アプリケーション レイヤ プロトコル検査の適用

この章では、アプリケーション検査の使用方法および設定手順について説明します。この章で説明する内容は、次のとおりです。

アプリケーション インスペクション エンジンの概要

ここでは、アプリケーション インスペクション エンジンの機能について説明します。FWSM がステートフル アプリケーション検査用に使用するアダプティブ セキュリティ アルゴリズム（ASA）により、アプリケーションとサービスの安全な使用が保証されます。一部のアプリケーションでは FWSM による特別な処理を必要とし、特定のアプリケーション インスペクション エンジンはこの目的のために提供されています。特別なアプリケーション インスペクション エンジンを必要とするアプリケーションには、ユーザ データ パケットに IP アドレス情報を組み込んでいるものや、ダイナミックに割り当てられるポート上でセカンダリ チャネルを開始するものがあります。

アプリケーション インスペクション エンジンは Network Address Translation（NAT; ネットワーク アドレス変換）と連携して、組み込まれたアドレス情報の場所を特定します。この連携により、NAT は組み込まれたアドレスを変換し、変換によって影響を受けたチェックサムまたは他のフィールドを更新できます。

各アプリケーション インスペクション エンジンはセッションをモニタして、セカンダリ チャネルのポート番号を決定します。ほとんどのプロトコルは、パフォーマンスを向上するため、セカンダリの TCP ポートまたは UDP ポートをオープンします。well-known ポートでの初期セッションは、ダイナミックに割り当てられるポート番号のネゴシエーションに使用されます。アプリケーション インスペクション エンジンはこのセッションをモニタし、ダイナミック ポートの割り当てを確認し、特定のセッションの間、これらのポート上でのデータ交換を許可します。

次の内容について説明します。

- インスペクション エンジンの機能（p.20-3）
- サポート対象プロトコル（p.20-4）

インスペクション エンジンの機能

図 20-1 に示すように、FWSM は基本的な動作で、次のデータベースを使用します。

- アクセス リスト — 特定のネットワーク、ホスト、サービス（TCP/UDP ポート番号）に基づいた接続の認証および許可に使用されます。
- 検査 — スタティックで定義済みのアプリケーションレベルの検査機能がセットで含まれます。
- 接続（XLATE テーブルおよび CONN テーブル）— 確立された各接続のステートや情報を保持します。この情報は、確立されたセッション内におけるトラフィック転送を効率的に行うために、ASA とカットスルー プロキシによって使用されます。

図 20-1　ASA の基本的な動作

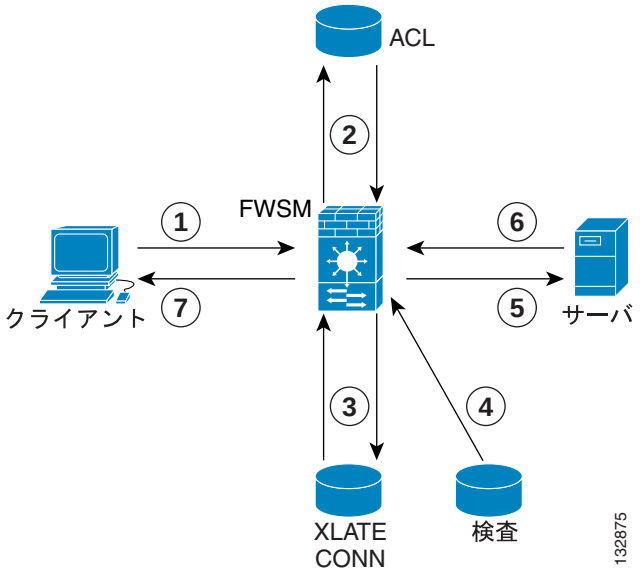


図 20-1 に、動作を実行順に説明します。

1. TCP SYN パケットが、新しい接続を確立するために FWSM に到着します。
2. FWSM は、アクセス リスト データベースを確認して、その接続を許可するかどうかを判別します。
3. FWSM は、接続データベース（XLATE テーブルおよび CONN テーブル）内に新しいエントリを作成します。
4. FWSM は、検査データベースを検証して、接続に対してアプリケーション レベルの検査が必要かどうかを判別します。
5. パケットに対してアプリケーション　インスペクション　エンジンの必要な処理が完了したら、FWSM は宛先システムにパケットを転送します。
6. 宛先システムは、初期要求に応答します。
7. FWSM は、応答パケットを受信し、接続データベースで接続を検索したあと、確立されたセッションに属しているパケットを転送します。

FWSM のデフォルトの設定には、アプリケーション検査エントリのセットが含まれています。このエントリは、サポートされているプロトコルを特定の TCP ポート番号または UDP ポート番号に対応付け、必要とされる特別な処理を指定します。

NAT、PAT、アプリケーション検査

NAT および Port Address Translation（PAT; ポートアドレス変換）を使用すると、次の方法でアプリケーション検査に影響を及ぼします。

- 一部のアプリケーション インスペクション エンジンは、変更できない固定ポートが割り当てられているので、NAT または PAT をサポートしません。アプリケーション エンジンがサポートする NAT および PAT の要約については、表 20-1 を参照してください。
- 検査されているトラフィックに PAT を設定すると、FWSM は、実ポート番号ではなく、変換されたポート番号でアプリケーション検査を実行します。

  変換されたポート番号を持つトラフィックに検査を適用するサービス ポリシーは、変換されたポート番号を使用して、トラフィックを識別するクラスマップを使用する必要があります。たとえば、PAT を実行してポート 2727 とポート 2427 をポート 1400 に変換する場合、well-known ポート 2427 とポート 2727 ではなく、ポート 1400 に送信されたトラフィックと一致するよう MGCP（メディア ゲートウェイ制御プロトコル）を設定する必要があります。

サポート対象プロトコル

FWSM は、次のインスペクション エンジンをサポートしています。

- CTIQBE — 「CTIQBE 検査」（p.20-9）を参照してください。
- Domain Name System（DNS; ドメイン ネーム システム）— 「DNS 検査」（p.20-13）を参照してください。
- FTP（ファイル転送プロトコル）— 「FTP 検査」（p.20-22）を参照してください。
- GTP — 「GTP 検査」（p.20-28）を参照してください。
- H.323 — 「H.323 検査」（p.20-34）を参照してください。
- HTTP — 「HTTP 検査」（p.20-42）を参照してください。
- Internet Control Message Protocol（ICMP）—『*Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*』の **inspect icmp** および **inspect icmp error** コマンド ページを参照してください。
- ILS —『*Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*』の **inspect ils** コマンド ページを参照してください。
- MGCP — 「MGCP 検査」（p.20-46）を参照してください。
- NetBIOS —『*Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*』の **inspect netbios** コマンド ページを参照してください。
- PPTP —『*Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*』の **inspect pptp** コマンド ページを参照してください。
- RSH —『*Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*』の **inspect rsh** コマンド ページを参照してください。
- RTSP — 「RTSP 検査」（p.20-53）を参照してください。
- SIP — 「SIP 検査」（p.20-57）を参照してください。
- Skinny — 「Skinny（SCCP）検査」（p.20-63）を参照してください。
- SMTP/ESMTP — 「SMTP および拡張 SMTP 検査」（p.20-67）を参照してください。
- SNMP — 「SNMP 検査」（p.20-70）を参照してください。
- SQL*Net —『*Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*』の **inspect sqlnet** コマンド ページを参照してください。
- SunRPC — 「Sun RPC 検査」（p.20-73）を参照してください。
- TFTP —『*Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*』の **inspect tftp** コマンド ページを参照してください。
- XDMCP —『*Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*』の **inspect xdmcp** コマンド ページを参照してください。

アプリケーション エンジンのデフォルト

表 20-1 に、FWSM でサポートされる各プロトコルに提供されたアプリケーション検査のタイプを要約します。

表 20-1　アプリケーション インスペクション エンジンおよびデフォルト

アプリケーション	デフォルトでのイネーブル化	PAT の有無	NAT（1 対 1）の有無	ポート設定の有無	デフォルトポート	標準	説明
CTIQBE	なし	あり	あり	あり	TCP/2748	—	—
DNS[1]	あり	あり	あり	なし	UDP/53	RFC 1123	NAT のみ転送します。PTR レコードの変更はありません。 デフォルトの最大パケット長は 512 バイトです。
FTP	あり	あり	あり	あり	TCP/21	RFC 959	デフォルトの FTP 検査は、RFC 規格への準拠を強制しません。強制するには、***strict*** キーワードを指定して **inspect ftp** コマンドを設定します。
GTP	なし	あり	あり	あり	UDP/3386 UDP/2123	—	特別なライセンスが必要です。
H.323	あり	あり	あり	あり	TCP/1720 UDP/1718 UDP（RAS） 1718 ～ 1719	ITU-T H.323、H.245、H225.0、Q.931、Q.932	デフォルトでは、RAS 検査と H.225 検査の両方がイネーブルです。
HTTP	なし	あり	あり	あり	TCP/80	RFC 2616	ActiveX および Java のストリッピング時の MTU 制限に注意してください[2]。
ICMP	なし	あり	あり	なし	—	—	—
ICMP ERROR	なし	あり	あり	なし	—	—	—
ILS（LDAP）	なし	あり	あり	あり	—	—	—
MGCP	なし	あり	あり	あり	2427、2727	RFC2705bis-05	—
NetBIOS Datagram Service / UDP	あり	あり	あり	なし	UDP/138	—	—
NetBIOS Name Service / UDP	あり	なし	なし	なし	UDP/137	—	WINS はサポートしません。
NetBIOS over IP[3]	あり	なし	なし	なし	—	—	—
PPTP	なし	あり	あり	あり	1723	RFC2637	—
RSH	あり	あり	あり	あり	TCP/514	Berkeley UNIX	—
RTSP	なし	なし	なし	あり	TCP/554	RFC 2326、RFC 2327、RFC 1889	HTTP クローキングは処理されません。
SIP	あり	あり	あり	あり	TCP/5060 UDP/5060	RFC 2543	—

表 20-1　アプリケーション インスペクション エンジンおよびデフォルト（続き）

アプリケーション	デフォルトでのイネーブル化	PAT の有無	NAT（1 対 1）の有無	ポート設定の有無	デフォルトポート	標準	説明
Skinny（SCCP）	あり	あり	あり	あり	TCP/2000	—	所定の状況では、TFTP でアップロードした Cisco IP Phone コンフィギュレーションは処理されません。
SNMP	あり	なし	なし	あり	UDP/161、162	RFC 1155、1157、1212、1213、1215	v.2 RFC 1902 ～ 1908、v.3 RFC 2570 ～ 2580
SMTP/ESMTP	あり	あり	あり	あり	TCP/25	RFC 821、1123	デフォルトでは、ESMTP 検査ではなく、SMTP 検査がイネーブルです。
SQL*Net	あり	あり	あり	あり	TCP/1521（v.1）	—	V.1 および v.2
Sun RPC	あり	なし	あり	なし	UDP/111 TCP/111	—	ペイロードは NAT 処理しません。
TFTP	あり	あり	あり	あり	TCP/69 UDP/69	RFC 1530	—
XDCMP	あり	なし	なし	なし	UDP/177	—	—

1. WINS による名前解決用の NAT はサポートされません。
2. MTU が小さすぎて Java タグまたは ActiveX タグを 1 つのパケットに納められない場合は、ストリッピングは行われません。
3. NetBIOS は、NetBIOS ネーム サービス UDP ポート 137 および NetBIOS データグラム サービス UDP ポート 138 に対してパケットの NAT 処理を実行することでサポートされます。

アプリケーション検査コンフィギュレーションの概要

アプリケーション インスペクション エンジンを設定するには、MPF コマンドを使用します。アプリケーション検査を設定する前に、第 18 章「モジュラ ポリシー フレームワークの使用」をお読みください。この章では、MPF の概念と、アプリケーション検査を設定するのに使用しなければならない共通コマンドについて説明しています。

アプリケーション インスペクション エンジンのイネーブル化と適用には、必ず以下が含まれます。

- FWSM がインスペクション エンジンに送信するトラフィックを識別するクラス マップ
- クラス マップ（および関連トラフィック）をインスペクション エンジンにリンクするポリシー マップ
- ポリシー マップを 1 つまたはすべてのインターフェイスに適用するサービス ポリシー

第 18 章「モジュラ ポリシー フレームワークの使用」では、MPF を構成する、すなわちアプリケーション検査を設定する前述の 3 つの要素に関する詳細な概要を示します。この章で述べたインスペクション エンジンについては、詳細な設定手順および設定例を示します。この章で言及しないインスペクション エンジンについては、『*Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*』の利用可能な **inspect** コマンド エントリを参照してください。

ただし、アプリケーション検査マップは例外です。これらのマップはアプリケーション検査コンフィギュレーションに固有のものです。アプリケーション検査マップにより、特定のインスペクション エンジン用に名前の付いたインスペクション パラメータのセットを作成します。検査マップをサポートするアプリケーション インスペクション エンジンにポリシー マップを設定する場合、名前で検査マップを指定できます。

次のプロトコルに対して、インスペクション エンジンはアプリケーション マップをサポートします。

- FTP — 詳細については、「request-command deny コマンド」（p.20-23）を参照してください。
- GTP — 詳細については、「GTP マップおよびコマンド」（p.20-29）を参照してください。
- HTTP — 詳細については、「拡張 HTTP 検査コマンド」（p.20-43）を参照してください。
- MGCP — 詳細については、「MGCP コール エージェントおよびゲートウェイの設定」（p.20-48）を参照してください。
- SIP — 詳細については、「IP アドレス プライバシー」（p.20-58）を参照してください。
- SNMP — 詳細については、「SNMP 検査の概要」（p.20-70）を参照してください。

デフォルトのアプリケーション検査

アプリケーション検査はデフォルトでは、すべてのプロトコルではないものの多くのプロトコルに対して、イネーブルです。表 20-1 には、アプリケーション インスペクション エンジンがデフォルトでイネーブルである場合の情報が示されています。ただし、デフォルトのポリシー コンフィギュレーションを検証して、デフォルトでイネーブルであるインスペクション エンジンを判別できます。内容は次のとおりです。

```
class-map inspection_default
 match default-inspection-traffic
policy-map global_policy
 class inspection_default
  inspect dns maximum-length 512
  inspect ftp
  inspect h323 h225
  inspect h323 ras
  inspect rsh
  inspect smtp
  inspect sqlnet
  inspect skinny
  inspect sunrpc
  inspect xdmcp
  inspect sip
  inspect netbios
  inspect tftp
service-policy global_policy global
```

CTIQBE 検査

ここでは、CTIQBE アプリケーション検査をイネーブルにして、デフォルトのポート設定を変更する手順について説明します。次の内容について説明します。

CTIQBE 検査の概要

inspect ctiqbe コマンドを使用すると、Computer Telephony Interface Quick Buffer Encoding（CTIQBE）プロトコル検査をイネーブルにします。これは、NAT、PAT、双方向 NAT をサポートします。これにより、Cisco IP SoftPhone および他の Cisco Telephony Application Programming Interface（TAPI）/Java Telephony Application Programming Interface（JTAPI）アプリケーションは、FWSM 上でコールセットアップのため、Cisco CallManager と連動できます。

TAPI および JTAPI は、多くの Cisco VoIP アプリケーションで使用されます。CTIQBE は、Cisco TAPI Service Provider（TSP; TAPI サービス プロバイダー）が Cisco CallManager と通信するために使用します。

制限事項および制約事項

次に、CTIQBE アプリケーション検査の使用時に適用される制限を要約します。

- CTIQBE アプリケーション検査では、**alias** コマンドを使用したコンフィギュレーションをサポートしていません。
- CTIQBE コールのステートフル フェールオーバーはサポートされて*いません*。
- **debug ctiqbe** コマンドを入力すると、メッセージの伝送が遅れる場合があり、リアルタイム環境のパフォーマンスに影響することがあります。このデバッグまたはログをイネーブルにし、FWSM を経由して Cisco IP SoftPhone でコール セットアップを完了できない場合は、Cisco IP SoftPhone が稼働するシステムで Cisco TSP 設定のタイムアウト値を増やします。

次に、CTIQBE アプリケーション検査を特定の事例で使用する場合に、特別に注意が必要な事項を要約します。

- 2 台の Cisco IP SoftPhone が異なる Cisco CallManager に登録されていて、各 CallManager が FWSM の異なるインターフェイスに接続されている場合、これら 2 台の電話間のコールは失敗します。
- Cisco IP SoftPhone と比較して Cisco CallManager の方がセキュリティの高いインターフェイス上に配置されており、NAT または外部 NAT が Cisco CallManager IP アドレスに必要な場合、マッピングはスタティックである必要があります。Cisco IP SoftPhone では Cisco CallManager IP アドレスを PC 上の Cisco TSP コンフィギュレーションで明示的に指定する必要があるからです。
- PAT または外部 PAT を使用しているときに Cisco CallManager IP アドレスを変換する場合、Cisco IP SoftPhone の登録を成功させるため、TCP ポート 2748 を PAT（インターフェイス）アドレスの**同一ポート**に対してスタティックにマッピングする必要があります。CTIQBE リスニング ポート（TCP 2748）は固定されていて、Cisco CallManager、Cisco IP SoftPhone、または Cisco TSP において、ユーザによる設定はできません。

CTIQBE 検査のイネーブル化および設定

CTIQBE 検査をイネーブルにする、または CTIQBE トラフィックの受信に使用するデフォルト ポートを変更する手順は、次のとおりです。

ステップ 1　CTIQBE トラフィックを識別するには、クラス マップを作成するか、または既存のクラス マップを変更します。**class-map** コマンドを次のように使用します。

```
hostname(config)# class-map class_map_name
hostname(config-cmap)#
```

class_map_name は、トラフィック クラスの名前です。**class-map** コマンドを入力すると、CLI はクラス マップ コンフィギュレーション モードを開始します。

ステップ 2　CTIQBE トラフィックを識別するには、次のように **match port** コマンドを使用します。

```
hostname(config-cmap)# match port tcp eq 2748
```

ステップ 3　CTIQBE インスペクション エンジンを FTP トラフィックに適用するために使用するポリシー マップを作成するか、または既存のポリシー マップを変更します。そのためには、**policy-map** コマンドを次のように使用します。

```
hostname(config-cmap)# policy-map policy_map_name
hostname(config-pmap)#
```

policy_map_name は、ポリシー マップの名前です。CLI はポリシー マップ コンフィギュレーション モードを開始し、プロンプトがそれに応じて変わります。

ステップ 4　ステップ 1 で作成したクラス マップを指定します。このクラス マップは CTIQBE トラフィックを識別します。**class** コマンドを次のように使用します。

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

class_map_name は、ステップ 1 で作成したクラス マップの名前です。CLI はポリシー マップ クラス コンフィギュレーション モードを開始し、プロンプトがそれに応じて変わります。

ステップ 5　CTIQBE アプリケーション検査をイネーブルにします。

```
hostname(config-pmap-c)# inspect ctiqbe
```

ステップ 6　ポリシー マップをグローバルに、または特定のインターフェイスに適用するには、**service-policy** コマンドを次のように使用します。

```
hostname(config-pmap-c)# service-policy policy_map_name [global | interface
interface_ID]
hostname(config)#
```

policy_map_name は、ステップ 3 で設定したポリシー マップです。ポリシー マップをすべてのインターフェイス上のトラフィックに適用する場合、**global** オプションを使用します。ポリシー マップを特定のインターフェイス上のトラフィックに適用する場合、***interface*** *interface_ID* オプションを使用します。*interface_ID* は、**nameif** コマンドでインターフェイスに割り当てられた名前です。

