] **uuid** *type*：DCERPC メッセージの汎用一意識別子（UUID）を照合します。*type* は次のいずれかです。
  - **ms-rpc-epm**：Microsoft RPC EPM メッセージを照合します。
  - **ms-rpc-isystemactivator**：ISystemMapper メッセージを照合します。
  - **ms-rpc-oxidresolver**：OxidResolver メッセージを照合します。

c) 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 DCERPC インスペクション ポリシー マップを作成します。**policy-map type inspect dcerpc** *policy_map_name*

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）説明をポリシー マップに追加します。**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。

- DCERPC 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。**class** *class_map_name*
- DCERPC 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) 次のコマンドのいずれかを入力して、一致する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。

- **reset** [**log**]：パケットをドロップし、接続を閉じてサーバーまたはクライアントに TCP リセットを送信します。
- **log**：システム ログ メッセージを送信します。このオプションは単独で使用するか、または他のアクション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。

例：

```
hostname(config)# policy-map type inspect dcerpc dcerpc-map
hostname(config-pmap)# match uuid ms-rpc-epm
hostname(config-pmap-c)# 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。

- **timeout pinhole** *hh:mm:ss*：DCERPC ピンホールのタイムアウトを設定し、2 分のグローバル システム ピンホール タイムアウトを上書きします。タイムアウトは 00:00 01 ～ 119:00:00 まで指定できます。
- **endpoint-mapper** [**epm-service-only**] [**lookup-operation** [**timeout** *hh:mm:ss*]]：エンドポイント マッパー トラフィックのオプションを設定します。**epm-service-only** キーワードを指定すると、バインド中にエンドポイント マッパー サービスを実行し、このサービスのトラフィックだけが処理されるようにします。**lookup-operation** キーワードを指定すると、エンドポイント マッパー サービスのルックアップ操作をイネーブルにします。ルックアップ操作で生成されたピンホールのタイムアウトを設定できます。

ルックアップ操作にタイムアウトが設定されていない場合は、timeout pinhole コマンドで指定した値かデフォルトの値が使用されます。

例

次の例は、DCERPCインスペクション ポリシーマップを定義し、DCERPC のピンホールのタイムアウトを設定する方法を示しています。

```
hostname(config)# policy-map type inspect dcerpc dcerpc_map
hostname(config-pmap)# timeout pinhole 0:10:00

hostname(config)# class-map dcerpc
hostname(config-cmap)# match port tcp eq 135

hostname(config)# policy-map global-policy
hostname(config-pmap)# class dcerpc
hostname(config-pmap-c)# inspect dcerpc dcerpc-map

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

次のタスク

マップを使用するためのインスペクション ポリシーを設定できるようになりました。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 を参照してください。

DNS インスペクション

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。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。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。
- DNS over TCP インスペクションは無効です。
- 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ーして DNS 応答の ID が DNS クエリの 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
```



（注） 毎秒多数のパケットが検査される場合、Cisco ASA で CPU 使用率が上昇する可能性があります。処理されるパケットの数が増えると（特にデバイスがスループットまたは接続キャパシティの最大値に近づいた場合）、CPU 使用率も上昇します。これは、Cisco ASA が毎秒多くの操作を実行する必要があり、CPU リソースの消費が増えるために発生します。

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。

始める前に

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との照合をグループ化します。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) クラス マップを作成します。**class-map type inspect dns** [**match-all** | **match-any**] *class_map_name*

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します。**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 サーバー）、**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 マップを作成します。**policy-map type inspect dns** *policy_map_name*

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）説明をポリシー マップに追加します。**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。**class** *class_map_name*
- 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) 次のコマンドのいずれかを入力して、一致する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。

- **drop** [**log**]：一致するすべてのパケットをドロップします。
- **drop-connection** [**log**]：パケットをドロップし、接続を閉じます。
- **mask** [**log**]：パケットの一致する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**：システム ログ メッセージを送信します。このオプションは単独で使用するか、または他のアクションのいずれかと一緒に使用できます。

- **enforce-tsig** [**drop**] [**log**]：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、複数のトラフィック クラスの処理方法を参照してください。

例：

```
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。

- **dnscrypt**：DNSCrypt を有効にしてデバイスと Cisco Umbrella 間の接続を暗号化します。DNSCrypt を有効にすると、Umbrella リゾルバとのキー交換スレッドが開始されます。キー交換スレッドは、1 時間ごとにリゾルバとのハンドシェイクを実行し、新しい秘密鍵でデバイスを更新します。DNSCrypt では UDP/443 を使用するため、そのポートが DNS インスペクションに使用するクラス マップに含まれていることを確認する必要があります。デフォルトのインスペクション クラスには DNS インスペクションに UDP/443 がすでに含まれています。
- **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ーして DNS 応答の ID が DNS クエリの 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