FWSM は、指定のとおりに CTIQBE トラフィックの検査を開始します。

例 20-1　CTIQBE 検査のイネーブル化および設定

次に、デフォルト ポート（2748）の CTIQBE トラフィックと一致し、CTIQBE トラフィックと一致するクラスを使用して、ポリシーで CTIQBE 検査をイネーブルにするクラス マップを作成する例を示します。それからサービス ポリシーを外部インターフェイスに適用します。

```
hostname(config)# class-map ctiqbe_port
hostname(config-cmap)# match port tcp eq 2748
hostname(config-cmap)# policy-map sample_policy
hostname(config-pmap)# class ctiqbe_port
hostname(config-pmap-c)# inspect ctiqbe
hostname(config-pmap-c)# service-policy sample_policy interface outside
hostname(config)#
```

CTIQBE 検査の確認およびモニタ

show ctiqbe コマンドは、FWSM を超えて確立された CTIQBE セッションに関する情報を表示します。このコマンドは、CTIQBE インスペクション エンジンによって割り当てられたメディア接続に関する情報を示します。

次に、以下の条件における **show ctiqbe** コマンドの出力例を示します。FWSM を越えてセットアップされているアクティブ CTIQBE セッションは 1 つだけです。そのセッションは、ローカル アドレス 10.0.0.99 の内部 CTI デバイス（たとえば、Cisco IP SoftPhone）と 172.29.1.77 の外部 Cisco Call Manager の間で確立されています。ここで、TCP ポート 2748 は Cisco CallManager です。このセッションのハートビート間隔は 120 秒です。

```
hostname# # show ctiqbe

Total: 1
        LOCAL           FOREIGN         STATE   HEARTBEAT
-----------------------------------------------------------------
1       10.0.0.99/1117  172.29.1.77/2748        1       120
        ---------------------------------------------
        RTP/RTCP: PAT xlates: mapped to 172.29.1.99(1028 - 1029)
        ---------------------------------------------
        MEDIA: Device ID 27     Call ID 0
               Foreign 172.29.1.99      (1028 - 1029)
               Local   172.29.1.88      (26822 - 26823)
        ---------------------------------------------
```

CTI デバイスは、CallManager に登録済みです。デバイスの内部アドレスと RTP リスニング ポートは、172.29.1.99 UDP ポート 1028 に PAT 変換されます。その RTCP リスニング ポートは、UDP 1029 に PAT 変換されます。

`RTP/RTCP:PAT xlates:` で始まる行は、内部 CTI デバイスが外部 CallManager に登録され、CTI デバイスのアドレスとポートがその外側インターフェイスに PAT 変換されている場合にのみ、表示されます。この行は、CallManager が内側インターフェイス上に位置する場合、または内部 CTI デバイスのアドレスとポートが、CallManager が使用しているのと同じ外側インターフェイスに NAT 処理されている場合は、表示されません。

この出力は、コールがこの CTI デバイスと 172.29.1.88 にある別の電話の間に確立されていることを示します。他の電話の RTP および RTCP リスニング ポートは、UDP 26822 および 26823 です。FWSM は 2 番めの電話と CallManager に関連する CTIQBE セッション レコードを維持できないので、他の電話は CallManager と同じインターフェイス上にあります。CTI デバイス側のアクティブ コール レグは、Device ID 27 および Call ID 0 で確認できます。

次に、これらの CTIBQE 接続に対する **show xlate debug** コマンドの出力例を示します。

```
hostname# show xlate debug
3 in use, 3 most used
Flags:  D - DNS, d - dump, I - identity, i - inside, n - no random,
        r - portmap, s - static
TCP PAT from inside:10.0.0.99/1117 to outside:172.29.1.99/1025 flags ri idle 0:00:22
timeout 0:00:30
UDP PAT from inside:10.0.0.99/16908 to outside:172.29.1.99/1028 flags ri idle 0:00:00
timeout 0:04:10
UDP PAT from inside:10.0.0.99/16909 to outside:172.29.1.99/1029 flags ri idle 0:00:23
timeout 0:04:10
```

show conn state ctiqbe　コマンドは、CTIQBE 接続のステータスを表示します。出力には、CTIQBE インスペクション エンジンによって割り当てられたメディア接続が「C」フラグで示されます。次に、**show conn state ctiqbe** コマンドの出力例を示します。

```
hostname# show conn state ctiqbe
1 in use, 10 most used
hostname# show conn state ctiqbe detail
1 in use, 10 most used
Flags: A - awaiting inside ACK to SYN, a - awaiting outside ACK to SYN,
       B - initial SYN from outside, C - CTIQBE media, D - DNS, d - dump,
       E - outside back connection, F - outside FIN, f - inside FIN,
       G - group, g - MGCP, H - H.323, h - H.225.0, I - inbound data,
       i - incomplete, J - GTP, j - GTP data, k - Skinny media,
       M - SMTP data, m - SIP media, O - outbound data, P - inside back connection,
       q - SQL*Net data, R - outside acknowledged FIN,
       R - UDP RPC, r - inside acknowledged FIN, S - awaiting inside SYN,
       s - awaiting outside SYN, T - SIP, t - SIP transient, U - up
```

DNS 検査

ここでは、DNS アプリケーション検査を管理する手順について説明します。次の内容について説明します。

DNS アプリケーション検査の動作

FWSM によって DNS 応答が転送されると、FWSM は DNS クエリーに対応付けられた DNS セッションをただちに終了します。FWSM はまた、DNS 応答の ID が DNS クエリーの ID と一致していることを確認するために、メッセージ交換をモニタします。

デフォルトで DNS 検査はイネーブルの場合、FWSM は、次の作業を追加します。

- **alias**、**static**、**nat** コマンドを使用して設定を作成し、これに基づいて DNS レコードを変換します（DNS Rewrite）。変換は DNS 応答の A レコードにのみ適用されます。したがって、DNS Rewrite は PTR レコードを要求するリバース検索に影響を及ぼしません。

（注）　複数の PAT 規則が各 A レコードに適用され、使用する PAT 規則があいまいになるので、DNS Rewrite は PAT には適用されません。

- DNS メッセージの最大長を適用します（デフォルトは 512 バイトで、最大長は 65,535 バイト）。パケット長が設定済みの最大長以下であるか確認するため、FWSM は必要に応じてリアセンブリします。パケット長が最大長を超えた場合、FWSM はパケットを廃棄します。

（注）　**maximum-length** オプションを指定しないで **inspect dns** コマンドを入力すると、DNS パケット サイズは検証されません。

- ドメイン名の長さを 255 バイトに、ラベルの長さを 63 バイトにします。
- 圧縮ポインタが DNS メッセージ内で発生すると、ポインタによって参照されるドメイン名の正当性を確認します。
- 圧縮ポインタのループが存在するかどうか検証します。

複数の DNS セッションが同じ 2 つのホストの間にあり、セッションが同じ 5 つのタプル（送信元 / 宛先 IP アドレス、送信元 / 宛先ポート、プロトコル）を取得していれば、単一の接続がこの複数の DNS セッション用に作成されます。DNS 識別は、*app_id* によって追跡され、各 app_id のアイドル タイマーは独立して動作します。

app_id は独立してタイムアウトになるので、正規の DNS 応答は制限された時間内で FWSM を通過するだけで、リソース構築はされません。ただし、**show conn** コマンドを入力すると、新しい DNS セッションによってリセットされる DNS 接続のアイドル タイマーが表示されます。これは共有の DNS 接続の特性と設計によるものです。

DNS Rewrite の動作

DNS 検査がイネーブルの場合、DNS Rewrite は任意のインターフェイスから発信される DNS メッセージの NAT を完全にサポートします。

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

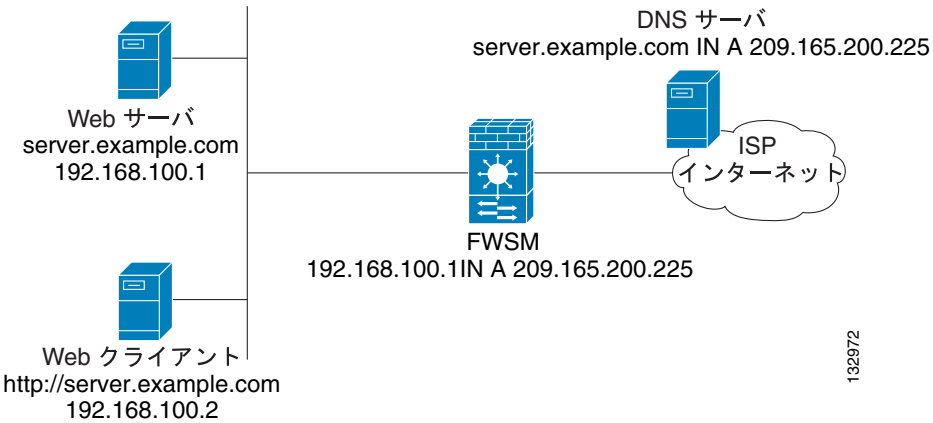
DNS 検査がイネーブルであれば、**alias**、**static** または **nat** コマンドを使用して、DNS Rewrite を設定できます。必要な設定の詳細については、「DNS Rewrite の設定」（p.20-15）を参照してください。

DNS Rewrite は次の 2 つの機能を実行します。

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

図 20-2 では、DNS サーバは外部（ISP）ネットワーク上に存在します。FWSM では、**static** コマンドは、Web サーバ（192.168.100.1）の実アドレスを ISP 割り当てアドレス（209.165.200.5）へマッピングします。内部インターフェイス上の Web クライアントが、URL http://server.example.com の Web サーバにアクセスしようとすると、Web クライアントが稼働するホストは Web サーバの IP アドレスを解決するため、DNS サーバに DNS 要求を送信します。FWSM は、IP ヘッダー内のルーティング不可能な送信元アドレスを変換して、その外部インターフェイス上の ISP ネットワークに要求を転送します。DNS 応答が返送されると、FWSM は宛先アドレスだけでなく、Web サーバの組み込み IP アドレスにもアドレス変換を適用します。この組み込み IP アドレスは、DNS 応答の A レコードに含まれています。結果として、内部ネットワーク上の Web クライアントは、内部ネットワーク上の Web サーバとの接続に必要な正しいアドレスを取得します。この例の NAT および DNS コンフィギュレーションの詳細については、例 20-2 を参照してください。これと同様の事例に関する設定手順については、「2 つの NAT ゾーンを使用した DNS Rewrite の設定」（p.20-16）を参照してください。

図 20-2　2 つの NAT ゾーンを使用した DNS Rewrite



DNS 要求を行うクライアントが DMZ ネットワーク上にあり、DNS サーバが内部インターフェイス上にある場合にも、DNS Rewrite は機能します。この事例に関する図および設定手順については、「3 つの NAT ゾーンを使用した DNS Rewrite」（p.20-17）を参照してください。

DNS Rewrite の設定

alias、**static**、または **nat** コマンドを使用して、DNS Rewrite を設定できます。**alias** および **static** コマンドは同じ意味で使用されます。ただし、**static** コマンドはより正確ではっきりしているので、新しい配置に使用することを推奨します。また、DNS Rewrite は **static** コマンド使用時のオプションです。

ここでは、**alias** および **static** コマンドを使用して DNS Rewrite を設定する手順について説明します。単純な事例や複雑な事例で、**static** コマンドを使用する設定手順を提供します。**nat** コマンドの使用は、DNS Rewrite がスタティック マッピングではなくダイナミック変換に基づいている点を除いて、**static** コマンドの使用と同様です。

次の内容について説明します。

- DNS Rewrite の alias コマンドの使用（p.20-15）
- DNS Rewrite の static コマンドの使用（p.20-15）
- 2 つの NAT ゾーンを使用した DNS Rewrite の設定（p.20-16）
- 3 つの NAT ゾーンを使用した DNS Rewrite（p.20-17）
- 3 つの NAT ゾーンを使用した DNS Rewrite の設定（p.20-19）

alias、**nat**、**static** コマンドの詳細な構文およびその他の機能については、『*Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*』の対応する コマンド ページを参照してください。

DNS Rewrite の alias コマンドの使用

alias コマンドを使用すると、FWSM は、*任意のインターフェイス*に存在する IP ネットワーク上のアドレスを、異なるインターフェイスを介して接続された別の IP ネットワークのアドレスに変換できます。このコマンドの構文は、次のとおりです。

```
hostname(config)# alias (inside) mapped-address real-address
```

次に、内部インターフェイス*以外の*任意のインターフェイス上の実アドレス（192.168.100.10）を、内部インターフェイス上でマッピングされたアドレス（209.165.200.225）に変換する例を示します。192.168.100.10 の場所は厳密に定義されていないことに注意してください。

```
hostname(config)# alias (inside) 209.165.200.225 192.168.100.10
```

（注）　DNS Rewrite を設定するのに **alias** コマンドを使用する場合、マッピングされたアドレスに対してプロキシ ARP が実行されます。これを回避するには、**alias** コマンドを入力したあとに、**sysopt noproxyarp internal_interface** コマンドを入力することで、プロキシ ARP をディセーブルにします。

DNS Rewrite の static コマンドの使用

static コマンドを使用すると、*特定のインターフェイス*に存在する IP ネットワーク上のアドレスを、異なるインターフェイス上の別の IP ネットワークのアドレスに変換できます。このコマンドの構文は、次のとおりです。

```
hostname(config)# static (inside,outside) mapped-address real-address dns
```

次に、内部インターフェイス上のアドレス 192.168.100.10 を外部インターフェイス上の 209.165.200.5 に変換する例を示します。

```
hostname(config)# static (inside,outside) 209.165.200.225 192.168.100.10 dns
```

（注）　**nat** コマンドの使用は、DNS Rewrite がスタティック マッピングではなくダイナミック変換に基づいている点を除いて、**static** コマンドの使用と同様です。

2 つの NAT ゾーンを使用した DNS Rewrite の設定

図 20-2 で示す事例のような DNS Rewrite の事例を実装する手順は、次のとおりです。

ステップ 1　次のように、Web サーバ用にスタティックな変換を作成します。

```
hostname(config)# static (inside,outside) mapped-address real-address netmask
255.255.255.255 dns
```

引数は次のとおりです。

- *inside* — FWSM の内部インターフェイスの名前
- *outside* — FWSM の外部インターフェイスの名前
- *mapped-address* — Web サーバの変換された IP アドレス
- *real-address* — Web サーバの IP 実アドレス

ステップ 2　HTTP 要求に対し、Web サーバが待ち受けるポートへのトラフィックを許可するアクセス リストを作成します。

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

引数は次のとおりです。

acl-name — アクセス リストに付けた名前

mapped-address — Web サーバの変換された IP アドレス

port — HTTP 要求に対し、Web サーバが待ち受ける TCP ポート

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

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

ステップ 4　DNS 検査がディセーブルである、または DNS 最大パケット長を変更する場合、DNS 検査を設定します。DNS アプリケーション検査はデフォルトではイネーブルで、DNS 最大パケット長は 512 バイトです。設定手順については、「DNS 検査の設定」（p.20-20）を参照してください。

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

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

domain-qualified-hostname は、server.example.com のようなドメイン サフィックスのあるホスト名です。ホスト名のあとのピリオドは重要です。*mapped-address* は、Web サーバの変換された IP アドレスです。

次に、図 20-2 で示す事例の FWSM を設定する例を示します。DNS 検査はすでにイネーブルであるとみなされます。

例 20-2　2 つの NAT ゾーンを使用した DNS Rewrite

```
hostname(config)# static (inside,outside) 209.165.200.225 192.168.100.1 netmask
255.255.255.255 dns
hostname(config)# access-list 101 permit tcp any host 209.165.200.225 eq www
hostname(config)# access-group 101 in interface outside
```

この設定は DNS サーバ上の次の A レコードを必要とします。

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

3 つの NAT ゾーンを使用した DNS Rewrite

図 20-3 に、より複雑な事例を示します。DNS 検査により、NAT は、最低限設定された DNS サーバを使用して透過的に動作できます。これと同様の事例に関する設定手順については、「3 つの NAT ゾーンを使用した DNS Rewrite の設定」（p.20-19）を参照してください。

図 20-3　3 つの NAT ゾーンを使用した DNS Rewrite

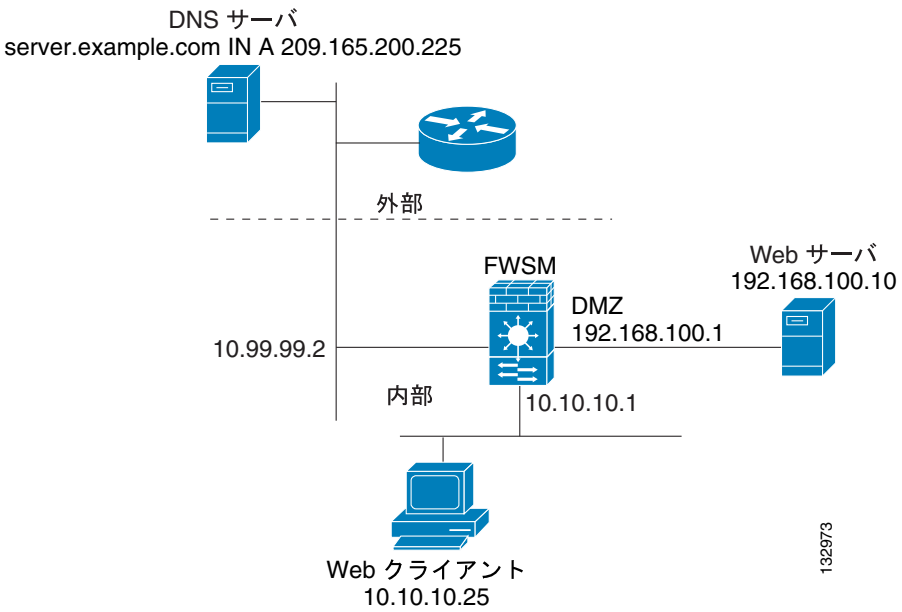


図 20-3 では、Web サーバ（server.example.com）には FWSM の DMZ インターフェイス上に実アドレス 192.168.100.10 があります。IP アドレス 10.10.10.25 を持つ Web クライアントは内部インターフェイス上にあり、パブリック DNS サーバは外部インターフェイス上にあります。サイト NAT ポリシーは次のとおりです。

- 外部 DNS サーバは、server.example.com 用に信頼性のあるアドレス レコードを保有します。
- 外部ネットワーク上のホストは、外部 DNS サーバ経由でドメイン名 server.example.com を持つ Web サーバ、または IP アドレス 209.165.200.225 を持つ Web サーバに接続できます。
- 内部ネットワーク上のクライアントは、外部 DNS サーバ経由でドメイン名 server.example.com を持つ Web サーバ、または IP アドレス 192.168.100.10 を持つ Web サーバにアクセスできます。

任意のインターフェイス上のホストまたはクライアントが DMZ Web サーバにアクセスする場合、サーバは server.example.com の A レコードに関して公開 DNS サーバに照会します。DNS サーバは、server.example.com がアドレス 209.165.200.225 にバインドすることを示す A レコードを戻します。

*外部*ネットワーク上の Web クライアントが http://server.example.com にアクセスしようとすると、次のイベントが発生します。

1. Web クライアントを実行するホストは、DNS サーバに server.example.com の IP アドレスの要求を送信します。
2. DNS サーバは、IP アドレス 209.165.200.225 で応答します。
3. Web クライアントはその HTTP 要求を 209.165.200.225 に送信します。
4. 外部ホストからのパケットは、外部インターフェイスで FWSM に到達します。
5. スタティック規則はアドレス 209.165.200.225 を 192.168.100.10 に変換し、FWSM は DMZ 上の Web サーバにパケットを転送します。

*内部*ネットワーク上の Web クライアントが http://server.example.com にアクセスしようとすると、次のイベントが発生します。

1. Web クライアントを実行するホストは、DNS サーバに server.example.com の IP アドレスの要求を送信します。
2. DNS サーバは、IP アドレス 209.165.200.225 で応答します。
3. FWSM は DNS 応答を受信し、DNS アプリケーション インスペクション エンジンに応答を送信します。
4. DNS アプリケーション インスペクション エンジンは以下を行います。
   a. 組み込み A レコード アドレス（[outside]:209.165.200.5）の変換を元に戻す NAT 規則を検索します。次に、スタティック コンフィギュレーションの例を示します。

      ```
      static (dmz,outside) 209.165.200.225 192.168.100.10 dns
      ```

   b. **dns** オプションが含まれているので、スタティック規則を使用して A レコードを次のように書き換えます。

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

      （注）　**dns** オプションは、**static** コマンドに含まれていないので、DNS Rewrite は実行されず、他のパケット処理が継続されます。

   c. 内部 Web クライアントと通信する場合、Web サーバ アドレス（[dmz]:192.168.100.10）を変換する NAT を検索します。

      NAT 規則は適用されないので、アプリケーション検査が完了します。

      NAT 規則（nat または static）を適用する場合、**dns** オプションを指定する必要があります。**dns** オプションを指定しない場合、ステップ b の A レコードの書き換えは無効になり、他のパケット処理が継続されます。
5. FWSM は、HTTP 要求を DMZ インターフェイス上の server.example.com に送信します。

3 つの NAT ゾーンを使用した DNS Rewrite の設定

図 20-3 の事例の NAT ポリシーをイネーブルにする手順は、次のとおりです。

ステップ 1　次のように、DMZ ネットワーク上の Web サーバにスタティックな変換を作成します。

```
hostname(config)# static (dmz,outside) mapped-address real-address dns
```

引数は次のとおりです。

- *dmz* — FWSM の DMZ インターフェイスの名前
- *outside* — FWSM の外部インターフェイスの名前
- *mapped-address* — Web サーバの変換された IP アドレス
- *real-address* — Web サーバの IP 実アドレス

ステップ 2　HTTP 要求に対し、Web サーバが待ち受けるポートへのトラフィックを許可するアクセス リストを作成します。

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

引数は次のとおりです。

acl-name — アクセス リストに付けた名前

mapped-address — Web サーバの変換された IP アドレス

port — HTTP 要求に対し、Web サーバが待ち受ける TCP ポート

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

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

ステップ 4　DNS 検査がディセーブルである、または DNS 最大パケット長を変更する場合、DNS 検査を設定します。DNS アプリケーション検査はデフォルトではイネーブルで、DNS 最大パケット長は 512 バイトです。設定手順については、「DNS 検査の設定」（p.20-20）を参照してください。

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

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

domain-qualified-hostname は、server.example.com のようなドメイン サフィックスのあるホスト名です。ホスト名のあとのピリオドは重要です。*mapped-address* は、Web サーバの変換された IP アドレスです。

次に、図 20-3 で示す事例の FWSM を設定する例を示します。DNS 検査はすでにイネーブルであるとみなされます。

例 20-3　3 つの NAT ゾーンを使用した DNS Rewrite

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

この設定は DNS サーバ上の次の A レコードを必要とします。

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

DNS 検査の設定

DNS 検査はデフォルトではイネーブルです。

DNS 検査をイネーブルにする（検査が以前にディセーブルであった場合に）、または DNS トラフィックの受信に使用するデフォルト ポートを変更するには、次の手順を実行します。

ステップ 1　DNS トラフィックを識別するには、クラス マップを作成するか、または既存のクラス マップを変更します。**class-map** コマンドを次のように使用します。

```
hostname(config)# class-map class_map_name
hostname(config-cmap)#
```

class_map_name は、トラフィック クラスの名前です。**class-map** コマンドを入力すると、CLI はクラス マップ コンフィギュレーション モードを開始します。

ステップ 2　**match port** コマンドを使用して、DNS トラフィックを識別します。DNS のデフォルト ポートは UDP ポート 53 です。

```
hostname(config-cmap)# match port udp eq 53
```

ステップ 3　DNS インスペクション エンジンを FTP トラフィックに適用するために使用するポリシー マップを作成するか、または 既存のポリシー マップを変更します。そのためには、**policy-map** コマンドを次のように使用します。

```
hostname(config-cmap)# policy-map policy_map_name
hostname(config-pmap)#
```

policy_map_name は、ポリシー マップの名前です。CLI はポリシー マップ コンフィギュレーション モードを開始し、プロンプトがそれに応じて変わります。

ステップ 4　DNS アプリケーション検査をイネーブルにします。そのためには、**inspect dns** コマンドを次のように使用します。

```
hostname(config-pmap-c)# inspect dns [maximum-length max-pkt-length]
```

DNS パケットの最大長をデフォルト（512）から変更するには、***maximum-length*** 引数を使用して、*max-pkt-length* に新しい数値を指定します。長いパケットは廃棄されます。DNS パケット長の検証をディセーブルにするには、**maximum-length** キーワードを指定しないで **inspect dns** コマンドを入力します。

ステップ 5　ポリシー マップをグローバルに、または特定のインターフェイスに適用するには、**service-policy** コマンドを次のように使用します。

```
hostname(config-pmap-c)# service-policy policy_map_name [global | interface
interface_ID]
hostname(config)#
```

policy_map_name は、ステップ 3 で設定したポリシー マップです。ポリシー マップをすべてのインターフェイス上のトラフィックに適用する場合、**global** オプションを使用します。ポリシー マップを特定のインターフェイス上のトラフィックに適用する場合、***interface*** *interface_ID* オプションを使用します。*interface_ID* は、**nameif** コマンドでインターフェイスに割り当てられた名前です。

FWSM は、指定のとおりに DNS トラフィックの検査を開始します。

例 20-4　DNS 検査のイネーブル化および設定

次に、デフォルト ポート（53）上の DNS トラフィックと一致するクラス マップを作成し、sample_policy ポリシー マップで DNS 検査をイネーブルにし、DNS 検査を外部インターフェイスに適用する例を示します。

```
hostname(config)# class-map dns_port
hostname(config-cmap)# match port udp eq 53
hostname(config-cmap)# policy-map sample_policy
hostname(config-pmap)# class dns_port
hostname(config-pmap-c)# inspect dns maximum-length 1500
hostname(config-pmap-c)# service-policy sample_policy interface outside
```

DNS 検査の確認およびモニタ

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 サーバを使用した接続については、接続の送信元ポートは、show conn コマンド出力で DNS サーバの IP アドレスに置き換えられる場合があります。

複数の DNS セッションが同じ 2 つのホストの間にあり、セッションが同じ 5 つのタプル（送信元 / 宛先 IP アドレス、送信元 / 宛先ポート、プロトコル）を取得していれば、単一の接続がこの DNS セッション用に作成されます。DNS 識別は、app_id によって追跡され、各 app_id のアイドル タイマーは独立して動作します。

app_id は独立してタイムアウトになるので、正規の DNS 応答は制限された時間内で FWSM を通過するだけで、リソース構築はされません。ただし、**show conn** コマンドを入力すると、新しい DNS セッションによってリセットされる 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 接続の特性と設計によるものです。

DNS アプリケーション検査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 検査

ここでは、FTP インスペクション エンジンの機能と、その設定を変更する手順について説明します。次の内容について説明します。

FTP 検査の概要

FTP アプリケーション検査は FTP セッションを検査し、4 つのタスクを実行します。

- 動的なセカンダリ データ接続の準備
- ***ftp*** コマンドの応答シーケンスの追跡
- 監査証跡の生成
- 組み込み IP アドレスの NAT 処理

FTP アプリケーション検査によって、FTP データ転送用にセカンダリ チャネルが用意されます。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。チャネルは、ファイルのアップロード、ファイルのダウンロード、またはディレクトリ リスト イベントの応答として割り当てられます。

（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した場合、発信ユーザが接続を開始できるのはパッシブ モードだけで、すべての着信 FTP はディセーブルになります。

strict オプションの使用

inspect ftp コマンドで **strict** オプションを使用すると、Web ブラウザが FTP 要求に組み込まれたコマンドを送信することを防止でき、保護ネットワークのセキュリティを向上させます。

ヒント　FWSM を通過することを許可されていない FTP コマンドを指定するには、FTP マップを作成し、FTP マップ コンフィギュレーション モードで **request-command deny** コマンドを入力します。

インターフェイスで **strict** オプションがイネーブルになったあと、FTP 検査は次の処理を実行します。

- FWSM が新しいコマンドを許可する前に、FTP コマンドに対して確認応答が返される必要があります。
- FWSM は、組み込みコマンドを送信する接続を廃棄します。
- 227 コマンドと PORT コマンドは、エラー文字列に表示されないように検証されます。

注意　*strict* オプションを使用すると、FTP RFC に厳密に準拠しない FTP クライアントの障害が発生する場合があります。

strict オプションがイネーブルである場合、次の異常動作について、各 *ftp* コマンドと応答シーケンスが追跡されます。

- 不完全なコマンド — PORT および PASV 応答コマンド内のカンマ数が 5 つかどうかが確認されます。5 つ以外の場合、PORT コマンドは不完全であるとみなされ、TCP 接続は終了します。
- 不正コマンド — RFC に規定されているように、*ftp* コマンドが <CR><LF> 文字で終了しているかどうかが確認されます。異なっている場合、接続は終了します。
- RETR コマンドおよび STOR コマンドのサイズ — 固定数になっているかどうかが確認されます。サイズが大きい場合、エラー メッセージが記録され、接続は終了します。
- 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る必要があります。PORT コマンドがサーバから送信されている場合、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る必要があります。PASV 応答コマンドがクライアントから送信されている場合、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」を実行した場合のセキュリティ ホールが防止されます。
- TCP ストリーム編集 — TCP ストリーム編集が検出されると、FWSM は接続を終了します。
- 無効なポートのネゴシエーション — ネゴシエートされたダイナミック ポートの値が 1024 未満かどうかが確認されます。1 ～ 1024 のポート番号は well-known 接続用に予約されているので、ネゴシエートされたポートがこの範囲内の場合には、TCP 接続は解放されます。
- コマンドのパイプライン化 — PORT コマンドおよび PASV 応答コマンド内のポート番号よりあとの文字数が、定数の 8 であるかどうかが相互確認されます。8 を超えている場合、TCP 接続は終了します。
- FWSM は、SYST コマンドに対する FTP サーバの応答を一連の X に置き換え、サーバが FTP クライアントに対し、そのシステム タイプを開示するのを回避します。このデフォルト動作を上書きするには、FTP マップ コンフィギュレーション モードで **no mask-syst-reply** コマンドを使用します。

request-command deny コマンド

request-command deny コマンドを使用すると、FWSM がどの FTP コマンドに対し、FTP トラフィックが FWSM を通過する許可を与えるか制御できます。このコマンドは FTP マップ コンフィギュレーション モードで利用可能です。そのため、このコマンドを利用するには、「FTP 検査の設定」（p.20-24）に従って FTP マップを作成し、FTP 検査をイネーブルにするときにそのマップを使用する必要があります。

表 20-2 に、**request-command deny** コマンドを使用することで、許可できなくする FTP コマンドを示します。

表 20-2　FTP マップの request-command deny オプション

request-command deny オプション	目的
appe	ファイルに追加するコマンドを許可しません。
cdup	現在のワーキング ディレクトリのペアレント ディレクトリに変わるコマンドを許可しません。
dele	サーバのファイルを削除するコマンドを許可しません。

表 20-2　FTP マップの request-command deny オプション（続き）

request-command deny オプション	目的
get	サーバからファイルを検索するクライアント コマンドを許可しません。
help	ヘルプ情報を提供するコマンドを許可しません。
mkd	サーバ上にディレクトリを作成するコマンドを許可しません。
put	サーバにファイルを送信するクライアント コマンドを許可しません。
rmd	サーバ上のディレクトリを削除するコマンドを許可しません。
rnfr	ファイル名から rename を指定するコマンドを許可しません。
rnto	ファイル名へ rename を指定するコマンドを許可しません。
site	サーバ システムに固有なコマンドを許可しません。通常はリモート管理用に使用されます。
stou	一意なファイル名を使用してファイルを保存するコマンドを許可しません。

FTP 検査の設定

FTP アプリケーション検査はデフォルトではイネーブルなので、次の方法で手順を実行する必要があるのはデフォルトの FTP 設定を変更する場合だけです。

- **strict** オプションをイネーブルにします。
- FWSM を通過することを許可されていない特定の FTP コマンドを識別します。
- デフォルトのポート番号を変更します。

FTP 検査を設定する手順は、次のとおりです。

ステップ 1　FWSM の後ろで FTP サーバが待ち受けるポートを決定します。デフォルトの FTP ポートは TCP ポート 21 です。ただし、thwart 攻撃に対する簡単な手段として、代替ポートがしばしば使用されます。すべての FTP トラフィックを検査することを確認するには、TCP ポート 21 以外のポートの使用に関して、FTP サーバを検証してください。

ステップ 2　FTP トラフィックを識別するには、クラス マップを作成するか、または既存のクラス マップを変更します。**class-map** コマンドを次のように使用します。

```
hostname(config)# class-map class_map_name
hostname(config-cmap)#
```

class_map_name は、トラフィック クラスの名前です。**class-map** コマンドを入力すると、CLI はクラス マップ コンフィギュレーション モードを開始します。

ステップ 3　ステップ 1 で決定した FTP ポートに送信されたトラフィックを識別します。そのためには、**match port** または **match access-list** コマンドを使用します。

連続しない複数のポートを特定する必要がある場合、**access-list extended** コマンドを使用してアクセス リストを作成し、各ポートと一致する Access Control Entry（ACE; アクセス制御エントリ）を追加してから、**match access-list** コマンドを使用します。次のコマンドは、アクセス リストを使用して、アクセス リストを持った複数の TCP ポートを特定します。

```
hostname(config)# access-list acl-name any any tcp eq port_number_1
hostname(config)# access-list acl-name any any tcp eq port_number_2
hostname(config)# class-map class_map_name
hostname(config-cmap)# match access-list acl-name
```

単一ポートを特定する必要がある場合、次のように **match port** コマンドを使用します。

```
hostname(config-cmap)# match port tcp port_number
```

port_number は、FWSM の後ろの FTP サーバが待ち受ける TCP ポートのみです。

単一プロトコルの連続したポート範囲を特定する必要がある場合、次のように **range** キーワードを指定して **match port** コマンドを使用します。

```
hostname(config-cmap)# match port tcp range begin_port_number end_port_number
```

begin_port_number は、FTP ポート範囲の最小ポートで、*end_port_number* は最大ポートです。

ステップ 4　（任意）FTP 検査を実行する場合は、次の手順を実行します。

- FTP クライアントに対して、FTP サーバのシステム タイプの開示を許可します。
- 許可された FTP コマンドを制限します。

FTP マップを作成して、設定します。そのためには、次の手順を実行します。

a. FTP 検査の追加パラメータを含んだ FTP マップを作成します。**ftp-map** コマンドを次のように使用します。

```
hostname(config-cmap)# ftp-map map_name
hostname(config-ftp-map)#
```

map_name は、FTP マップの名前です。CLI は、FTP マップ コンフィギュレーション コマンドを開始します。

b. （任意）SYST メッセージに応答して、FTP サーバにシステム タイプを FTP クライアントに開示させる場合、次のように **mask-syst-reply** コマンドの **no** 形式を使用します。

```
hostname(config-ftp-map)# no mask-syst-reply
hostname(config-ftp-map)#
```

（注）　デフォルトでは、FTP 検査がイネーブルである場合、SYST メッセージへの応答はマスクされます。SYST 応答マスキングをディセーブルにした場合、**mask-syst-response** コマンドを使用して、マスキングを再度イネーブルにできます。

c. （任意）特定の FTP コマンドを許可しない場合、**request-command deny** コマンドを使用して、許可しない各 FTP コマンドを次のように指定します。

```
hostname(config-ftp-map)# request-command deny ftp_command [ftp_command...]
hostname(config-ftp-map)#
```

ftp_command は、制限する 1 つまたは複数の FTP コマンドです。制限可能な FTP コマンドのリストについては、表 20-2 を参照してください。

ステップ 5　FTP インスペクション エンジンを FTP トラフィックに適用するために使用するポリシー マップを作成するか、または既存のポリシー マップを変更します。そのためには、**policy-map** コマンドを次のように使用します。

```
hostname(config-cmap)# policy-map policy_map_name
hostname(config-pmap)#
```

policy_map_name は、ポリシー マップの名前です。CLI はポリシー マップ コンフィギュレーション モードを開始し、プロンプトがそれに応じて変わります。

ステップ 6　ステップ 2 で作成したクラス マップを指定します。このクラス マップは FTP トラフィックを識別します。**class** コマンドを次のように使用します。

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

class_map_name は、ステップ 2 で作成したクラス マップの名前です。CLI はポリシー マップ クラス コンフィギュレーション モードを開始し、プロンプトがそれに応じて変わります。

ステップ 7　必要なオプションを指定して、FTP アプリケーション検査をイネーブルにします。そのためには、次の手順のいずれかを実行します。

- 完全な FTP 検査をイネーブルにする場合、次のように ***strict*** キーワードを指定して、**inspect ftp** コマンドを使用します。

  ```
  hostname(config-pmap-c)# inspect ftp strict
  ```

- ステップ 4 で設定したオプションの FTP マップを使用して完全な FTP 検査をイネーブルにする場合、次のように ***strict*** キーワードと FTP マップ名を指定して、**inspect ftp** コマンドを使用します。

  ```
  hostname(config-pmap-c)# inspect ftp strict ftp_map_name
  ```

- デフォルトの FTP 検査に戻す場合、次のように キーワードを指定しないで、**inspect ftp** コマンドを使用します。

  ```
  hostname(config-pmap-c)# inspect ftp
  ```

ステップ 8　ポリシー マップをグローバルに、または特定のインターフェイスに適用するには、**service-policy** コマンドを次のように使用します。

```
hostname(config-pmap-c)# service-policy policy_map_name [global | interface
interface_ID]
hostname(config)#
```

policy_map_name は、ステップ 5 で設定したポリシー マップです。ポリシー マップをすべてのインターフェイス上のトラフィックに適用する場合、**global** オプションを使用します。ポリシー マップを特定のインターフェイス上のトラフィックに適用する場合、***interface*** *interface_ID* オプションを使用します。*interface_ID* は、**nameif** コマンドでインターフェイスに割り当てられた名前です。

FWSM は、指定のとおりに FTP トラフィックの検査を開始します。

次に、FTP トラフィックを識別し、FTP マップを定義し、ポリシーを定義し、そのポリシーを外部インターフェイスに適用する例を示します。

例 20-5　完全な FTP 検査のイネーブル化および設定

```
hostname(config)# class-map ftp_port
hostname(config-cmap)# match port tcp eq 21
hostname(config-cmap)# ftp-map sample_map
hostname(config-ftp-map)# request-command deny put stou appe
hostname(config-ftp-map)# policy-map sample_policy
hostname(config-pmap)# class ftp_port
hostname(config-pmap-c)# inspect ftp strict sample_map
hostname(config-pmap-c)# service-policy sample_policy interface outside
```