またはサーバー 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。
- **tcp-inspection**：DNS over TCP トラフィックのインスペクションを有効にします。DNS/TCP ポート 53 トラフィックが、DNS インスペクションを適用するクラスの一部であることを確認します。インスペクションのデフォルト クラスには、TCP/53 が含まれています。
- **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。
- **umbrella** ［**tag** *umbrella_policy*］［**fail-open**］：Cisco Umbrella をイネーブルにし、必要に応じてデバイスに適用する Cisco Umbrella のポリシー名（**tag**）を指定します。ポリシーを指定しない場合は、デフォルトの ACL が適用されます。詳細については、Cisco Umbrellaを参照してください。

  Umbrella DNS サーバーが使用できない場合に DNS 解決を動作させるには、**fail-open** キーワードを追加します。フェール オープンの状態で Cisco Umbrella DNS サーバーが使用できない場合は、このポリシー マップで Umbrella 自体がディセーブルになり、DNS 要求をシステム上に設定された他の DNS サーバー（存在する場合）に移動できるようになります。Umbrella DNS サーバーが再度使用可能になると、ポリシーマップはそれらの使用を再開します。このオプションが含まれていない場合、DNS 要求は到達不能の Umbrella リゾルバへ移動し続けるので、応答は取得されません。

例：

```
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。

```
regex domain_example “example\.com”
regex domain_foo “foo\.com”

! define the domain names that the server serves
```

```
class-map type 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

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

次のタスク

マップを使用するためのインスペクション ポリシーを設定できるようになりました。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 を参照してください。

FTP インスペクション

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。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

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。

- FTP データ転送のために動的なセカンダリ データ接続チャネルを準備します。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。
- FTP コマンド/応答シーケンスを追跡します。
- 監査証跡を生成します。

- 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。
- Audit record 201005 is generated if the secondary dynamic channel preparation failed due to memory shortage.

- 埋め込み IP アドレスを変換します。

（注） FTPインスペクションをディセーブルにすると、発信ユーザーはパッシブ モードでしか接続を開始できなくなり、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。

厳密な FTP インスペクションでは、次の動作が強制されます。

- 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 厳密な FTP 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。さらに、インスペクションをFTPポートのみに適用する必要があります（通常の FTP ポートは TCP/21 です）。非 FTP トラフィックに厳密な FTP インスペクションを適用すると、（特にHTTPトラフィックで）予期しないトラフィック損失が発生する可能性があります。

厳密な FTP インスペクションでは、各 FTP コマンドと応答のシーケンスを追跡し、次の異常なアクティビティがないかをチェックします。

- 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 コマンドがサーバー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、常にサーバーから送信されます。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、ユーザーが「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 サーバーの応答を連続した X で置き換えて、サーバーのシステム タイプが FTP クライアントに知られないようにします。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 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、ユーザーの値に基づいて 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ー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 サーバーがそのシステム 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。

始める前に

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との照合をグループ化します。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) クラス マップを作成します。**class-map type inspect ftp** [**match-all** | **match-any**] *class_map_name*

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します。**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**：サーバーのファイルを削除します。

- **GET**：サーバー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**：サーバーにディレクトリを作成します。
- **PUT**：ファイルをサーバーに送信します。
- **RMD**：サーバー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**：サーバー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 サーバー名を、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 ユーザー名を、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。**policy-map type inspect ftp** *policy_map_name*

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）説明をポリシー マップに追加します。**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。**class** *class_map_name*
- 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。

- **reset** [**log**]：パケットをドロップし、接続を閉じてサーバー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、複数のトラフィック クラスの処理方法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。