FTP 検査の確認およびモニタ

FTP アプリケーション検査では、次のログ メッセージが生成されます。

- 取り込まれた、またはアップロードされた各ファイルについて、監査記録 302002 が生成されます。
- FTP コマンドが RETR または STOR であるかを判断するため検証され、retrieve コマンドおよび store コマンドが記録されます。
- ユーザ名は、IP アドレスを提供するテーブルを検索することで取得されます。
- ユーザ名、送信元 IP アドレス、宛先 IP アドレス、NAT アドレス、ファイル操作が記録されます。
- メモリ不足によってセカンダリ ダイナミック チャネルの準備に失敗した場合、監査記録 201005 が生成されます。

NAT と連携することにより、FTP アプリケーション検査では、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。詳細については RFC 959 に規定されています。

GTP 検査

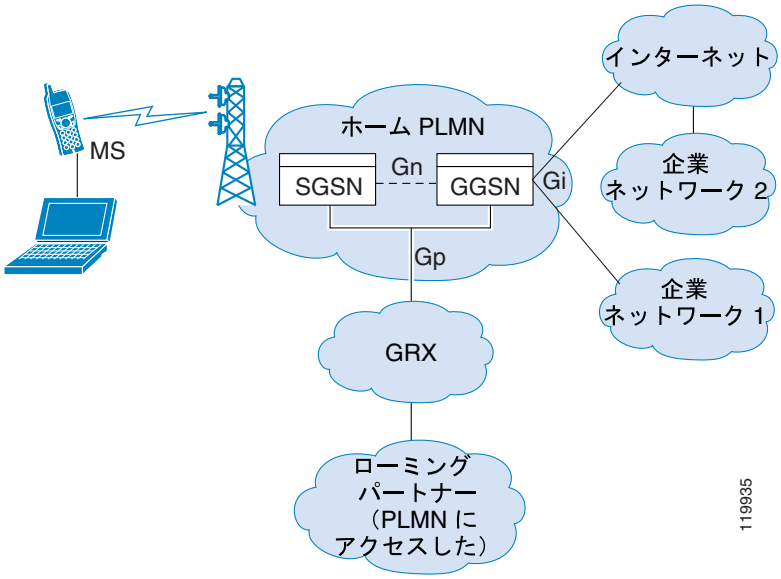
ここでは、GTP インスペクション エンジンの機能と、その設定を変更する手順について説明します。次の内容について説明します。

（注）　GTP 検査には、特別なライセンスが必要です。必要なライセンスなしで FWSM で GTP 関連コマンドを入力すると、FWSM はエラー メッセージを表示します。

GTP 検査の概要

General Packet Radio Service（GPRS）は、GSM ネットワークと企業ネットワーク、またはインターネットの間で、携帯電話加入者に連続した接続を提供します。Gateway GPRS Support Node（GGSN; ゲートウェイ GPRS サポート ノード）は、GPRS ワイヤレス データ ネットワークと他のネットワークの間のインターフェイスです。Serving GPRS Support Node（SGSN）は、モビリティ管理、データ セッション管理、データ圧縮を行います（図 20-4 を参照）。

図 20-4　GPRS トンネリング プロトコル



Universal Telecommunications System（UMTS）は、固定回線電話、携帯電話、インターネット、コンピュータ技術を融合したものです。Universal Terrestrial Radio Access Network（UTRAN）は、このシステムでワイヤレス ネットワークを実装するのに使用するネットワーキング プロトコルです。GTP により、GGSN、SGSN、UTRAN の間の UMTS/GPRS バックボーンを介して、マルチプロトコル パケットをトンネリングできます。

GTP には、固有のセキュリティまたはユーザ データの暗号化は含まれませんが、FWSM で GTP を使用するとリスクからネットワークを保護できます。

SGSN は GTP を使用して、論理的に GGSN に接続されています。GTP により、マルチプロトコル パケットを GSN の間の GPRS バックボーンを介してトンネリングできます。GTP は、トンネルを作成、変更、削除することで、SGSN にモバイル ステーションの GPRS ネットワーク アクセスを許可するトンネル制御および管理プロトコルを提供します。GTP は、ユーザ データ パケットの伝送サービスを提供するため、トンネリング メカニズムを使用します。

（注）フェールオーバーのある GTP を使用して GTP 接続が確立され、データがトンネル上で転送される前にアクティブ ユニットが失敗した場合、GTP データ接続（[j] フラグ セット付き）はスタンバイ ユニットに複製されません。これが発生するのは、アクティブ ユニットがスタンバイ ユニットへの初期接続を複製しないからです。

GTP マップおよびコマンド

GTP トラフィック上に追加の検査パラメータを実行できます。**gtp-map** コマンドは、検査パラメータを指定します。**inspect gtp** コマンドを使用して GTP 検査をイネーブルにする場合、GTP マップを指定するオプションがあります。

inspect gtp コマンドを使用してマップを指定しない場合、FWSM はデフォルトの GTP マップを使用します。このマップは、次のデフォルト値で設定済みのものです。

- **request-queue** 200
- **timeout gsn** 0:30:00
- **timeout pdp-context** 0:30:00
- **timeout request** 0:01:00
- **timeout signaling** 0:30:00
- **timeout tunnel** 0:01:00
- **tunnel-limit** 500

表 20-3 に、GTP 検査パラメータの設定に使用するコマンドを要約します。次のコマンドは、GTP マップ コンフィギュレーション モードで利用できます。各コマンドの詳細な構文については、『*Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*』の対応するコマンド ページを参照してください。

表 20-3　GTP マップ コンフィギュレーション コマンド

コマンド	説明
description	GTP コンフィギュレーション マップの説明を指定します。
drop	廃棄するメッセージ ID、APN、または GTP バージョンを指定します。
mcc	3 桁の Mobile Country Code（000 ～ 999）を指定します。1 桁または 2 桁のエントリの場合は前に 0 が付きます。
message-length	最小および最大メッセージ長を指定します。
permit errors	エラーのあるパケットまたは異なる GTP バージョンのパケットを許可します。
request-queue	キューで許可される要求の最大数を指定します。
timeout（**gtp-map**）	GSN、PDP コンテキスト、要求、シグナリング接続、トンネルのアイドル タイムアウトを指定します。
tunnel-limit	許可されるトンネルの最大数を指定します。

GTP 検査のイネーブル化および設定

GTP アプリケーション検査はデフォルトではディセーブルなので、GTP 検査をイネーブルにするには、ここで説明する手順を実行する必要があります。

（注）　GTP 検査には、特別なライセンスが必要です。必要なライセンスなしで FWSM で GTP 関連コマンドを入力すると、FWSM はエラー メッセージを表示します。

GTP 設定をイネーブルにする、または変更する手順は、次のとおりです。

ステップ 1　GTP トラフィックに必要なポートを特定する ACE を持つアクセス リストを定義します。標準ポートは、UDP ポート 2123 および 3386 です。アクセス リストを作成するには、次のように各 ACE に 1 回だけ **access-list extended** コマンドを使用します。

```
hostname(config)# access-list acl-name permit {udp | tcp} any any eq port
```

acl-name はアクセス リストに割り当てられた名前で、*port* は ACE が識別する GTP ポートです。

ステップ 2　GTP トラフィックを識別するには、クラス マップを作成するか、または 既存のクラス マップを変更します。**class-map** コマンドを次のように使用します。

```
hostname(config)# class-map class_map_name
hostname(config-cmap)#
```

class_map_name は、トラフィック クラスの名前です。**class-map** コマンドを入力すると、CLI はクラス マップ コンフィギュレーション モードを開始します。

ステップ 3　**match access-list** コマンドを使用して、ステップ 1 で作成したアクセス リストで GTP トラフィックを識別します。

```
hostname(config-cmap)# match access-list acl-name
```

ステップ 4　（任意）GTP トラフィック上で追加パラメータを実行する場合、GTP マップを作成および設定します。GTP マップを指定しない場合の GTP マップと実行されるデフォルト値の詳細については、「GTP マップおよびコマンド」（p.20-29）を参照してください。GTP マップを作成および設定する手順は、次のとおりです。

a.　GTP 検査の追加パラメータを含んだ GTP マップを作成します。**gtp-map** コマンドを次のように使用します。

```
hostname(config-cmap)# gtp-map map_name
hostname(config-gtp-map)#
```

map_name は、GTP マップの名前です。CLI は、GTP マップ コンフィギュレーション モードを開始します。

b.　GTP 検査のパラメータ設定します。そのためには、実行する GTP マップ コンフィギュレーション モード コマンドを使用します。コマンド リストについては、表 20-3 を参照してください。

ステップ 5　GTP インスペクション エンジンを GTP トラフィックに適用するために使用するポリシー マップを作成するか、または既存のポリシー マップを変更します。そのためには、**policy-map** コマンドを次のように使用します。

```
hostname(config-cmap)# policy-map policy_map_name
hostname(config-pmap)#
```

policy_map_name は、ポリシー マップの名前です。CLI はポリシー マップ コンフィギュレーション モードを開始し、プロンプトがそれに応じて変わります。

ステップ 6　ステップ 2 で作成したクラス マップを指定します。このクラス マップは GTP トラフィックを識別します。**class** コマンドを次のように使用します。

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

class_map_name は、ステップ 2 で作成したクラス マップの名前です。CLI はポリシー マップ クラス コンフィギュレーション モードを開始し、プロンプトがそれに応じて変わります。

ステップ 7　GTP アプリケーション検査をイネーブルにします。そのためには、**inspect gtp** コマンドを次のように使用します。

```
hostname(config-pmap-c)# inspect gtp [map_name]
hostname(config-pmap-c)#
```

map_name は、ステップ 4 （任意）で作成した GTP マップです。

ステップ 8　ポリシー マップをグローバルに、または特定のインターフェイスに適用するには、**service-policy** コマンドを次のように使用します。

```
hostname(config-pmap-c)# service-policy policy_map_name [global | interface
interface_ID]
hostname(config)#
```

policy_map_name は、ステップ 5 で設定したポリシー マップです。ポリシー マップをすべてのインターフェイス上のトラフィックに適用する場合、**global** オプションを使用します。ポリシー マップを特定のインターフェイス上のトラフィックに適用する場合、***interface*** *interface_ID* オプションを使用します。*interface_ID* は、**nameif** コマンドでインターフェイスに割り当てられた名前です。

FWSM は、指定のとおりに GTP トラフィックの検査を開始します。

次に、アクセス リストを使用して、GTP トラフィックを識別し、GTP マップを定義し、ポリシーを定義し、そのポリシーを外部インターフェイスに適用する例を示します。

例 20-6　GTP 検査のイネーブル化および設定

```
hostname(config)# access-list gtp_acl permit udp any any eq 3386
hostname(config)# access-list gtp_acl permit udp any any eq 2123
hostname(config)# class-map gtp-traffic
hostname(config-cmap)# match access-list gtp_acl
hostname(config-cmap)# gtp-map sample_map
hostname(config-gtp-map)# request-queue 300
hostname(config-gtp-map)# permit mcc 111 mnc 222
hostname(config-gtp-map)# message-length min 20 max 300
hostname(config-gtp-map)# drop message 20
hostname(config-gtp-map)# tunnel-limit 10000
hostname(config)# policy-map sample_policy
hostname(config-pmap)# class gtp-traffic
hostname(config-pmap-c)# inspect gtp sample_map
hostname(config)# service-policy sample_policy outside
```

GTP 検査の確認およびモニタ

GTP 設定を表示するには、イネーブル EXEC モードで **show service-policy inspect gtp** コマンドを入力します。このコマンドの詳細な構文については、『*Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*』のコマンド ページを参照してください。

GTP 検査の統計情報を表示するには、**show service-policy inspect gtp statistics** コマンドを使用します。次に、**show service-policy inspect gtp statistics** コマンドの出力例を示します。

```
hostname# show service-policy inspect gtp statistics
GPRS GTP Statistics:
  version_not_support                0     msg_too_short             0
  unknown_msg                        0     unexpected_sig_msg        0
  unexpected_data_msg                0     ie_duplicated             0
  mandatory_ie_missing               0     mandatory_ie_incorrect    0
  optional_ie_incorrect              0     ie_unknown                0
  ie_out_of_order                    0     ie_unexpected             0
  total_forwarded                    0     total_dropped             0
  signalling_msg_dropped             0     data_msg_dropped          0
  signalling_msg_forwarded           0     data_msg_forwarded        0
  total created_pdp                  0     total deleted_pdp         0
  total created_pdpmcb               0     total deleted_pdpmcb      0
  pdp_non_existent                   0
```

表示をフィルタリングするには、縦棒（|）を使用します。詳細なフィルタ オプションを表示するには、**?|** と入力します。

PDP コンテキスト関連情報を表示するには、**show service-policy inspect gtp pdp-context** コマンドを使用します。次に、**show service-policy inspect gtp pdp-context** コマンドの出力例を示します。

```
hostname# show service-policy inspect gtp pdp-context detail
1 in use, 1 most used, timeout 0:00:00

Version TID                  MS Addr        SGSN Addr    Idle      APN
v1      1234567890123425         10.0.1.1        10.0.0.2 0:00:13  gprs.example.com

    user_name (IMSI): 214365870921435    MS address:         1.1.1.1
    primary pdp: Y                       nsapi: 2
    sgsn_addr_signal:        10.0.0.2    sgsn_addr_data:          10.0.0.2
    ggsn_addr_signal:        10.1.1.1    ggsn_addr_data:          10.1.1.1
    sgsn control teid:     0x000001d1    sgsn data teid:        0x000001d3
    ggsn control teid:     0x6306ffa0    ggsn data teid:        0x6305f9fc
    seq_tpdu_up:                    0    seq_tpdu_down:                  0
    signal_sequence:                0
    upstream_signal_flow:          0     upstream_data_flow:          0
    downstream_signal_flow:        0     downstream_data_flow:        0
    RAupdate_flow:                 0
```

PDP コンテキストはトンネル ID によって識別されます。トンネル ID は IMSI 値と NSAPI 値を組み合わせたものです。GTP トンネルは、異なる GSN ノードで対応付けられた 2 つの PDP コンテキストによって定義され、トンネル ID で識別されます。GTP トンネルは、外部パケット データ ネットワークと MS ユーザの間でパケットを転送するのに必要です。

次に、縦棒（|）を使用して表示をフィルタリングする例を示します。

```
hostname# show service-policy gtp statistics | grep gsn
```

H.323 検査

ここでは、H.323 アプリケーション検査をイネーブルにして、デフォルトのポート設定を変更する手順について説明します。次の内容について説明します。

H.323 検査の概要

H.323 検査は、H.323 準拠アプリケーション（Cisco CallManager および VocalTec Gatekeeper など）をサポートします。H.323 は、International Telecommunication Union（ITU; 国際電気通信連合）により、LAN 上でのマルチメディア会議用として定義されているプロトコル スイートです。FWSM では、H.323 v3 機能の Multiple Calls on One Call Signaling Channel（1 つのコール シグナリング チャネルでの複数コール）を含め、H.323 Version 4 がサポートされます。

H.323 検査がイネーブルの場合、FWSM では、H.323 Version 3 で導入された機能である同一コール シグナリング チャネルでの複数コールをサポートします。この機能によってセットアップ時間が短縮され、FWSM でのポート使用が減少します。

H.323 検査には、2 つの主要機能があります。

- H.225 および H.245 メッセージ内に組み込まれた必要な IPv4 アドレスに対し、NAT を実行します。H.323 メッセージは PER 符号化フォーマットで符号化されているので、FWSM は ASN.1 デコーダを使用して H.323 メッセージをデコードします。
- ネゴシエートされた H.245 および RTP/RTCP 接続が、動的に割り当てられます。

H.323 の動作

H.323 のプロトコルは、合計で、最大 2 つの TCP 接続と 4 ～ 6 つの UDP 接続を使用できます。FastConnect は 1 つの TCP 接続だけを使用します。RAS は登録、アドミッション、ステータスに 1 つの UDP 接続を使用します。

H.323 クライアントでは、最初に TCP ポート 1720 を使用して H.323 サーバへの TCP 接続を確立し、Q.931 コールのセットアップを要求できます。コール セットアップ プロセスの一環として、H.323 端末は、H.245 TCP 接続に使用するポート番号をクライアントに提供します。H.323 ゲートキーパを使用している環境では、最初のパケットは UDP によって送信されます。

H.323 検査は、Q.931 TCP 接続をモニタして、H.245 のポート番号を判別します。H.323 端末が FastConnect を使用していない場合、FWSM は、H.225 メッセージの検査に基づいて H.245 接続を動的に割り当てます。

各 H.245 メッセージ内で、H.323 エンドポイントは、以降の UDP データ ストリームに使用するポート番号を交換します。H.323 検査は、H.245 メッセージを検査してこれらのポートを識別し、メディア交換用の接続を動的に作成します。Real-Time Transport Protocol（RTP）はネゴシエートされたポート番号を使用しますが、RTP Control Protocol（RTCP）は次の上位ポート番号を使用します。

H.323 制御チャネルは、H.225、H.245、および H.323 RAS を処理します。H.323 検査は、次のポートを使用します。

- UDP ポート 1718 — ゲートキーパ検出
- UDP ポート 1719 — RAS
- TCP ポート 1720 — 制御ポート

H.225 コール シグナリングについて、well-known H.323 ポート 1720 のトラフィックを許可しておく必要があります。ただし、H.245 シグナリング ポートは、H.225 シグナリング内のエンドポイントの間でネゴシエートされます。H.323 ゲートキーパが使用されている場合、FWSM は、AdmissionConfirm（ACF）メッセージの検査に基づいて H.225 接続を開始します。

FWSM は、H.225 メッセージを検査したあと、H.245 チャネルを開き、H.245 チャネル上で送信されたトラフィックを同様に検査します。すなわち、FWSM を通過した H.245 メッセージはすべて H.245 アプリケーション検査を通過し、組み込み IP アドレスが NAT 処理され、H.245 メッセージでネゴシエートされたメディア チャネルが開始されることを意味します。

H.323 ITU 標準規格では、信頼性のある接続上に送信する前に、H.225 および H.245 の前に TPKT ヘッダーによりメッセージの長さを定義することが規定されています。TPKT ヘッダーは H.225 および H.245 メッセージと同じ TCP パケットで送信されるとは限らないので、メッセージを適切に処理およびデコードするには、FWSM で TPKT 長を保持しておく必要があります。FWSM は各接続に対して、1 つのレコードを維持します。このレコードには次に送信されるメッセージの TPKT 長が含まれます。

FWSM でメッセージ内の IP アドレスを NAT 処理する必要がある場合、チェックサム、User-User Information Element（UUIE; ユーザ対ユーザ情報要素）の長さ、TPKT（H.225 メッセージの TCP パケットに含まれている場合）を変更する必要があります。TPKT が別の TCP パケットで送信される場合には、FWSM は TPKT のプロキシ ACK を実行し、H.245 メッセージに新しい長さの TPKT を付加します。

（注）　FWSM による TPKT のプロキシ ACK では、TCP オプションはサポートされません。

H.323 検査を通過するパケットを使用する各 UDP 接続は、H.323 接続としてマークされ、***timeout*** コマンドによって設定した H.323 タイムアウトが適用されます。

制限事項および制約事項

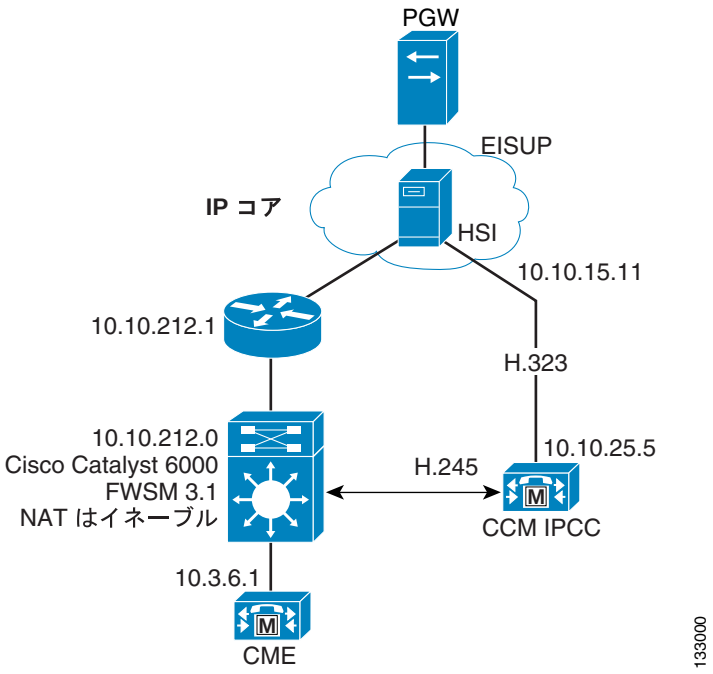
H.323 アプリケーション検査の一部の既知の問題と制限事項は、次のとおりです。

- スタティック PAT は、H.323 メッセージ内のオプション フィールドに組み込まれた IP アドレスを正しく変換しない場合があります。この種類の問題が発生したら、H.323 ではスタティック PAT を使用しないでください。
- NetMeeting クライアントが H.323 ゲートキーパに登録し、H.323 ゲートキーパに登録された H.323 ゲートウェイを呼び出そうとする場合、接続は確立されますが、音声はどの方向でも聞こえません。この問題は、FWSM とは無関係です。
- ネットワーク スタティック アドレスがサードパーティのネットマスクおよびアドレスと同じであるときに、ネットワーク スタティック アドレスを設定する場合、発信 H.323 接続は切断されます。

H.225 設定を必要とするトポロジー

FWSM を介して接続している H.323 エンドポイントの間でコール制御が発生したトポロジーでは、一部の追加 H.225 設定が必要です（図 20-5 を参照）。

図 20-5　H.225 設定を必要とするトポロジー



このトポロジーでは、FWSM の片側の Cisco CallManager と HSI の間で、もう片方の側の HSI と Cisco CallManager エンドポイントの間で、コール シグナリングが発生します。それから、Cisco CallManager と Cisco CallManager エンドポイントの間でコール制御が発生します。HSI と 1 つのエンドポイントが FWSM によって保護されたネットワーク上にあり、もう 1 つのエンドポイントが別のネットワーク上にある場合、H.225 設定を追加しないとコール制御は実施されません。

FWSM は、このトポロジーの Cisco CallManager の存在を認識していません。ファイアウォール経由で発生するパケット フローのみでは、FWSM はコールを成功させる適切なピンホールをオープンできません。したがって、この事例ではいくつかの追加 H.225 設定が必要となります。

必要な設定を提供するには、HSI と HSI グループ内の関連エンドポイントを特定します。設定が完了すると、FWSM は HSI を H.225 接続の通信ホストの 1 つとみなし、HSI グループ内のエンドポイントの間で H.245 ホールをオープンします。実際の H.245 接続は、これらのピンホールの 1 つと一致し、正しく実行されます。

H.225 マップ コマンド

H.225 マップにより、HSI が H.225 コール シグナリングに関与している場合に、FWSM は H.245 接続のため、動的なポート固有のピンホールをオープンします。H.225 マップは、HSI および関連するエンドポイントに関する情報を提供します。この情報は、FWSM によって保護されたネットワーク セキュリティを損なうことなく、この接続を確立するのに必要です。

h225-map コマンドは、H.225 マップを作成します。1 つの H225 マップには、最大 5 つの HSI グループを含めることができます。表 20-4 に、H.225 マップ コンフィギュレーション モードで利用可能なコマンドを示します。

表 20-4　H.225 コンフィギュレーション コマンド

コマンド	コンフィギュレーション モード	説明
hsi-group	H.225 マップ コンフィギュレーション モード	HSI グループを定義し、HSI グループ コンフィギュレーション モードを開始します。各 HSI グループには最大で 10 個のエンドポイントを含めることができます。
hsi	HSI グループ コンフィギュレーション モード	HSI を特定します。
endpoint	HSI グループ コンフィギュレーション モード	HSI グループ内の 1 つまたは複数のエンドポイントを特定します。

H.323 検査のイネーブル化および設定

H.323 検査はデフォルトではイネーブルです。

H.225 マップのオプション使用を含めた、H.323 検査をイネーブルにする手順は、次のとおりです。

ステップ 1　H.323 トラフィックに必要なポートを特定する ACE を持つアクセス リストを定義します。標準ポートは、UDP ポート 1718 と 1719、TCP ポート 1720 です。アクセス リストを作成するには、次のように各 ACE に 1 回だけ **access-list extended** コマンドを使用します。

```
hostname(config)# access-list acl-name permit {udp | tcp} any any eq port
```

acl-name はアクセス リストに割り当てられた名前で、*port* は ACE が識別する H.323 ポートです。

ステップ 2　H.323 トラフィックを識別するには、クラス マップを作成するか、または既存のクラス マップを変更します。**class-map** コマンドを次のように使用します。

```
hostname(config)# class-map class_map_name
hostname(config-cmap)#
```

class_map_name は、トラフィック クラスの名前です。**class-map** コマンドを入力すると、CLI はクラス マップ コンフィギュレーション モードを開始します。

ステップ 3　**match access-list** コマンドを使用して、ステップ 1 で作成したアクセス リストで H.323 トラフィックを識別します。

```
hostname(config-cmap)# match access-list acl-name
```

ステップ 4　（任意）ネットワーク トポロジーで必要な場合、H.225 マップを設定します。ネットワークが H.225 マップを必要とするかどうかについては、「H.225 設定を必要とするトポロジー」（p.20-36）を参照してください。H.225 マップを作成および設定する手順は、次のとおりです。

a. H.225 マップを作成します。

```
hostname(config)# h225-map map_name
hostname(config-h225-map)#
```

システムは H.225 マップ コンフィギュレーション モードを開始し、CLI プロンプトがそれに応じて変わります。

b. HSI グループを識別します。そのためには、**hsi-group** コマンドを次のように使用します。

```
hostname(config-h225-map)# hsi-group group_ID
hostname(config-h225-map-hsi-grp)#
```

group_ID は、0 ～ 2147483647 の番号で、HSI グループを識別します。

（注）　H.225 マップ単位で許可された HSI グループの最大数は 5 です。

システムは HSI グループ コンフィギュレーション モードを開始し、CLI プロンプトがそれに応じて変わります。

c. グループの HIS を定義します。

```
hostname(config-h225-map-hsi-grp)# hsi ip_address
```

ip_address は、HIS のアドレスです。

d. 最大 10 個のエンドポイントを定義します。そのためには、エンドポイントごとに **endpoint** コマンドを 1 回、次のように使用します。

```
hostname(config-h225-map-hsi-grp)# endpoint ip_address interface
```

interface は、エンドポイントに接続された FWSM 上のインターフェイスです。*ip_address* はそのエンドポイントのアドレスです。

e. 追加の HSI グループを作成する場合、ステップ b ～ d を繰り返します。

ステップ 5　H.323 インスペクション エンジンを H.323 トラフィックに適用するために使用するポリシー マップを作成するか、または既存のポリシー マップを変更します。そのためには、**policy-map** コマンドを次のように使用します。

```
hostname(config-cmap)# policy-map policy_map_name
hostname(config-pmap)#
```

policy_map_name は、ポリシー マップの名前です。CLI はポリシー マップ コンフィギュレーション モードを開始し、プロンプトがそれに応じて変わります。

ステップ 6　ステップ 2 で作成したクラス マップを指定します。このクラス マップは H.323 トラフィックを識別します。**class** コマンドを次のように使用します。

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

class_map_name は、ステップ 2 で作成したクラス マップの名前です。CLI はポリシー マップ クラス コンフィギュレーション モードを開始し、プロンプトがそれに応じて変わります。

ステップ 7　H.323 アプリケーション検査をイネーブルにします。そのためには、**inspect h323** コマンドを次のように使用します。

```
hostname(config-pmap-c)# inspect h323 [h225 map_name]
hostname(config-pmap-c)#
```

map_name は、ステップ 4 （任意）で作成した H.225 マップです。

ステップ 8　ポリシー マップをグローバルに、または特定のインターフェイスに適用するには、**service-policy** コマンドを次のように使用します。

```
hostname(config-pmap-c)# service-policy policy_map_name [global | interface
interface_ID]
hostname(config)#
```

policy_map_name は、ステップ 5 で設定したポリシー マップです。ポリシー マップをすべてのインターフェイス上のトラフィックに適用する場合、**global** オプションを使用します。ポリシー マップを特定のインターフェイス上のトラフィックに適用する場合、***interface*** *interface_ID* オプションを使用します。*interface_ID* は、**nameif** コマンドでインターフェイスに割り当てられた名前です。

FWSM は、指定のとおりに H.323 トラフィックの検査を開始します。

例 20-7　H.225 マップを使用しない場合の H.323 検査の設定

次の例では、H.323 インスペクション エンジンをイネーブルにし、デフォルト ポート（1720）の H.323 トラフィックと一致するクラス マップを作成します。それからサービス ポリシーを外部インターフェイスに適用します。

```
hostname(config)# access-list h323_acl permit udp any any eq 1718
hostname(config)# access-list h323_acl permit udp any any eq 1719
hostname(config)# access-list h323_acl permit tcp any any eq 1720
hostname(config)# class-map h323-traffic
hostname(config-cmap)# match access-list h323_acl
hostname(config-cmap)# policy-map sample_policy
hostname(config-pmap)# class h323_port
hostname(config-pmap-c)# inspect h323 ras
hostname(config-pmap-c)# inspect h323 h225
hostname(config-pmap-c)# service-policy sample_policy interface outside
hostname(config)#
```

例 20-8 には、H.323 設定全体の一部として、2 つの HSI グループを持った H.225 マップが含まれます。

例 20-8　H.225 マップを使用した場合の H.323 検査の設定

```
hostname(config)# access-list h323_acl permit udp any any eq 1718
hostname(config)# access-list h323_acl permit udp any any eq 1719
hostname(config)# access-list h323_acl permit tcp any any eq 1720
hostname(config)# class-map h323-traffic
hostname(config-cmap)# match access-list h323_acl
hostname(config-cmap)# h225-map sample_map
hostname(config-h225-map)# hsi-group 1
hostname(config-h225-map-hsi-grp)# hsi 10.10.15.11
hostname(config-h225-map-hsi-grp)# endpoint 10.3.6.1 inside
hostname(config-h225-map-hsi-grp)# endpoint 10.10.25.5 outside
hostname(config-h225-map-hsi-grp)# policy-map sample_policy
hostname(config-pmap)# class h323_port
hostname(config-pmap-c)# inspect h323 ras
hostname(config-pmap-c)# inspect h323 h225 sample_map
hostname(config-pmap-c)# service-policy sample_policy interface outside
hostname(config)#
```

H.323 および H.225 タイムアウト値の設定

H.225 シグナリング接続が終了するまでのアイドル時間を設定するには、**timeout h225** コマンドを使用します。H.225 タイムアウトのデフォルト値は 1 時間です。

H.323 制御接続が終了するまでのアイドル時間を設定するには、**timeout h323** コマンドを使用します。デフォルトは 5 分です。

H.323 検査の確認およびモニタ

ここでは、H.323 セッションに関する情報を表示する手順について説明します。次の内容について説明します。

- H.225 セッションのモニタ（p.20-40）
- H.245 セッションのモニタ（p.20-41）
- H.323 RAS セッションのモニタ（p.20-41）

H.225 セッションのモニタ

show h225 コマンドは、FWSM を超えて確立された H.225 セッションに関する情報を表示します。**debug h323 h225 event**、**debug h323 h245 event**、**show local-host** コマンドとともに、このコマンドは、H.323 インスペクション エンジンの問題のトラブルシューティングに使用されます。

show h225、**show h245**、または **show h323-ras** コマンドを入力する前に、**pager** コマンドの設定を推奨します。多くのセッション レコードが存在し、**pager** コマンドが設定されていない場合、**show** コマンドの出力が最後まで到達するには、しばらく時間がかかることがあります。非常に膨大な接続数がある場合、デフォルトのタイムアウト値または設定された値に基づいて、セッションがタイムアウトするか検証します。セッションがタイムアウトしない場合、調査を必要とする問題があります。

次に、**show h225** コマンドの出力例を示します。

```
hostname# show h225
Total H.323 Calls: 1
1 Concurrent Call(s) for
    Local:   10.130.56.3/1040   Foreign: 172.30.254.203/1720
    1. CRV 9861
    Local:   10.130.56.3/1040   Foreign: 172.30.254.203/1720
0 Concurrent Call(s) for
    Local:   10.130.56.4/1050   Foreign: 172.30.254.205/1720
```

この出力では、現在 FWSM を通過しているアクティブ H.323 コールが 1 つ、ローカル エンドポイント 10.130.56.3 と外部のホスト 172.30.254.203 の間にあることを示します。また、これらの特定のエンドポイントの間に、同時コールが 1 つあり、そのコールの Call Reference Value（CRV）が 9861 であることを示します。

ローカル エンドポイント 10.130.56.4 と外部ホスト 172.30.254.205 に対して、同時コールは 0 です。つまり H.225 セッションがまだ存在しているものの、このエンドポイントの間にはアクティブ コールがないことを意味します。この状況は、**show h225** コマンドを実行したときに、コールはすでに終了しているが、H.225 セッションがまだ削除されていない場合に発生する可能性があります。または、2 つのエンドポイントが、「maintainConnection」を TRUE に設定しているため、TCP 接続をまだ開いたままにしていることを意味する可能性もあります。したがって、「maintainConnection」を再度 FALSE に設定するまで、または設定内の H.225 タイムアウト値に基づいてセッションがタイムアウトするまで、セッションは開いたままになります。

H.245 セッションのモニタ

show h245 コマンドは、スロースタートを使用しているエンドポイントが FWSM を超えて確立された H.245 セッションに関する情報を表示します。スロースタートは、コールの 2 つのエンドポイントが H.245 用の別の TCP コントロール チャネルを開いた場合です。ファスト スタートは、H.245 メッセージが H.225 コントロール チャネル上の H.225 メッセージの一部として交換された場合です。**debug h323 h245 event**、**debug h323 h225 event**、**show local-host** コマンドとともに、このコマンドは、H.323 インスペクション エンジンの問題のトラブルシューティングに使用されます。

次に、**show h245** コマンドの出力例を示します。

```
hostname# show h245
Total: 1
        LOCAL           TPKT    FOREIGN         TPKT
1       10.130.56.3/1041        0       172.30.254.203/1245    0
        MEDIA: LCN 258 Foreign 172.30.254.203 RTP 49608 RTCP 49609
                       Local   10.130.56.3 RTP 49608 RTCP 49609
        MEDIA: LCN 259 Foreign 172.30.254.203 RTP 49606 RTCP 49607
                       Local   10.130.56.3 RTP 49606 RTCP 49607
```

FWSM を越えたアクティブな H.245 コントロール セッションが、現在 1 つあります。ローカル エンドポイントは、10.130.56.3 であり、TPKT 値が 0 であることから、このエンドポイントからの次のパケットには TPKT ヘッダーがあると予測します。TKTP ヘッダーは、各 H.225/H.245 メッセージの前に置かれる 4 バイトのヘッダーです。このヘッダーで、この 4 バイトのヘッダーを含むメッセージの長さが分かります。外部のホストのエンドポイントは、172.30.254.203 であり、TPKT 値が 0 であることから、このエンドポイントからの次のパケットには TPKT ヘッダーがあると予測します。

これらのエンドポイント間でネゴシエートされたメディアには、258 という LCN（論理チャネル番号）があり、外部に 172.30.254.203/49608 という RTP IP アドレス / ポート ペアと 172.30.254.203/49609 という RTCP IP アドレス / ポート ペアを持ち、ローカルに 10.130.56.3/49608 という RTP IP アドレス / ポート ペアと 49609 という RTCP ポートを持っています。

259 という 2 番めの LCN には、外部に 172.30.254.203/49606 という RTP IP アドレス / ポート ペアと 172.30.254.203/49607 という RTCP IP アドレス / ポート ペアがあり、ローカルに 10.130.56.3/49606 という RTP IP アドレス / ポート ペアと 49607 という RTCP ポートを持っています。

H.323 RAS セッションのモニタ

show h323-ras コマンドは、FWSM を越えてゲートキーパとその H.323 エンドポイントの間に確立された H.323 RAS セッションの情報を表示します。**debug h323 ras event** および **show local-host** コマンドとともに、このコマンドは、H.323 RAS インスペクション エンジンの問題のトラブルシューティングに使用されます。

show h323-ras コマンドは、H.323 RAS インスペクション エンジンの問題をトラブルシューティングするための接続情報を表示します。次に、**show h323-ras** コマンドの出力例を示します。

```
hostname# show h323-ras
Total: 1
        GK                      Caller
        172.30.254.214 10.130.56.14
```

この出力は、ゲートキーパ 172.30.254.214 とそのクライアント 10.130.56.14 の間にアクティブな登録が 1 つあることを示します。

HTTP 検査

ここでは、HTTP インスペクション エンジンの機能と、その設定を変更する手順について説明します。次の内容について説明します。

HTTP 検査の概要

Use the **inspect http** コマンドを使用して、HTTP 固有の攻撃や HTTP に対応付けられた他の脅威からネットワークを保護します。HTTP 検査では、次の機能を実行します。

- 拡張 HTTP 検査
- N2H2 または Websense を介した URL スクリーニング
- Java および ActiveX のフィルタリング

あとの 2 つの機能は、**filter** コマンドとともに設定します。フィルタリングの詳細については、第 20 章「アプリケーション レイヤ プロトコル検査の適用」を参照してください。

（注）　また、**no inspect http** コマンドは、**filter url** コマンドをディセーブルにします。

拡張 HTTP 検査機能は、アプリケーション ファイアウォールとして知られており、攻撃側がネットワーク セキュリティ ポリシーを回避するため HTTP メッセージを使用するのを防ぎます。この機能は次の HTTP メッセージすべてを確認します。

- RFC 2616 への適合
- RFC 定義方法のみを使用
- 表 20-5 のコマンドで定義された追加基準への準拠

inspect http コマンドで HTTP マップを指定する場合、拡張 HTTP 検査をイネーブルにします。拡張 HTTP インスペクション エンジンのパラメータは、HTTP マップによって定義されます。このマップは、**http-map** コマンドを使用して作成し、HTTP マップ コンフィギュレーション モードで利用可能なコマンドを使用して設定されます。

（注）　HTTP マップのある HTTP 検査をイネーブルにすると、アクションのある完全な HTTP 検査はリセットされ、デフォルトではログはイネーブルになります。検査失敗に応じて実行されるアクションを変更できますが、HTTP マップがイネーブルであるかぎり、完全な検査をディセーブルにできません。

拡張 HTTP 検査コマンド

表 20-5 に、拡張 HTTP 検査パラメータの設定に使用するコマンドを要約します。次のコマンドは、HTTP マップ コンフィギュレーション モードで利用できます。各コマンドは、コマンドによって実行されたパラメータに対して、メッセージが違反した場合に取るアクションを指定します。このアクションには、メッセージの許可、リセット メッセージの送信、メッセージの廃棄が含まれます。このアクションのほかに、イベントを記録するかどうか指定できます。

各コマンドの詳細な構文については、『*Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*』の対応するコマンド ページを参照してください。