- **mask-banner**：FTP サーバー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

ユーザー名とパスワードを送信する前に、すべての FTP ユーザー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
```

次のタスク

マップを使用するためのインスペクション ポリシーを設定できるようになりました。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 を参照してください。

HTTP インスペクション

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。

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

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 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。

始める前に

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との照合をグループ化します。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) クラス マップを作成します。**class-map type inspect http** [**match-all** | **match-any**] *class_map_name*

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します。**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。**policy-map type inspect http** *policy_map_name*

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）説明をポリシー マップに追加します。**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。**class** *class_map_name*
- 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) 次のコマンドのいずれかを入力して、一致する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。

- **drop-connection** [**log**]：パケットをドロップし、接続を閉じます。

- **reset** [**log**]：パケットをドロップし、接続を閉じてサーバーまたはクライアントに TCP リセットを送信します。
- **log**：システム ログ メッセージを送信します。このオプションは単独で使用するか、または他のアクション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、複数のトラフィック クラスの処理方法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。

- **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。デフォルト値は 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 プロトコル違反について確認します。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*：サーバーのヘッダー フィールドを文字列に置き換えます。WebVPN ストリームは spoof-server コマンドの対象になりません。

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
```

次のタスク

マップを使用するためのインスペクション ポリシーを設定できるようになりました。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 を参照してください。

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信される ICMP トラフィックは、ICMP インスペクションをイネーブルにした場合でも検査されません。したがって、ASA がバックアップ デフォルト ルートを介して到達できる送信元からエコー要求が送信された場合など、特定の状況下では、インターフェイスへの ping（エコー要求）が失敗する可能性があります。

（注） NAT は、ICMP インスペクションを無効にしても、パケットを変換するときに ICMP インスペクションを使用します。

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。

（注） NAT が ICMP パケットで使用される可能性がある場合は、常に ICMP エラー インスペクションを有効にする必要があります。NAT は、ICMP インスペクションを無効にしていても ICMP パケットに対して ICMP インスペクションを自動的に実行するため、マッピングされた宛先アドレスを送信元アドレスとして使用すると、スキャナーがネットワークを検査しているように見える可能性があります。たとえば、ICMP エラーインスペクションも有効になっていない場合、ICMP タイム超過応答に埋め込まれたエコー要求パケットの宛先が変換されると、タイム超過要求の外部ヘッダーでは、変換された宛先が送信元アドレスとして使用されます。ICMP エラーインスペクションを有効にすると、タイム超過になった送信元アドレスに正しい値が設定されます。

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を参照してください。

ILS インスペクション

Internet Locator Service（ILS）インスペクション エンジンは、LDAP を使用してディレクトリ情報を ILS サーバーと交換する Microsoft NetMeeting、SiteServer、および Active Directory の各製品に対して NAT をサポートします。LDAP データベースには IP アドレスだけが保存されるため、ILS インスペクションで PAT は使用できません。

LDAP サーバーが外部にある場合、内部ピアが外部 LDAP サーバーに登録された状態でローカルに通信できるように、検索応答に対して NAT を使用することを検討してください。NAT を使用する必要がなければ、パフォーマンスを向上させるためにインスペクション エンジンをオフにすることを推奨します。

ILS サーバーが ASA 境界の内部にある場合は、さらに設定が必要なことがあります。この場合、外部クライアントが指定されたポート（通常は TCP 389）の LDAP サーバーにアクセスするためのホールが必要となります。

（注） ILS トラフィック（H225 コール シグナリング）はセカンダリ UDP チャネルだけで発生するため、TCP 接続は TCP 非アクティブ間隔の後に切断されます。デフォルトでは、この間隔は 60 分です。この値は、TCP timeout コマンドを使用して調整できます。ASDM では、これは [Configuration] > [Firewall] > [Advanced] > [Global Timeouts] ペインにあります。

ILS インスペクションには、次の制限事項があります。

- 照会要求や応答はサポートされません。
- 複数のディレクトリのユーザーは統合されません。
- 複数のディレクトリに複数の ID を持っている単一のユーザーは NAT には認識されません。

ILS 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。

IM インスペクションを実装する場合は、メッセージがパラメータに違反した場合のアクションを指定する IM インスペクション ポリシー マップを設定することもできます。次の手順では、IM インスペクション ポリシー マップについて説明します。

始める前に

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との照合をグループ化します。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) クラスマップを作成します。**class-map type inspect im** [**match-all** | **match-any**] *class_map_name*

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します。**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。**policy-map type inspect im** *policy_map_name*

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）説明をポリシー マップに追加します。**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。**class** *class_map_name*
- 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。

- **drop-connection** [**log**]：パケットをドロップし、接続を閉じます。
- **reset** [**log**]：パケットをドロップし、接続を閉じてサーバーまたはクライアントに TCP リセットを送信します。
- **log**：システム ログ メッセージを送信します。このオプションは単独で使用するか、または他のアクション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、複数のトラフィック クラスの処理方法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
```

次のタスク

マップを使用するためのインスペクション ポリシーを設定できるようになりました。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 を参照してください。

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、パケット ヘッダーの [IP Options] フィールドのコンテンツに基づいてどの IP パケットを許可するかについて制御できます。望ましくないオプションがあるパケットをドロップしたり、オプションをクリア（してパケットを許可）したり、変更なしでパケットを許可したりできます。

IP オプションで提供される制御機能は、一部の状況では必須ですが、ほとんどの一般的な状況では不要です。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 オプションのインスペクションはデフォルトで有効になっていますが、RSVP トラフィックに対してのみとなっています。デフォルトのマップが許可しているもの以外に追加のオプションを許可するか、またはデフォルト以外のインスペクション トラフィック クラス マップを使用することによって他のタイプのトラフィックに適用する場合にのみ、これを設定する必要があります。

（注） IP オプション インスペクションは、フラグメント化されたパケットでは動作しません。たとえば、オプションはフラグメントからクリアされません。

次の項では、IP オプション インスペクショ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、RSVP トラフィックのデフォルトのみで有効になります。

- Router Alert オプションは許可されます。

  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。

- 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。

  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。

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 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、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 マップを作成します。**policy-map type inspect ip-options** *policy_map_name*

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。**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 許可するオプションを特定します。

次のオプションを検査できます。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。

マップからオプションを削除するには、このコマンドの **no** 形式を使用します。パケットに他の許可されているオプションまたはクリアされたオプションが含まれている場合でも、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。

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。

- **default action** {**allow** | **clear**}：マップに明示的に含まれていないオプションに対するデフォルト アクションを設定します。許可またはクリアのデフォルト アクションを設定しないと、許可されていないオプションを含むパケットはドロップされます。
- **basic-security action** {**allow** | **clear**}：Security（SEC）オプションを許可またはクリアします。
- **commercial-security action** {**allow** | **clear** }：Commercial Security（CIPSO）オプションを許可またはクリアします。
- **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。
- **exp-flow-control action** {**allow** | **clear**}：Experimental Flow Control（FINN）オプションを許可またはクリアします。
- **exp-measurement action** {**allow** | **clear**}：Experimental Measurement（ZSU）オプションを許可またはクリアします。
- **extended-security action** {**allow** | **clear**}：Extended Security（E-SEC）オプションを許可またはクリアします。
- **imi-traffic-descriptor action** {**allow** | **clear**}：IMI Traffic Descriptor（IMITD）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。
- **quick-start action** {**allow** | **clear**}：Quick-Start（QS）オプションを許可またはクリアします。
- **record-route action** {**allow** | **clear**}：Record Route（RR）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。
- **timestamp action** {**allow** | **clear**}：Time Stamp（TS）オプションを許可またはクリアします。

- {**0-255**} **action** {**allow** | **clear**}：オプション タイプ番号によって識別されるオプションを許可またはクリアします。番号は全オプション タイプのオクテット（コピー、クラス、およびオプション番号）で、オクテットのオプションの番号部分だけではありません。これらのオプション タイプは、実際のオプションに表示されない可能性があります。非標準オプションは、インターネット プロトコル RFC 791、http://tools.ietf.org/html/rfc791 で定義された予測されるタイプ/長さ/値の形式である必要があります。

次のタスク

マップを使用するためのインスペクション ポリシーを設定できるようになりました。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 を参照してください。

IPsec パススルー インスペクション

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。

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

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 ユーザーまたはサーバーなど）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 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 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。**policy-map type inspect ipsec-pass-thru** *policy_map_name*

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。**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。

- **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
```

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。

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 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。**policy-map type inspect ipv6** *policy_map_name*

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。**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。**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。

- **drop** [**log**]：一致するすべてのパケットをドロップします。
- **log**：システム ログ メッセージを送信します。このオプションは単独で使用するか、または他のアクションのいずれかと一緒に使用できます。

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
```

次のタスク

マップを使用するためのインスペクション ポリシーを設定できるようになりました。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 を参照してください。

NetBIOS インスペクション

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（NBNS）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。

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。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。次の手順で、NetBIOS インスペクション ポリシー マップを設定する方法について説明します。

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。**policy-map type inspect netbios** *policy_map_name*

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。**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。**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)# no inspect netbios
hostname(config-pmap-c)# inspect netbios netbios_map
```

次のタスク

マップを使用するためのインスペクション ポリシーを設定できるようになりました。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 を参照してください。

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。接続および xlate は、以降のセカンダリ 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を参照してください。