表 20-5　HTTP マップ コンフィギュレーション コマンド

コマンド	説明
content-length	HTTP コンテンツの長さに基づいて、検査をイネーブルにします。
content-type-verification	HTTP コンテンツのタイプに基づいて、検査をイネーブルにします。
max-header-length	HTTP ヘッダーの長さに基づいて、検査をイネーブルにします。
max-uri-length	URI の長さに基づいて、検査をイネーブルにします。
port-misuse	アプリケーション ファイアウォール検査をイネーブルにします。
request-method	HTTP 要求方法に基づいて、検査をイネーブルにします。
strict-http	完全な HTTP 検査をイネーブルにします。
transfer-encoding	転送符号化タイプに基づいて、検査をイネーブルにします。

拡張 HTTP 検査のイネーブル化および設定

拡張 HTTP 検査をイネーブルにして設定する手順は、次のとおりです。

ステップ 1　FWSM の後ろの HTTP サーバが HTTP トラフィックを待ち受けるポートを決定します。デフォルトの ポートは TCP ポート 80 です。ただし、thwart 攻撃に対する簡単な手段として、代替ポートがしばしば使用されます。すべての HTTP トラフィックを検査対象とするには、TCP ポート 80 以外のポートの使用に関して、HTTP サーバを検証してください。

ステップ 2　HTTP トラフィックを識別するには、クラス マップを作成するか、または既存のクラス マップを変更します。**class-map** コマンドを次のように使用します。

```
hostname(config)# class-map class_map_name
hostname(config-cmap)#
```

class_map_name は、トラフィック クラスの名前です。**class-map** コマンドを入力すると、CLI はクラス マップ コンフィギュレーション モードを開始します。

ステップ 3　ステップ 1 で決定した HTTP ポートに送信されたトラフィックを識別します。そのためには、**match port** または **match access-list** コマンドを使用します。

連続しない複数のポートを特定する必要がある場合、**access-list extended** コマンドを使用してアクセス リストを作成し、各ポートと一致する ACE を追加してから、**match access-list** コマンドを使用します。次のコマンドは、アクセス リストを使用して、アクセス リストを持った複数の TCP ポートを特定します。

```
hostname(config)# access-list acl-name any any tcp eq port_number_1
hostname(config)# access-list acl-name any any tcp eq port_number_2
hostname(config)# class-map class_map_name
hostname(config-cmap)# match access-list acl-name
```

単一ポートを特定する必要がある場合、次のように **match port** コマンドを使用します。

```
hostname(config-cmap)# match port tcp port_number
```

port_number は、FWSM の後ろの HTTP サーバが待ち受ける TCP ポートのみです。

単一プロトコルの連続したポート範囲を特定する必要がある場合、次のように **range** キーワードを指定して **match port** コマンドを使用します。

```
hostname(config-cmap)# match port tcp range begin_port_number end_port_number
```

begin_port_number は、HTTP ポート範囲の最小ポートで、*end_port_number* は最大ポートです。

ステップ 4　（任意）拡張 HTTP 検査をイネーブルにする場合、次の手順を実行します。

a. HTTP 検査の追加パラメータを含む HTTP マップを作成します。**http-map** コマンドを次のように使用します。

```
hostname(config-cmap)# http-map map_name
hostname(config-http-map)#
```

map_name は、HTTP マップの名前です。CLI は、HTTP マップ コンフィギュレーション モードを開始します。

b. 拡張 HTTP 検査パラメータを設定します。そのためには、使用する拡張 HTTP コマンドを決定します。コマンド リストについては、表 20-5 を参照してください。

ステップ 5　HTTP インスペクション エンジンを HTTP トラフィックに適用するために使用するポリシー マップを作成するか、または既存のポリシー マップを変更します。そのためには、**policy-map** コマンドを次のように使用します。

```
hostname(config-cmap)# policy-map policy_map_name
hostname(config-pmap)#
```

policy_map_name は、ポリシー マップの名前です。CLI はポリシー マップ コンフィギュレーション モードを開始し、プロンプトがそれに応じて変わります。

ステップ 6　ステップ 2 で作成したクラス マップを指定します。このクラス マップは HTTP トラフィックを識別します。**class** コマンドを次のように使用します。

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

class_map_name は、ステップ 2 で作成したクラス マップの名前です。CLI はポリシー マップ クラス コンフィギュレーション モードを開始し、プロンプトがそれに応じて変わります。

ステップ 7　HTTP アプリケーション検査をイネーブルにします。そのためには、**inspect http** コマンドを次のように使用します。

```
hostname(config-pmap-c)# inspect http [map_name]
hostname(config-pmap-c)#
```

map_name は、ステップ 4 （任意）で作成した HTTP マップです。

ステップ 8　ポリシー マップをグローバルに、または特定のインターフェイスに適用するには、**service-policy** コマンドを次のように使用します。

```
hostname(config-pmap-c)# service-policy policy_map_name [global | interface
interface_ID]
hostname(config)#
```

policy_map_name は、ステップ 5 で設定したポリシー マップです。ポリシー マップをすべてのインターフェイス上のトラフィックに適用する場合、**global** オプションを使用します。ポリシー マップを特定のインターフェイス上のトラフィックに適用する場合、***interface*** *interface_ID* オプションを使用します。*interface_ID* は、**nameif** コマンドでインターフェイスに割り当てられた名前です。

FWSM は、指定のとおりに HTTP トラフィックの検査を開始します。

例 20-9　拡張 HTTP 検査のイネーブル化および設定

次に、アクセス リストを使用して、HTTP トラフィックを識別し、HTTP マップを定義し、ポリシーを定義し、そのポリシーを外部インターフェイスに適用する例を示します。

```
hostname(config)# class-map http_port
hostname(config-cmap)# match port tcp eq 80
hostname(config-cmap)# http-map sample_map
hostname(config-http-map)# content-length min 100 max 2000 action reset log
hostname(config-http-map)# content-type-verification match-req-rsp action reset log
hostname(config-http-map)# max-header-length request 100 action reset log
hostname(config-http-map)# max-uri-length 100 action reset log
hostname(config-http-map)# policy-map sample_policy
hostname(config-pmap)# class http_port
hostname(config-pmap-c)# inspect http sample_map
hostname(config-pmap-c)# service-policy sample_policy interface outside
hostname(config)#
```

ICMP 検査

ICMP 検査は、デフォルトではディセーブルです。

ICMP 検査の詳細については、『*Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*』の **inspect icmp** および **inspect icmp error** コマンド ページを参照してください。

ILS 検査

ILS 検査は、デフォルトではディセーブルです。

ILS 検査の詳細については、『*Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*』の **inspect ils** コマンド ページを参照してください。

MGCP 検査

ここでは、MGCP アプリケーション検査をイネーブルにして設定し、デフォルトのポート設定を変更する手順について説明します。次の内容について説明します。

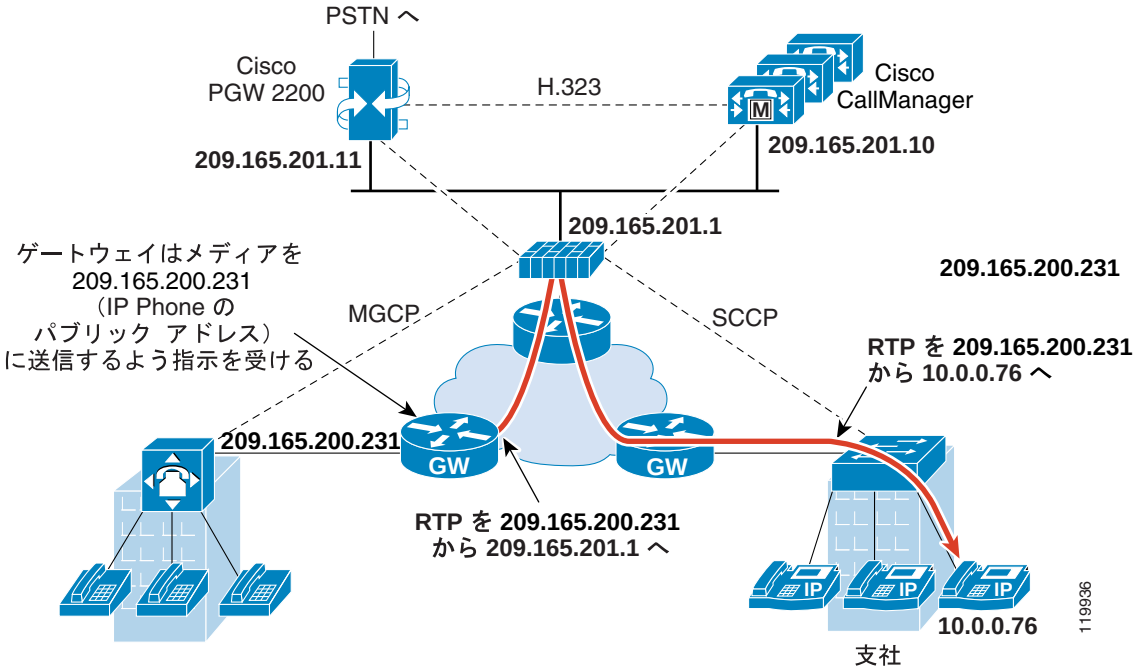
MGCP 検査の概要

MGCP は、メディア ゲートウェイ コントローラまたはコール エージェントと呼ばれる外部コール制御エレメントからメディア ゲートウェイを制御するために使用するマスター/ スレーブ プロトコルです。メディア ゲートウェイは、一般的に、電話回線上で伝送されるオーディオ信号と、インターネットまたは他のパケット ネットワーク上で伝送されるデータ パケット間の変換を行うネットワーク エレメントです。MGCP とともに NAT や PAT を使用すると、制限された外部（グローバル）アドレスを持った内部ネットワーク上で、多数のデバイスをサポートします。次に、メディア ゲートウェイの例を示します。

- トランキング ゲートウェイ。電話網と Voice over IP（VoIP）ネットワーク間のインターフェイスです。このゲートウェイは一般的に多数のデジタル回線を管理します。
- レジデンシャル ゲートウェイ。VoIP ネットワークに従来のアナログ（RJ11）インターフェイスを提供します。レジデンシャル ゲートウェイの例としては、ケーブル モデム / ケーブル セットトップ ボックス、xDSL デバイス、ブロードバンド ワイヤレス デバイスなどがあります。
- ビジネス ゲートウェイ。VoIP ネットワークに従来のデジタル PBX（構内交換機）インターフェイスまたは統合ソフト PBX インターフェイスを提供します。

MGCP メッセージは、UDP 上で転送されます。応答は、コマンドの送信元（IP アドレスおよび UDP ポート番号）に返送されますが、コマンドの宛先と同じアドレスから応答が戻されるとは限りません。たとえば、複数のコール エージェントがフェールオーバー設定に使用され、コマンドを受信したコール エージェントからバックアップ コール エージェントに制御が渡されたあとで、応答が戻される場合です。図 20-6 に、MGCP とともに NAT を使用する例を示します。

図 20-6　MGCP と NAT の使用



MGCP エンドポイントは、データ用の物理的または仮想の送信元 / 宛先です。メディア ゲートウェイには、コール エージェントが、他のマルチメディア エンドポイントとのメディア セッションを確立し制御するための接続を実行、変更、削除できるエンドポイントが含まれます。また、コール エージェントはエンドポイントに対し、所定のイベントを検出し、信号を生成するよう指示できます。エンドポイントは、サービス ステートの変更を自動的にコール エージェントに通知します。

MGCP トランザクションは、コマンドと必須応答で構成されています。次に、コマンドのタイプを示します。

- CreateConnection
- ModifyConnection
- DeleteConnection
- NotificationRequest
- Notify
- AuditEndpoint
- AuditConnection
- RestartInProgress

最初の 4 つのコマンドはコール エージェントによってゲートウェイに送信されます。Notify コマンドはゲートウェイによってコール エージェントに送信されます。ゲートウェイは、DeleteConnection コマンドも送信します。MGCP　ゲートウェイをコール　エージェントに登録することは、RestartInProgress コマンドを使用してできます。AuditEndpoint コマンドおよび AuditConnection コマンドは、コール エージェントによってゲートウェイに送信されます。

すべてのコマンドは Command ヘッダーで構成され、セッションの説明があとに続く場合があります。すべての応答は Response ヘッダーで構成され、セッションの説明があとに続く場合があります。

MGCP を使用するには、通常、次の 2 つのポートに送信されるトラフィックの検査を設定する必要があります。

- ゲートウェイがコール　エージェントからコマンドを受信するポート。ゲートウェイは通常、UDP ポート 2427 を待ち受けます。
- コール エージェントがゲートウェイからコマンドを受信するポート。コール エージェントは通常、UDP ポート 2727 を待ち受けます。

（注）　MGCP 検査は、MGCP シグナリングおよび RTP データの異なる IP アドレスの使用をサポートしません。共通の推奨する操作は、レジリエント IP アドレス（ループバックまたは仮想 IP アドレスなど）から RTP データを送信することです。ただし、FWSM は RTP データに対し、MGCP シグナリングと同じアドレスから着信するよう要求します。

MGCP コール エージェントおよびゲートウェイの設定

1 つまたは複数のゲートウェイを管理できるコール エージェントのグループを指定するには、**call-agent** コマンドを使用します。コール エージェント情報は、（ゲートウェイからのコマンド送信先と異なる）グループ内のコール エージェントで接続を開始する場合に使用されます。したがって、どのコール エージェントからでも応答を送信できます。同じ *group_id* を持つコール エージェントは、同じグループに属します。コール エージェントは複数のグループに所属できます。*group_id* オプションは 0 ～ 4,294,967,295 の数字です。*ip_address* オプションはコール エージェントの IP アドレスを指定します。

コール エージェント グループを指定するには、MGCP マップ コンフィギュレーション モードで **call-agent** コマンドを入力します。これは、グローバル コンフィギュレーション モードで **mgcp-map** コマンドを入力することで利用できます。

特定のゲートウェイを管理するコール エージェント グループを指定するには、**gateway** コマンドを入力します。ゲートウェイの IP アドレスは、*ip_address* オプションで指定します。*group_id* オプションに、0 ～ 4,294,967,295 の数値を指定します。この値は、ゲートウェイを管理しているコール エージェントの *group_id* と一致している必要があります。ゲートウェイは、1 つのグループにのみ、所属できます。

（注）　MGCP コール エージェントは、MGCP エンドポイントがあるかどうかを判別するため、AUEP メッセージを送信します。これにより FWSM 経由のフローが確立され、MGCP エンドポイントはコール エージェントに登録できます。

MGCP 検査の設定およびイネーブル化

MGCP アプリケーション検査をイネーブルにして設定する手順は、次のとおりです。

ステップ 1　MGCP トラフィックの受信に必要な次の 2 つのポートを特定する ACE を持つアクセス リストを定義します。標準ポートは、UDP ポート 2427 および 2727 です。アクセス リストを作成するには、次のように **access-list extended** コマンドを使用します。

```
hostname(config)# access-list acl-name permit udp any any eq port-1
hostname(config)# access-list acl-name permit udp any any eq port-2
```

acl-name はアクセス リストに割り当てられた名前です。*port-1* は最初の MGCP ポートで、*port-2* は 2 番めの MGCP ポートです。

ステップ 2　MGCP トラフィックを識別するには、クラス マップを作成するか、または既存のクラス マップを変更します。**class-map** コマンドを次のように使用します。

```
hostname(config)# class-map class_map_name
hostname(config-cmap)#
```

class_map_name は、トラフィック クラスの名前です。**class-map** コマンドを入力すると、CLI はクラス マップ コンフィギュレーション モードを開始します。

ステップ 3　**match access-list** コマンドを使用して、ステップ 1 で作成したアクセス リストで MGCP トラフィックを識別します。

```
hostname(config-cmap)# match access-list acl-name
```

ステップ 4　（任意）FWSM がピンホールをオープンする必要がある複数のコール エージェントおよびゲートウェイがネットワークにある場合、MGCP マップを作成します。そのためには、次の手順を実行します。

a. **mgcp-map** コマンドを使用して、MGCP マップを作成します。**mgcp-map** コマンドは、GCP 検査のパラメータを作成します。次のように、**mgcp-map** コマンドを使用します。

```
hostname(config-cmap)# mgcp-map map_name
hostname(config-mgcp-map)#
```

map_name は、MGCP マップの名前です。システムは MGCP マップ コンフィギュレーション モードを開始し、CLI プロンプトがそれに応じて変わります。

b. コール エージェントを設定します。そのためには、コール エージェントごとに **call-agent** コマンドを 1 回、次のように使用します。

```
hostname(config-mgcp-map)# call-agent ip_address group_id
```

c. ゲートウェイを設定します。そのためには、ゲートウェイごとに **gateway** コマンドを 1 回、次のように使用します。

```
hostname(config-mgcp-map)# gateway ip_address group_id
```

d. （任意）MGCP コマンド キューで許可されたコマンドの最大数を変更する場合、次のように **command-queue** コマンドを使用します。

```
hostname(config-mgcp-map)# command-queue command_limit
```

ステップ 5　MGCP インスペクション エンジンを MGCP トラフィックに適用するために使用するポリシー マップを作成するか、または既存のポリシー マップを変更します。そのためには、**policy-map** コマンドを次のように使用します。

```
hostname(config-cmap)# policy-map policy_map_name
hostname(config-pmap)#
```

policy_map_name は、ポリシー マップの名前です。CLI はポリシー マップ コンフィギュレーション モードを開始し、プロンプトがそれに応じて変わります。

ステップ 6　ステップ 2 で作成したクラス マップを指定します。このクラス マップは MGCP トラフィックを識別します。**class** コマンドを次のように使用します。

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

class_map_name は、ステップ 2 で作成したクラス マップの名前です。CLI はポリシー マップ クラス コンフィギュレーション モードを開始し、プロンプトがそれに応じて変わります。

ステップ 7　MGCP アプリケーション検査をイネーブルにします。そのためには、**inspect mgcp** コマンドを次のように使用します。

```
hostname(config-pmap-c)# inspect mgcp [map_name]
hostname(config-pmap-c)#
```

map_name は、ステップ 4（任意）で作成した MGCP マップです。

ステップ 8　ポリシー マップをグローバルに、または特定のインターフェイスに適用するには、**service-policy** コマンドを次のように使用します。

```
hostname(config-pmap-c)# service-policy policy_map_name [global | interface
interface_ID]
hostname(config)#
```

policy_map_name は、ステップ 5 で設定したポリシー マップです。ポリシー マップをすべてのインターフェイス上のトラフィックに適用する場合、**global** オプションを使用します。ポリシー マップを特定のインターフェイス上のトラフィックに適用する場合、***interface*** *interface_ID* オプションを使用します。*interface_ID* は、**nameif** コマンドでインターフェイスに割り当てられた名前です。

FWSM は、指定のとおりに MGCP トラフィックの検査を開始します。

例 20-10 に、MGCP トラフィックを識別し、MGCP マップを定義し、ポリシーを定義し、そのポリシーを外部インターフェイスに適用する例を示します。これは、デフォルト ポート（2427 および 2727）上の MGCP トラフィックと一致するクラス マップを作成します。この設定により、コール エージェント 10.10.11.5 および 10.10.11.6 がゲートウェイ 10.10.10.115 を制御し、コール エージェント 10.10.11.7 および 10.10.11.8 がゲートウェイ 10.10.10.116 および 10.10.10.117 の両方を制御できます。MGCP コマンド キューの最大数は 150 です。サービス ポリシーが外部インターフェイスに適用されます。