RSH インスペクション

RSH 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。RSH プロトコルは、TCP ポート 514 で RSH クライアントから RSH サーバーへの TCP 接続を使用します。クライアントとサーバーは、クライアントが STDERR 出力ストリームを受信する TCP ポート番号をネゴシエートします。RSH インスペクションは、必要に応じて、ネゴシエートされたポート番号の NAT をサポートします。

RSH インスペクションのイネーブル化の詳細については、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 インスペクションでは、スパム、フィッシング、不正形式メッセージ攻撃、バッファオーバーフロー/アンダーフロー攻撃などの攻撃を検出します。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、送受信者およびメール中継のブロックも行います。

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 マップとは異なる処理が必要な場合にのみ、設定する必要があります。

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 ESMTP インスペクションの概要

拡張 SMTP（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、モニター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。

ESMTP アプリケーション インスペクションは、ユーザーが使用できるコマンドとサーバーが返送するメッセージを制御し、その数を減らします。ESMTP 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。サポートされるコマンドは次のとおりです。
  - 拡張 SMTP：AUTH、EHLO、ETRN、HELP、SAML、SEND、SOML、STARTTLS、および VRFY。
  - SMTP（RFC 821）：DATA、HELO、MAIL、NOOP、QUIT、RCPT、RSET。
- SMTP コマンド応答シーケンスをモニターします。
- 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。

ESMTP 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ーします。

- 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 サーバーによる不意の移行

- 未知またはサポート対象外のコマンドに対し、インスペクションエンジンは、パケット内のすべての文字を X に変更し、それらは内部サーバーによって拒否されます。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、破棄されます。

  サポート対象外の ESMTP コマンドは ATRN、ONEX、VERB、CHUNKING で、プライベート拡張子です。

- TCP ストリーム編集
- コマンド パイプライン

（注） ESMTP 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。

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。

- サーバー バナーはマスクされます。ESMTP インスペクション エンジンは、文字「2」、「0」、「0」を除くサーバーの 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。
- 暗号化接続が可能ですが、検査されません。
- 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
 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 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。

始める前に

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 マップを作成します。**policy-map type inspect esmtp** *policy_map_name*

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。**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を照合します。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を照合します。
- **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) 次のコマンドのいずれかを入力して、一致する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。

- **drop-connection** [**log**]：パケットをドロップし、接続を閉じます。
- **mask** [**log**]：パケットの一致する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** [**log**]：パケットをドロップし、接続を閉じてサーバーまたはクライアントに TCP リセットを送信します。
- **log**：システム ログ メッセージを送信します。このオプションは単独で使用するか、または他のアクションのいずれかと一緒に使用できます。
- **rate-limit** *message_rate*：1 秒あたりのパケット内の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、複数のトラフィック クラスの処理方法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。

- **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 サーバー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。デフォルトでは、インスペクションのない TLS セッションを許可します。**no allow-tls** を指定すると、システムはセッション接続から STARTTLS インジケータを削除し、強制的にプレーンテキスト接続を行います。

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
```

次のタスク

マップを使用するためのインスペクション ポリシーを設定できるようになりました。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 を参照してください。

SNMP インスペクション

SNMP アプリケーション インスペクションは、デバイスへのトラフィックとデバイス経由のトラフィックの両方に適用されます。このインスペクションは、ユーザーが特定の SNMP ホストに制限される SNMP v3 を設定する場合に必要です。インスペクションなしの場合、定義された v3 ユーザーは任意の許可されたホストからデバイスをポーリングできます。SNMP インスペクションはデフォルトポートではデフォルトで有効になっているため、デフォルト以外のポートを使用する場合にのみ設定する必要があります。デフォルトポートは UDP/161、162 であり（すべてのデバイスタイプ）、FXOS は UDP/161 でリッスンするため、FXOS も実行するデバイスでは UDP/4161 です。

デフォルトでは、SNMP インスペクションはポーリングを構成されたバージョンに制限します。

（注） FXOS も実行しているデバイスで SNMP を設定する場合、SNMP インスペクションは必須であり、無効にすると再度有効になります。SNMP インスペクションは、ポート UDP/4161 を含むトラフィック クラスマップで有効になります。

必要に応じて、SNMP トラフィックを特定のバージョンの SNMP に制限することもできます。以前のバージョンの SNMP は安全性が低いため、セキュリティ ポリシーを使用して特定の SNMP バージョンを拒否する必要が生じる場合もあります。システムは、SNMP バージョン 1、2、2c、または 3 を拒否できます。許可するバージョンは、以下に説明するように、SNMP マップを作成して制御します。バージョンを制御する必要がない場合は、マップなしで SNMP インスペクションを有効にします。

手順

SNMP マップを作成します。

snmp-map *map_name* コマンドを使ってマップを作成して SNMP マップ 設定モードに入り、次に **deny version** *version* コマンドで拒否するバージョンを識別します。バージョンは 1、 2、 2c、 3 があります。

例：

次の例では、SNMP バージョン 1 および 2 を拒否しています。

```
hostname(config)# snmp-map sample_map
hostname(config-snmp-map)# deny version 1
hostname(config-snmp-map)# deny version 2
```

次のタスク

マップを使用するためのインスペクション ポリシーを設定できるようになりました。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 を参照してください。

SQL*Net インスペクション

SQL*Net 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。インスペクション エンジンは、SQL*Net バージョン 1 および 2 をサポートしていますが、形式は Transparent Network Substrate（TNS）のみです。インスペクションでは、表形式データ ストリーム（TDS）形式をサポートしていません。SQL*Net メッセージは、埋め込まれたアドレスとポートについてスキャンされ、必要に応じて NAT の書き換えが適用されます。

SQL*Net のデフォルトのポート割り当ては 1521 です。これは、Oracle が SQL*Net 用に使用している値ですが、構造化照会言語（SQL）の IANA ポート割り当てとは一致しません。アプリケーションが別のポートを使用する場合は、そのポートを含むトラフィック クラスに SQL*Net インスペクションを適用します。

次の場合は、SQL*Net インスペクションを無効にします。

- SQL 制御 TCP ポート 1521 と同じポートで SQL データ転送が行われます。SQL*Net インスペクションがイネーブルになっていると、セキュリティ アプライアンスはプロキシとして機能し、クライアントのウィンドウ サイズを 65000 から約 16000 に減らすため、データ転送の問題が発生します。
- 高レートの SQL 通信のインスペクションにより、CPU 使用率の許容できないスパイクが発生します。

SQL*Net インスペクションを無効にした後、 **clear conn port 1521** コマンドを使用して、インスペクションなしで接続を再構築できるようにします。

SQL*Net 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を参照してください。

Sun RPC インスペクション

この項では、Sun RPC 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。

Sun RPC インスペクションの概要

Sun RPC プロトコル インスペクションはデフォルトではイネーブルです。Sun RPC サーバー テーブルを管理するだけで、ファイアウォールの通過を許可されているサービスを識別できます。ただし、NFS のピンホール化は、サーバー テーブルの設定がなくても各サーバーで実行されます。

Sun RPC は、NFS および NIS で使用されます。Sun RPC サービスはどのポート上でも実行できます。サーバー上の Sun RPC サービスにアクセスしようとするクライアントは、そのサービスが実行されているポートを知る必要があります。そのためには、予約済みポート 111 でポート マッパー プロセス（通常は rpcbind）に照会します。

クライアントがサービスの Sun RPC プログラム番号を送信すると、ポート マッパー プロセスはサービスのポート番号を応答します。クライアントは、ポート マッパー プロセスによって特定されたポートを指定して、Sun RPC クエリーをサーバーに送信します。サーバーが応答すると、ASA はこのパケットを代行受信し、そのポートで TCP と UDP の両方の初期接続を開きます。

Sun RPC ペイロード情報の NAT または PAT はサポートされていません。

Sun RPC サービスの管理

Sun RPC サービス テーブルを使用して、確立された Sun RPC セッションに基づいて Sun RPC トラフィックを制御します。

手順

ステップ 1 Sun RPC サービス プロパティを設定します。

sunrpc-server *interface_name ip_address mask* **service** *service_type* **protocol** {**tcp** | **udp**} *port*[*-port*] **timeout** *hh:mm:ss*

それぞれの説明は次のとおりです。

- *interface_name*：サーバーへのトラフィックが伝送されるインターフェイス。
- *ip_address mask*：Sun RPC サーバーのアドレス。
- **service** *service_type*：特定のサービス タイプとそのサービスに使用するポート番号の間のマッピングである、サーバー上のサービス タイプ。サービス タイプ（100003 など）を判定するには、Sun RPC サーバー マシンの UNIX または Linux コマンドラインで、sunrpcinfo コマンドを使用します。
- **protocol** {**tcp** | **udp**}：サービスが TCP と UDP のどちらを使用するかを示します。
- *port*[*-port*]：サービスによって使用されるポートまたはポートの範囲。ポート範囲を指定するには、範囲の開始ポート番号と終了ポート番号をハイフンで区切ります（111-113 など）。

- **timeout** *hh:mm:ss*：Sun RPC インスペクションによって接続のために開かれたピンホールのアイドル タイムアウト。

例：

たとえば、IP アドレスが 192.168.100.2 の Sun RPC サーバーに対して 30 分のタイムアウトを作成するには、次のコマンドを入力します。この例では、Sun RPC サーバーは TCP ポート 111 を使用する内部インターフェイスにあります。