例 20-10　MGCP 検査のイネーブル化および設定

```
hostname(config)# access-list mgcp_acl permit udp any any eq 2427
hostname(config)# access-list mgcp_acl permit udp any any eq 2727
hostname(config)# class-map mgcp-traffic
hostname(config-cmap)# match access-list mgcp_acl
hostname(config-cmap)# mgcp-map sample_map
hostname(config-mgcp-map)# call-agent 10.10.11.5 101
hostname(config-mgcp-map)# call-agent 10.10.11.6 101
hostname(config-mgcp-map)# call-agent 10.10.11.7 102
hostname(config-mgcp-map)# call-agent 10.10.11.8 102
hostname(config-mgcp-map)# gateway 10.10.10.115 101
hostname(config-mgcp-map)# gateway 10.10.10.116 102
hostname(config-mgcp-map)# gateway 10.10.10.117 102
hostname(config-mgcp-map)# command-queue 150
hostname(config-mgcp-map)# policy-map sample_policy
hostname(config-pmap)# class mgcp_port
hostname(config-pmap-c)# inspect mgcp sample_map
hostname(config-pmap-c)# service-policy sample_policy interface outside
```

MGCP タイムアウト値の設定

timeout mgcp コマンドは、MGCP メディア接続が終了したあとの、非アクティビティ間隔を設定します。デフォルトは 5 分です。

timeout mgcp-pat コマンドは、PAT xlate のタイムアウトを設定します。MGCP にはキープアライブ メカニズムがないので、シスコ製ではない MGCP ゲートウェイ（コール エージェント）を使用する場合、30 秒のデフォルト タイムアウト間隔が終了すると PAT xlate は切断されます。

MGCP 検査の確認およびモニタ

show mgcp commands コマンドは、コマンド キュー内の MGCP コマンドの個数を表示します。**show mgcp sessions** コマンドは、既存の MGCP セッションの個数を表示します。**detail** オプションを指定すると、各コマンド（またはセッション）に関する追加情報が含まれます。次に、**show mgcp commands** コマンドの出力例を示します。

```
hostname# show mgcp commands
1 in use, 1 most used, 200 maximum allowed
CRCX, gateway IP: host-pc-2, transaction ID: 2052, idle: 0:00:07
```

次に、**show mgcp detail** コマンドの出力例を示します。

```
hostname# show mgcp commands detail
1 in use, 1 most used, 200 maximum allowed
CRCX, idle: 0:00:10
        Gateway IP      host-pc-2
        Transaction ID  2052
        Endpoint name   aaln/1
        Call ID         9876543210abcdef
        Connection ID
        Media IP        192.168.5.7
        Media port      6058
```

次に、**show mgcp sessions** コマンドの出力例を示します。

```
hostname# show mgcp sessions
1 in use, 1 most used
Gateway IP host-pc-2, connection ID 6789af54c9, active 0:00:11
```

次に、**show mgcp sessions detail** コマンドの出力例を示します。

```
hostname# show mgcp sessions detail
1 in use, 1 most used
Session active 0:00:14
        Gateway IP      host-pc-2
        Call ID         9876543210abcdef
        Connection ID   6789af54c9
        Endpoint name   aaln/1
        Media lcl port  6166
        Media rmt IP    192.168.5.7
        Media rmt port  6058
```

NetBIOS 検査

NetBIOS 検査はデフォルトではイネーブルです。

NetBIOS 検査の詳細については、『*Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*』の **inspect netbios** コマンド ページを参照してください。

PPTP 検査

PPTP 検査はデフォルトではディセーブルです。

PPTP 検査の詳細については、『*Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*』の **inspect pptp** コマンド ページを参照してください。

RSH 検査

RSH 検査はデフォルトではイネーブルです。

RSH 検査の詳細については、『*Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*』の **inspect rsh** コマンド ページを参照してください。

RTSP 検査

ここでは、RTSP アプリケーション検査をイネーブルにして、デフォルトのポート設定を変更する手順について説明します。次の内容について説明します。

RTSP 検査の概要

inspect rtsp コマンドを使用すると、RTSP アプリケーション検査を制御します。このコマンドは、ポリシー マップ クラス コンフィギュレーション モードで利用できます。このコマンドは、デフォルトではディセーブルです。**inspect rtsp** コマンドは、FWSM に RTSP パケットを通過させます。RTSP は、RealAudio、RealNetworks、Apple QuickTime 4、RealPlayer、Cisco IP/TV 接続によって使用されます。

（注）　Cisco IP/TV の場合、RTSP TCP ポート 554 および TCP 8554 を使用します。

RTSP アプリケーションは、well-known ポート 554 の TCP（まれに UDP）を制御チャネルとして使用します。FWSM は、RFC 2326 に基づき、TCP だけをサポートしています。TCP 制御チャネルは、クライアント上に設定されたトランスポート モードに応じて、オーディオ / ビデオ トラフィックの伝送に使用するデータ チャネルをネゴシエートするために使用されます。

サポートされる RDT トランスポートは、rtp/avp、rtp/avp/udp、x-real-rdt、x-real-rdt/udp、x-pn-tng/udp です。

FWSM は、SETUP 応答メッセージをステータス コード 200 によって解析します。FWSM よりも外部にあるサーバからの応答メッセージを着信させるには、サーバからの着信接続用にダイナミック チャネルをオープンする必要があります。応答メッセージを発信する場合は、FWSM でダイナミック チャネルをオープンする必要はありません。

RFC 2326 は、SETUP 応答にクライアント ポートとサーバ ポートを含めることを規定していないので、FWSM でステートを保持し、SETUP メッセージ内のクライアント ポートを記憶します。QuickTime では、SETUP メッセージにクライアント ポートが設定され、サーバはサーバ ポートでのみ応答します。

RTSP 検査は PAT またはデュアル NAT をサポートしません。FWSM では、RTSP メッセージが HTTP メッセージ内に隠されている場合、HTTP クローキングを認識できません。

RealPlayer の使用

RealPlayer を使用する場合は、トランスポート モードを正しく設定することが重要です。FWSM では、サーバからクライアントに、またはクライアントからサーバに、**access-list** コマンドを追加します。RealPlayer では、**Options** >**Preferences** >**Transport** >**RTSP Settings** の順にクリックして、トランスポート モードを変更します。

RealPlayer で TCP モードを使用する場合は、**Use TCP to Connect to Server** および **Attempt to use TCP for all content** のチェックボックスを選択します。FWSM でインスペクション エンジンを設定する必要はありません。

RealPlayer で UDP モードを使用する場合は、**Use TCP to Connect to Server** および **Attempt to use UDP for static content** のチェックボックスを選択します。マルチキャスト経由でライブ コンテンツは利用できません。FWSM で、**inspect rtsp** *port* コマンドを追加します。

制限事項および制約事項

RTSP 検査には、次の制限が適用されます。

- FWSM は、UDP 上のマルチキャスト RTSP または RTSP メッセージをサポートしません。
- PAT はサポートされません。
- FWSM では、RTSP メッセージが HTTP メッセージ内に隠されている場合に HTTP クローキングを認識する機能はありません。
- FWSM では、RTSP メッセージ上で NAT を実行できません。組み込み IP アドレスが、HTTP または RTSP メッセージの一部である Session Description Protocol（SDP）ファイル内に含まれているからです。パケットは分割されることがあります。FWSM は、分割されたパケットに対しては NAT を実行できません。
- Cisco IP/TV を使用する場合、FWSM がメッセージの SDP 部分で実行する NAT 数は、Content Manager 内のプログラム リスト数に比例します（各プログラム リストには最低 6 つの IP アドレスが組み込まれています）。
- Apple QuickTime 4 または RealPlayer については、NAT を設定できます。Cisco IP/TV は、Viewer および Content Manager が外部ネットワーク上にあり、サーバが内部ネットワーク上にある場合に限り、NAT をサポートします。

RTSP 検査のイネーブル化および設定

RTSP アプリケーション検査をイネーブルにして設定する手順は、次のとおりです。

ステップ 1　FWSM の後ろで RTSP SETUP メッセージを受信するポートを決定します。デフォルト ポートは、TCP ポート 554 および 8554 です。

ステップ 2　RTSP SETUP メッセージを識別するアクセス リストを作成します。**access-list extended** コマンドを使用して、次のように各ポートと一致する ACE を追加します。

```
hostname(config)# access-list acl-name any any tcp eq port_number
```

ヒント　1 つのポート上にのみ、または連続したポート範囲に RTSP SETUP メッセージを許可する場合、アクセス リストの作成を省略できます。ステップ 4 では、**match access-list** コマンドではなく、**match port** コマンドを使用します。

ステップ 3　RTSP トラフィックを識別するには、クラス マップを作成するか、または既存のクラス マップを変更します。**class-map** コマンドを次のように使用します。

```
hostname(config)# class-map class_map_name
hostname(config-cmap)#
```

class_map_name は、トラフィック クラスの名前です。**class-map** コマンドを入力すると、CLI はクラス マップ コンフィギュレーション モードを開始します。

ステップ 4　ステップ 1 で決定した RTSP ポートに送信されたトラフィックを識別します。そのためには、**match access-list** コマンドを次のように使用します。

```
hostname(config-cmap)# match access-list acl-name
```

ステップ 5　RTSP インスペクション エンジンを RTSP トラフィックに適用するために使用するポリシー マップを作成するか、または既存のポリシー マップを変更します。そのためには、**policy-map** コマンドを次のように使用します。

```
hostname(config-cmap)# policy-map policy_map_name
hostname(config-pmap)#
```

policy_map_name は、ポリシー マップの名前です。CLI はポリシー マップ コンフィギュレーション モードを開始し、プロンプトがそれに応じて変わります。

ステップ 6　ステップ 3 で作成したクラス マップを指定します。このクラス マップは RTSP トラフィックを識別します。**class** コマンドを次のように使用します。

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

class_map_name は、ステップ 2 で作成したクラス マップの名前です。CLI はポリシー マップ クラス コンフィギュレーション モードを開始し、プロンプトがそれに応じて変わります。

ステップ 7　RTSP アプリケーション検査をイネーブルにします。そのためには、**inspect rtsp** コマンドを次のように使用します。

```
hostname(config-pmap-c)# inspect rtsp
hostname(config-pmap-c)#
```

ステップ 8　ポリシー マップをグローバルに、または特定のインターフェイスに適用するには、**service-policy** コマンドを次のように使用します。

```
hostname(config-pmap-c)# service-policy policy_map_name [global | interface
interface_ID]
hostname(config)#
```

policy_map_name は、ステップ 5 で設定したポリシー マップです。ポリシー マップをすべてのインターフェイス上のトラフィックに適用する場合、**global** オプションを使用します。ポリシー マップを特定のインターフェイス上のトラフィックに適用する場合、***interface*** *interface_ID* オプションを使用します。*interface_ID* は、**nameif** コマンドでインターフェイスに割り当てられた名前です。

FWSM は、指定のとおりに RTSP トラフィックの検査を開始します。

例 20-11 に、デフォルト ポート（554 および 8554）上で RTSP インスペクション エンジンの RTSP トラフィックをイネーブルにする手順を示します。サービス ポリシーを外部インターフェイスに適用します。

例 20-11　RTSP 検査のイネーブル化および設定

```
hostname(config)# access-list rtsp_acl permit tcp any any eq 554
hostname(config)# access-list rtsp_acl permit tcp any any eq 8554
hostname(config)# class-map rtsp-traffic
hostname(config-cmap)# match access-list rtsp_acl
hostname(config-cmap)# policy-map sample_policy
hostname(config-pmap)# class rtsp_port
hostname(config-pmap-c)# inspect rtsp 554
hostname(config-pmap-c)# inspect rtsp 8554
hostname(config-pmap-c)# service-policy sample_policy interface outside
```

SIP 検査

ここでは、SIP アプリケーション検査をイネーブルにして、デフォルトのポート設定を変更する手順について説明します。次の内容について説明します。

SIP 検査の概要

Session Initiation Protocol（SIP）は、Internet Engineering Task Force（IETF）に定義されているように、コール処理セッションをイネーブルにします。特に、二者間オーディオ会議または「コール」に使用されます。SIP は、SDP と連携してコール シグナリングを処理します。SDP は、メディア ストリームのポートを指定します。SIP を使用すると、FWSM は、任意の SIP VoIP ゲートウェイと VoIP プロキシ サーバをサポートできます。SIP と SDP の定義は、次の RFC で定義されています。

- SIP：Session Initiation Protocol、RFC 2543
- SDP：Session Description Protocol、RFC 2327

FWSM 経由の SIP コールをサポートするには、メディア接続アドレス、メディア ポート、メディアの初期接続のシグナリング メッセージを検査する必要があります。SIP シグナリングが well-known 宛先ポート（UDP/TCP 50/60）に送信される間に、メディア ストリームはダイナミックに割り当てられたポートを使用します。また、SIP は、IP パケットのユーザ データ部分に IP アドレスを組み込みます。SIP 検査は、これらの組み込まれた IP アドレスに対して NAT を適用します。

SIP とともに PAT を使用する場合に、次の制限事項および制約事項が適用されます。

- リモート エンドポイントが、FWSM によって保護されたネットワーク上の SIP プロキシに登録しようとする場合、次の特殊な状態では登録できません。
  - PAT がリモート エンドポイントに設定されています。
  - SIP 登録サーバが、外部ネットワーク上にあります。
  - エンドポイントによってプロキシ サーバに送信された REGISTER メッセージ内の接続フィールドで、ポートが失われます。
- SIP デバイスが、SDP 部分に owner/creator フィールド（o=）の IP アドレスがあるパケットを送信する場合、o= フィールドの IP アドレスは正しく変換されないことがあります。この owner/creator フィールドの IP アドレスは、接続フィールド（c=）の IP アドレスとは異なります。これは、SIP プロトコルの制限事項によるもので、o= フィールドにポート値を提供しません。

SIP インスタント メッセージング

インスタント メッセージングは、ユーザの間でほぼリアルタイムでメッセージを転送することです。SIP は、Windows Messenger RTC Client バージョン 4.7.0105 のみを使用して、Windows XP 上でチャット機能をサポートします。MESSAGE/INFO 方法および 202 Accept 応答は、次の RFC で定義された IM をサポートするのに使用されます。

- Session Initiation Protocol（SIP）-Specific Event Notification、RFC 3265
- Session Initiation Protocol（SIP）Extension for Instant Messaging、RFC 3428

MESSAGE/INFO 要求は、登録 / 加入後いつでも行うことができます。たとえば、2 人のユーザが常時オンライン上にいても、数時間チャットできません。したがって、SIP インスペクション エンジンは、設定された SIP タイムアウト値に従って、タイムアウトするピンホールを開きます。この値は、Subscription 時間よりも 5 分以上長く設定する必要があります。Subscription 時間は、Contact Expires 値で定義され、一般的に 30 分です。

MESSAGE/INFO 要求は一般的に、ポート 5060 以外のダイナミックに割り当てられたポートを使用して送信されるので、SIP インスペクション エンジンを通過する必要があります。

（注）　現在、チャット機能のみサポートされます。ホワイトボード、ファイル転送、アプリケーション共有 はサポートされていません。RTC Client 5.0 はサポートされません。

SIP 検査は、SIP のテキスト ベースのメッセージで NAT を実行し、メッセージの SDP 部分のコンテンツの長さを再計算し、さらにパケット長とチェックサムを再計算します。また、SIP メッセージの SDP 部分に指定されたポートのメディア接続をダイナミックにオープンします。エンドポイントは、そのアドレス / ポート上で待ち受けるからです。

SIP 検査には、SIP ペイロードからの CALL_ID/FROM/TO インデックスを持ったデータベースがあります。このインデックスは、コール、送信元、宛先を識別します。このデータベースには、SDP メディア情報フィールドのメディア アドレスとメディア ポート、およびメディアのタイプが保管されます。1 つのセッションに複数のメディア アドレスとポートが存在することもあります。FWSM は、これらのメディア アドレス / ポートを使用して 2 つのエンドポイント間で RTP/RTCP 接続をオープンします。

初回のコール セットアップ（INVITE）メッセージには、well-known ポート 5060 を使用する必要があります。ただし、以降のメッセージには、このポート番号を含める必要はありません。SIP インスペクション エンジンは、シグナリング接続ピンホールをオープンし、これらの接続を SIP 接続としてマークします。これは、メッセージに SIP を適用し、NAT を実行するためです。

コールがセットアップされると、SIP セッションは、接続先エンドポイントからの応答メッセージにより、接続先エンドポイントが待ち受ける RTP ポートを示すメディア アドレスとメディア ポートを受信するまで、「一時的な」ステートになります。1 分以内に応答メッセージを受信しなかった場合、そのシグナリング接続は切断されます。

最終ハンドシェイクが完了すると、コール ステートがアクティブになり、BYE メッセージを受信するまで、シグナリング接続が持続されます。

内部エンドポイントから外部エンドポイントにコールを開始する場合には、内部エンドポイントからの INVITE メッセージに指定される内部エンドポイントのメディア アドレスおよびメディア ポートに RTP/RTCP UDP パケットが転送されるように、外部インターフェイスに対してメディア ホールがオープンされます。内部インターフェイスへの非送信要求 RTP/RTCP UDP パケットは、FWSM の設定で具体的に許可されている場合を除き、FWSM を通過しません。

IP アドレス プライバシー

IP アドレス プライバシーをイネーブルにすると、IP Phone コールまたはインスタント メッセージング セッションに参加している 2 つの SIP エンドポイントが、同じ内部ファイアウォール インターフェイスを使用して、外部ファイアウォール インターフェイス上の SIP プロキシ サーバに接続する場合、すべての SIP シグナリング メッセージは SIP プロキシ サーバを通過します。

SIP over TCP または UDP アプリケーション検査がイネーブルならば、IP アドレス プライバシーをイネーブルにできます。この機能は、デフォルトではディセーブルです。IP アドレス プライバシーがイネーブルの場合、FWSM は、着信 SIP トラフィックの TCP または UDP ペイロードに組み込まれた内部および外部ホスト IP アドレスを変換せず、IP アドレスの変換規則は無視されます。

SIP マップ コンフィギュレーション モードで **ip-address-privacy** コマンドを使用すると、この機能がイネーブルであるかどうか制御できます。

SIP 検査のイネーブル化および設定

SIP 検査はデフォルトではイネーブルです。

SIP 検査をイネーブルにするには、IP アドレス プライバシー機能をイネーブルにすることに関係なく、次の手順を実行します。

ステップ 1　FWSM の後ろの SIP サーバが SIP トラフィックを待ち受けるポートを決定します。デフォルト ポートは、TCP および UDP ポート 5060 です。

ステップ 2　SIP トラフィックを識別するには、クラス マップを作成するか、または既存のクラス マップを変更します。**class-map** コマンドを次のように使用します。

```
hostname(config)# class-map class_map_name
hostname(config-cmap)#
```

class_map_name は、トラフィック クラスの名前です。**class-map** コマンドを入力すると、CLI はクラス マップ コンフィギュレーション モードを開始します。

ステップ 3　ステップ 1 で決定した SIP ポートに送信されたトラフィックを識別します。そのためには、**match port** または **match access-list** コマンドを使用します。

UDP および TCP ポートを識別する、または連続しない 2 つ以上のポートを特定する必要がある場合、**access-list extended** コマンドを使用してアクセス リストを作成し、各ポートと一致する ACE を追加してから、**match access-list** コマンドを使用します。次のコマンドは、アクセス リストを使用して、アクセス リストを持った UDP および TCP ポートを識別します。

```
hostname(config)# access-list acl-name any any tcp eq port_number
hostname(config)# access-list acl-name any any udp eq port_number
hostname(config)# class-map class_map_name
hostname(config-cmap)# match access-list acl-name
```