```
hostname(config)# sunrpc-server inside 192.168.100.2 255.255.255.255
service 100003 protocol tcp 111 timeout 00:30:00
```

ステップ 2 （オプション）これらのサービス用に作成されたピンホールをモニターします。

Sun RPC サービスで開かれているピンホールを表示するには、**show sunrpc-server active** コマンドを入力します。次に例を示します。

```
hostname# show sunrpc-server active
LOCAL FOREIGN SERVICE TIMEOUT
-----------------------------------------------
1 209.165.200.5/0 192.168.100.2/2049 100003 0:30:00
2 209.165.200.5/0 192.168.100.2/2049 100003 0:30:00
3 209.165.200.5/0 192.168.100.2/647 100005 0:30:00
4 209.165.200.5/0 192.168.100.2/650 100005 0:30:00
```

LOCAL カラムのエントリは、内部インターフェイスのクライアントまたはサーバーの IP アドレスを示します。FOREIGN カラムの値は、外部インターフェイスのクライアントまたはサーバーの IP アドレスを示します。

必要に応じ、次のコマンドを使用してこれらのサービスをクリアすることができます。 **clear sunrpc-server active**

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 サーバー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。

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、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ーの間のファイル転送を許可します。

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 サーバーだけがセカンダリ 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ー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。サーバー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を参照してください。

XDMCP インスペクション

XDMCP は、UDP ポート 177 を使用して X セッションをネゴシエートするプロトコルです。X セッションは確立時に TCP を使用します。

XWindows セッションを正常にネゴシエートして開始するために、ASA は、Xhosted コンピュータからの TCP 戻り接続を許可する必要があります。戻り接続を許可するには、TCP ポートを許可するアクセス ルールを使用できます。または、ASA で **established** コマンドを使用できます。XDMCP がディスプレイを送信するポートをネゴシエートすると、**established** コマンドが参照され、この戻り接続を許可すべきかどうかが確認されます。

XWindows セッション中、マネージャは予約済みポート 6000 | n 上でディスプレイ Xserver と通信します。次の端末設定を行うと、各ディスプレイは別々に Xserver と接続します。

```
setenv DISPLAY Xserver:n
```

n はディスプレイ番号です。

XDMCP が使用されている場合、ディスプレイは IP アドレスを使用してネゴシエートされます。IP アドレスは、ASA が必要に応じて NAT を行うことができます。XDCMP インスペクションでは、PAT はサポートされません。

XDMCP インスペクションのイネーブル化の詳細については、 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 を参照してください。

VXLAN インスペクション

Virtual Extensible Local Area Network （VXLAN）インスペクションは、ASA を通過する VXLAN のカプセル化されたトラフィックで機能します。VXLAN ヘッダーフォーマットが標準に準拠し、不正な形式のパケットをドロップすることを確認します。VXLAN インスペクションは、ASA が VXLAN トンネル エンドポイント（VTEP）または VXLAN ゲートウェイとして機能するトラフィックでは行われません。これは、それらのチェックが VXLAN パケットの通常の非カプセル化の一部として行われるためです。

VXLAN パケットは通常、ポート 4789 の UDP です。このポートは、default-inspection-traffic クラスの一部であるため、inspection_default サービス ポリシー ルールに VXLAN インスペクショ

ンを追加するだけです。または、それに対してポートまたは ACL マッチングを使用してクラスを作成することもできます。

基本的なインターネットプロトコルインスペクションの履歴

機能名	リリース	機能情報
DCERPC インスペクションで ISystemMapper UUID メッセージ RemoteGetClassObject opnum3 をサポート。	9.4(1)	ASA は、リリース 8.3 で EPM 以外の DCERPC メッセージのサポートを開始し、ISystemMapper UUID メッセージ RemoteCreateInstance opnum4 をサポートしています。この変更により、RemoteGetClassObject opnum3 メッセージまでサポートが拡張されます。 変更されたコマンドはありません。
VXLAN パケット インスペクション	9.4(1)	ASA は、標準形式に準拠するために VXLAN ヘッダーを検査できます。 **inspect vxlan** コマンドが導入されました。
ESMTP インスペクションの TLS セッションでのデフォルトの動作の変更。	9.4(1)	ESMTP インスペクションのデフォルトが、検査されない、TLS セッションを許可するように変更されました。ただし、このデフォルトは新しい、または再イメージングされたシステムに適用されます。**no allow-tls** を含むシステムをアップグレードする場合、このコマンドは変更されません。 デフォルトの動作の変更は、古いバージョンでも行われました：8.4（7.25）、8.5（1.23）、8.6（1.16）、8.7（1.15）、9.0（4.28）、9.1（6.1）、9.2（3.2）、9.3（1.2）、9.3（2.2）。
IP オプション インスペクションの改善	9.5(1)	IP オプション インスペクションは、すべての有効な IP オプションをサポートするようになりました。まだ定義されていないオプションを含む、標準または試行的なオプションを許可、クリア、またはドロップするようにインスペクションを調整できます。また、IP オプション インスペクション マップで明示的に定義されていないオプションのデフォルトの動作を設定できます。 **basic-security**、**commercial-security**、**default**、**exp-flow-control**、**exp-measure**、**extended-security**、**imi-traffic-description**、**quick-start**、**record-route**、**timestamp**、および {**0-255**}（IP オプション タイプ番号を示します）の各コマンドが追加されました。

機能名	リリース	機能情報
DCERPC インスペクションの改善および UUID フィルタリング	9.5(2)	DCERPC インスペクションは、OxidResolver ServerAlive2 opnum5 メッセージに対して NAT をサポートするようになりました。また、DCERPC メッセージの汎用一意識別子（UUID）でフィルタリングし、特定のメッセージ タイプをリセットするかログに記録できるようになりました。UUID フィルタリング用の新しい DCERPC インスペクション クラス マップがあります。 次のコマンドが導入されました。**match** [**not**] **uuid**。次のコマンドが変更されました。**class-map type inspect**。
DNS over TCP インスペクション。	9.6(2)	DNS over TCP トラフィック（TCP/53）を検査できるようになりました。 **tcp-inspection** コマンドが追加されました。
Cisco Umbrella サポート。	9.10(1)	Cisco Umbrella で定義されている エンタープライズ セキュリティ ポリシーをユーザー接続に適用できるように DNS 要求を Cisco Umbrella へリダイレクトするようにデバイスを設定できます。FQDN に基づいて接続を許可またはブロックできます。または、疑わしい FQDN の場合は Cisco Umbrella インテリジェント プロキシにユーザーをリダイレクトして URL フィルタリングを実行できます。Umbrella の設定は、DNS インスペクション ポリシーに含まれています。 **umbrella**（グローバルおよびポリシー マップ パラメータのコンフィギュレーション モード）、**token**、**public-key**、**timeout edns**、**dnscrypt**、**show service-policy inspect dns detail** の各コマンドが追加または変更されました。
Cisco Umbrella の強化	9.12(1)	Cisco Umbrella をバイパスする必要があるローカル ドメイン名を特定できるようになりました。これらのドメインの DNS 要求は、Umbrella を処理せず DNS サーバーに直接送信されます。また、DNS 要求の解決に使用する Umbrella サーバーも特定できるようになりました。さらに、Umbrella サーバーを使用できない場合は、DNS 要求がブロックされないように、Umbrella インスペクション ポリシーをフェール オープンに定義することができます。 **local-domain-bypass**、**resolver**、**umbrella fail-open** の各コマンドが追加または変更されました。

機能名	リリース	機能情報
新規インストールでは、デフォルトでXDMCPインスペクションが無効になっています。	9.15(1)	以前は、すべてのトラフィックに対してXDMCPインスペクションがデフォルトで有効になっていました。新しいシステムと再イメージ化されたシステムを含む新規インストールでは、XDMCPはデフォルトで無効になっています。このインスペクションが必要な場合は、有効にしてください。アップグレードでは、デフォルトのインスペクション設定を使用してXDMCPインスペクションを有効にしただけでも、XDMCPインスペクションの現在の設定は保持されます。

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。

翻訳について
このドキュメントは、米国シスコ発行ドキュメントの参考和訳です。リンク情報につきましては、日本語版掲載時点で、英語版にアップデートがあり、リンク先のページが移動/変更されている場合がありますことをご了承ください。あくまでも参考和訳となりますので、正式な内容については米国サイトのドキュメントを参照ください。