単一プロトコルを使用して単一ポートを特定する必要がある場合、次のように **match port** コマンドを使用します。

```
hostname(config-cmap)# match port {tcp | udp} port_number
```

port_number は、FWSM の後ろの SIP サーバが待ち受けるポートのみです。

単一プロトコルの連続したポート範囲を特定する必要がある場合、次のように **range** キーワードを指定して **match port** コマンドを使用します。

```
hostname(config-cmap)# match port tcp range begin_port_number end_port_number
```

begin_port_number は SIP ポート範囲の最小ポートで、*end_port_number* は最大ポートです。

ステップ 4　（任意）IP アドレス プライバシーをイネーブルにする場合、次の手順を実行します。

a. SIP 検査のパラメータを含む SIP マップを作成します。**sip-map** コマンドを次のように使用します。

```
hostname(config-cmap)# sip-map map_name
hostname(config-sip-map)#
```

map_name は、SIP マップの名前です。CLI は、SIP マップ コンフィギュレーション コマンドを開始します。

b. 次のコマンドを入力して、SIP マップの設定を定義します。

```
hostname(config-sip-map)# ip-address-privacy
```

ステップ 5　SIP インスペクション エンジンを SIP トラフィックに適用するために使用するポリシー マップを作成するか、または既存のポリシー マップを変更します。そのためには、**policy-map** コマンドを次のように使用します。

```
hostname(config-cmap)# policy-map policy_map_name
hostname(config-pmap)#
```

policy_map_name は、ポリシー マップの名前です。CLI はポリシー マップ コンフィギュレーション モードを開始し、プロンプトがそれに応じて変わります。

ステップ 6　ステップ 2 で作成したクラス マップを指定します。このクラス マップは SIP トラフィックを識別します。**class** コマンドを次のように使用します。

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

class_map_name は、ステップ 2 で作成したクラス マップの名前です。CLI はポリシー マップ クラス コンフィギュレーション モードを開始し、プロンプトがそれに応じて変わります。

ステップ 7　SIP アプリケーション検査をイネーブルにします。そのためには、**inspect sip** コマンドを次のように使用します。

```
hostname(config-pmap-c)# inspect sip [map_name]
hostname(config-pmap-c)#
```

map_name は、ステップ 4 （任意）で作成した SIP マップです。

ステップ 8　ポリシー マップをグローバルに、または特定のインターフェイスに適用するには、**service-policy** コマンドを次のように使用します。

```
hostname(config-pmap-c)# service-policy policy_map_name [global | interface
interface_ID]
hostname(config)#
```

policy_map_name は、ステップ 5 で設定したポリシー マップです。ポリシー マップをすべてのインターフェイス上のトラフィックに適用する場合、**global** オプションを使用します。ポリシー マップを特定のインターフェイス上のトラフィックに適用する場合、***interface*** *interface_ID* オプションを使用します。*interface_ID* は、**nameif** コマンドでインターフェイスに割り当てられた名前です。

FWSM は、指定のとおりに SIP トラフィックの検査を開始します。

例 20-12 に示すように、SIP インスペクション エンジンをイネーブルにし、デフォルト ポート（5060）の SIP トラフィックと一致するクラス マップを作成します。それからサービス ポリシーを外部インターフェイスに適用します。

例 20-12　SIP アプリケーション検査のイネーブル化

```
hostname(config)# class-map sip_port
hostname(config-cmap)# match port tcp eq 5060
hostname(config-cmap)# sip-map sample_map
hostname(config-snmp-map)# ip-address-privacy
hostname(config-snmp-map)# policy-map sample_policy
hostname(config-pmap)# class sip_port
hostname(config-pmap-c)# inspect sip sample_map
hostname(config-pmap-c)# service-policy sample_policy interface outside
```

SIP タイムアウト値の設定

メディア接続は、接続が休止してから 2 分以内に切断されます。ただし、このタイムアウトの値は変更できるので、より短い（または長い）時間を設定できます。SIP 制御接続のタイムアウト値を設定するには、次のコマンドを使用します。

```
hostname(config)# timeout sip hh:mm:ss
```

このコマンドを使用して、SIP 制御接続を終了するまでののアイドル タイムアウトを設定します。

SIP メディア接続のタイムアウト値を設定するには、次のコマンドを使用します。

```
hostname(config)# timeout sip_media hh:mm:ss
```

このコマンドを使用して、SIP メディア接続を終了するまでのアイドル タイムアウトを設定します。

SIP 検査の確認およびモニタ

show sip コマンドは、SIP インスペクション エンジンの問題のトラブルシューティングに役立ち、**inspect protocol sip udp 5060** コマンドで説明します。**show timeout sip** コマンドは、指定されたプロトコルのタイムアウト値を表示します。

show sip コマンドは、FWSM を超えて確立された SIP セッションに関する情報を表示します。**debug sip** および **show local-host** コマンドとともに、このコマンドは、SIP インスペクション エンジンの問題のトラブルシューティングに使用されます。

（注）　**show sip** コマンドを入力する前に、**pager** コマンドを設定することを推奨します。多くの SIP セッション レコードが存在し、**pager** コマンドが設定されていない場合、**show sip** コマンドの出力が最後まで到達するには、しばらく時間がかかります。

次に、**show sip** コマンドの出力例を示します。

```
hostname# show sip
Total: 2
call-id c3943000-960ca-2e43-228f@10.130.56.44
    state Call init, idle 0:00:01
call-id c3943000-860ca-7e1f-11f7@10.130.56.45
    state Active, idle 0:00:06
```

この例では、FWSM 上に 2 つのアクティブ SIP セッションがあります（Total フィールドを参照）。各 call-id　はコールを示します。

最初のセッションは、call-id c3943000-960ca-2e43-228f@10.130.56.44 で、Call Init 状態にあります。これは、このセッションはまだコール セットアップ中であることを示しています。コール セットアップは、コールへの最後の応答が受信されるまでは完了しません。たとえば、発信者はすでに INVITE を送信して、100 Response を受信した可能性がありますが、200 OK はまだ受信していません。したがって、コール セットアップはまだ完了していません。1xx で始まっていない応答メッセージは最後の応答と考えられます。このセッションは、1 秒間アイドル状態でした。

2 番めのセッションは、Active ステートです。ここでは、コール セットアップは完了して、エンドポイントはメディアを交換しています。このセッションは、6 秒間アイドル状態でした。

Skinny（SCCP）検査

ここでは、SCCP アプリケーション検査をイネーブルにして、デフォルトのポート設定を変更する手順について説明します。次の内容について説明します。

SCCP 検査の概要

Skinny（SCCP）は、VoIP ネットワークで使用される簡易プロトコルです。SCCP を使用する Cisco IP Phone は、H.323 環境と併用できます。Cisco CallManager を使用することにより、SCCP クライアントと H.323 準拠端末を相互運用できます。FWSM におけるアプリケーション レイヤ機能では、SCCP Version 3.3 を認識します。SCCP プロトコルには、2.4、3.0.4、3.1.1、3.2、3.3.2 の 5 つのバージョンがあります。FWSM は、Version 3.3.2 までのすべてのバージョンをサポートします。

FWSM は、SCCP の PAT と NAT をサポートします。PAT は、使用する IP Phone のグローバル IP アドレスより IP Phone の数が多い場合に必要です。SCCP シグナリング パケットの NAT および PAT をサポートすることで、Skinny アプリケーション検査では、SCCP シグナリング パケットとメディア パケットのすべてが FWSM を通過することを保証します。

Cisco CallManager と Cisco IP Phone の間の通常のトラフィックは SCCP を使用し、特別な設定をすることなく SCCP 検査によって処理されます。FWSM は、DHCP オプション 150 および 66 もサポートしているので、Cisco IP Phone および他の DHCP クライアントに TFTP サーバの場所を送信できます。Cisco IP Phone は、要求に DHCP オプション 3 を含めることもあります。これは、デフォルト ルートを設定します。詳細については、「DHCP サーバで Cisco IP Phone を使用する方法」（p.8-33）を参照してください。

Cisco IP Phone のサポート

Cisco IP Phone と比較して Cisco CallManager の方がセキュリティの高いインターフェイス上に配置されているトポロジーでは、NAT が Cisco CallManager IP アドレスに必要な場合、マッピングは**スタティック**である必要があります。Cisco IP Phone では Cisco CallManager IP アドレスをその設定で明示的に指定する必要があるからです。スタティック アイデンティティ エントリにより、セキュリティの高いインターフェイス上の Cisco CallManager は Cisco IP Phone からの登録を受け入れることができます。Cisco IP Phone では、TFTP サーバにアクセスして、Cisco CallManager サーバに接続するのに必要な設定情報をダウンロードする必要があります。

TFTP サーバと比較して Cisco IP Phone の方がセキュリティの低いインターフェイス上にある場合は、アクセス リストを使用して UDP ポート 69 上の保護された TFTP サーバに接続する必要があります。TFTP サーバに対してはスタティック アイデンティティ エントリが必要ですが、アイデンティティ スタティック エントリにする必要はありません。NAT を使用する場合、スタティック アイデンティティ エントリは同じ IP アドレスにマッピングされます。PAT を使用する場合は、同じ IP アドレスとポートにマッピングされます。

TFTP サーバと Cisco CallManager と比較して Cisco IP Phone の方がセキュリティの*高い*インターフェイス上にある場合、Cisco IP Phone で接続を開始するためにアクセス リストもスタティック アイデンティティ エントリも必要ありません。

制限事項および制約事項

SCCP に対する現在のバージョンの PAT および NAT サポートに適用される制限は、次のとおりです。

- PAT は、**alias** コマンドを使用する設定とは連動しません。
- 外部 NAT または PAT も*サポートされません*。

内部 Cisco CallManager のアドレスが別のアドレスまたはポートに対して NAT または PAT 用に設定されている場合、外部 Cisco IP Phone の登録は失敗します。これは、FWSM が TFTP で転送されたファイル内容の NAT または PAT をサポートしていないからです。FWSM は TFTP メッセージの NAT をサポートし、TFTP ファイル用にピンホールを開きますが、FWSM は 電話の登録中に TFTP によって転送された Cisco IP Phone のコンフィギュレーション ファイルに埋め込まれた Cisco CallManager の IP アドレスおよびポートを変換できません。

（注）　FWSM では、コール セットアップ中であるコール以外の SCCP コールのステートフル フェールオーバーはサポートされていません。

SCCP 検査の設定およびイネーブル化

SCCP 検査はデフォルトではイネーブルです。

SCCP 検査をイネーブルにする、または SCCP トラフィックの受信に使用するデフォルト ポートを変更する手順は、次のとおりです。

ステップ 1　グローバル コンフィギュレーション モードで次のコマンドを入力して、トラフィック クラスの名前を指定します。

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

class_map_name に、次のようなトラフィック クラスの名前を指定します。

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

class-map コマンドを入力すると、CLI はクラス マップ コンフィギュレーション モードを開始し、プロンプトが変わります。次に例を示します。

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

ステップ 2　クラス マップ コンフィギュレーション モードで、**match** コマンドを定義します。次に例を示します。

```
hostname(config-cmap)# match port tcp eq 2000
hostname(config-cmap)# exit
hostname(config)#
```

連続したポート範囲を割り当てるには、**range** キーワードを入力します。次に例を示します。

```
hostname(config-cmap)# match port tcp range 2000-2010
```

連続しない複数のポートを SCCP 検査に割り当てるには、**access-list extended** コマンドを入力して、各ポートと一致するよう ACE を定義します。それから **match** コマンドを入力して、アクセス リストと SCCP トラフィック クラスを対応付けます。

ステップ 3　次のコマンドを入力して、ポリシー マップの名前を指定します。

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

policy_map_name にポリシー マップの名前を指定します。次に例を示します。

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

CLI はポリシー マップ コンフィギュレーション モードを開始し、それに応じてプロンプトが次のように変わります。

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

ステップ 4　次のコマンドを入力して、ステップ 1 で定義されたトラフィック クラスをポリシー マップに含めるよう指定します。

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

たとえば、次のコマンドを使用すると、sccp_port トラフィック クラスを現在のポリシー マップに割り当てます。

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

CLI はポリシー マップ クラス コンフィギュレーション モードを開始し、それに応じてプロンプトが次のように変わります。

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

ステップ 5　（任意）SCCP トラフィックを受信するため FWSM が使用するデフォルト ポートを変更するには、次のコマンドを入力します。

```
hostname(config-pmap-c)# inspect skinny
```

ステップ 6　次のコマンドを入力して、ポリシー マップ コンフィギュレーション モードに戻ります。

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

ステップ 7　次のコマンドを入力して、グローバル コンフィギュレーション モードに戻ります。

```
hostname(config-pmap)# exit
hostname(config)#
```

ステップ 8　次のコマンドを入力して、ポリシー マップをグローバルに、または特定のインターフェイスに適用します。

```
hostname(config)# service-policy policy_map_name [global | interface interface_ID
```

policy_map_name に ステップ 3 で設定したポリシー マップを指定し、**global** オプションを使用してすべてのインターフェイスを識別する、あるいは **nameif** コマンドで割り当てられた名前を使用して特定のインターフェイスを識別します。

たとえば、次のコマンドは sample_policy を外部インターフェイスに適用します。

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

次のコマンドは sample_policy をすべての FWSM インターフェイスに適用します。

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

例 20-13 に示すように、SCCP インスペクション エンジンをイネーブルにし、デフォルト ポート（2000）の SCCP トラフィックと一致するクラス マップを作成します。それからサービス ポリシーを外部インターフェイスに適用します。

例 20-13　SCCP アプリケーション検査のイネーブル化

```
hostname(config)# class-map sccp_port
hostname(config-cmap)# match port tcp eq 2000
hostname(config-cmap)# exit
hostname(config)# policy-map sample_policy
hostname(config-pmap)# class sccp_port
hostname(config-pmap-c)# inspect skinny
hostname(config-pmap-c)# exit
hostname(config)# service-policy sample_policy interface outside
```

SCCP 検査の確認およびモニタ

show skinny コマンドは、SCCP（Skinny）インスペクション エンジンの問題のトラブルシューティングに役立ちます。次に、以下の条件における **show skinny** コマンドの出力例を示します。FWSM を越えてセットアップされているアクティブな Skinny セッションが 2 つあります。最初のセッションは、ローカル アドレス 10.0.0.11 にある内部 Cisco IP Phone と 172.18.1.33 にある外部 Cisco CallManager の間に確立されたオーディオ接続です。TCP ポート 2000 は CallManager です。2 番めのセッションは、ローカル アドレス 10.0.0.22 にある別の内部 Cisco IP Phone と同じ Cisco CallManager の間に確立されたビデオ接続です。

```
hostname# show skinny
        LOCAL                   FOREIGN                 STATE
-----------------------------------------------------------------
1       10.0.0.11/52238         172.18.1.33/2000                1
  AUDIO 10.0.0.11/22948         172.18.1.22/20798
2       10.0.0.22/52232         172.18.1.33/2000                1
  VIDEO 10.0.0.22/20798         172.18.1.11/22948
```

この出力は、両方の内部 Cisco IP Phone の間でコールが確立されていることを示します。最初と 2 番めの電話の RTP リスニング ポートは、それぞれ UDP 22948 と 20798 です。

次に、これらの Skinny 接続に対する **show xlate debug** コマンドの出力例を示します。

```
hostname# show xlate debug
2 in use, 2 most used
Flags:  D - DNS, d - dump, I - identity, i - inside, n - no random,
        r - portmap, s - static
NAT from inside:10.0.0.11 to outside:172.18.1.11 flags si idle 0:00:16 timeout 0:05:00
NAT from inside:10.0.0.22 to outside:172.18.1.22 flags si idle 0:00:14 timeout 0:05:00
```

SMTP および拡張 SMTP 検査

ここでは、SMTP および ESMTP アプリケーション検査をイネーブルにして、デフォルトのポート設定を変更する手順について説明します。次の内容について説明します。

- SMTP および拡張 SMTP 検査の概要（p.20-67）
- SMTP および拡張 SMTP アプリケーション検査の設定およびイネーブル化（p.20-68）

SMTP および拡張 SMTP 検査の概要

FWSM は、SMTP および ESMTP のアプリケーション検査をサポートします。これらのプロトコルのアプリケーション検査は、FWSM を通過できる SMTP または ESMTP コマンドのタイプを制限したり、モニタ機能を追加することで、攻撃から保護します。

ESMTP は SMTP プロトコルの機能を強化したもので、SMTP と同様の機能を持ちます。便宜上、ここでは SMTP という用語を使用して、SMTP と ESMTP 両方を表します。ESMTP のアプリケーション検査プロセスには、SMTP セッションのサポートが含まれます。ESMTP セッションで使用するコマンドの多くは、SMTP セッションで使用するコマンドと同じですが、ESMTP セッションはかなり高速で、信頼性とセキュリティに関連したオプション（配信ステータス通知など）をより多く提供します。

inspect smtp コマンドは、7 つの RFC 821 コマンド（DATA、HELO、MAIL、NOOP、QUIT、RCPT、RSET）をサポートします。**inspect esmtp** コマンドは、この 7 つのコマンドをサポートし、拡張 SMTP コマンド（AUTH、HELP、EHLO、ETRN、SAML、SEND、SOML、VRFY）もサポートします。

他の SMTP または ESMTP コマンド、および ESMTP のプライベートな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 X に変換され、FWSM で保護された SMTP サーバによって拒否されます。この場合、「500 Command unknown: 'XXX'」というメッセージが表示されます。不完全なコマンドは廃棄されます。

SMTP アプリケーション検査は、**inspect smtp** コマンドでイネーブルになって、高速パス処理で実行されます。したがって、この検査は FWSM 上の 3 つのネットワーク プロセッサのいずれかで実行されます。ESMTP アプリケーション検査は、**inspect esmtp** コマンドでイネーブルになって、制御プレーン パス処理で実行されます。したがって、この検査は FWSM 上の 1 つの汎用プロセッサで実行されます。

（注）ポリシー マップに **inspect smtp** コマンドと **inspect esmtp** コマンドの両方が含まれる場合、ポリシー マップに記載された最初のコマンドのみが一致するトラフィックに適用されます。

検査では、「2」、「0」、「0」の文字を除き、サーバの SMTP バナーの文字をアスタリスクに変更します。Carriage Return（CR; 復帰）および Linefeed（LF; 改行）の文字は無視されます。

SMTP 検査がイネーブルで、次の規則が順守されていない場合、インタラクティブ SMTP に使用する Telnet セッションが中断することがあります。SMTP コマンドの長さは 4 文字以上で、CR および LF で終了する必要があります。また、次の応答を発行する前に、応答を待つ必要があります。

SMTP サーバは、応答コードの番号と任意の読み取り可能な文字列によって、クライアントの要求に応答します。SMTP アプリケーション検査は、ユーザが使用できるコマンドおよびサーバが戻すメッセージを制御して、削減します。SMTP 検査は、次の 3 つの主要なタスクを実行します。

- SMTP 要求を、7 つの基本的な 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。
- 監査追跡の生成 — メール アドレス内の無効文字が置換された場合、監査記録 108002 が生成されます。詳細については、RFC 821 を参照してください。

SMTP 検査は、コマンドと応答のシーケンスをモニタして、次の異常を検出します。

- 不完全なコマンド
- コマンドの不正な終了（<CR><LR> で終了していない）
- MAIL コマンドおよび RCPT コマンドには、メールの送信者および受信者が指定されています。不正な文字が含まれていないかどうか、メール アドレスがスキャンされます。パイプライン文字（|）は削除されます（空白スペースに変更されます）。「<」および「>」は、メール アドレスを定義している場合にのみ認められます（「>」の前に必ず「<」があることが前提です）。
- SMTP サーバによる予期しない移行
- 未知のコマンドがあると、FWSM 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、サーバからクライアントにエラー コードが戻されます。パケット内が変更されるので、TCP チェックサムが再計算または調整されます。
- TCP ストリームの編集
- コマンドのパイプライン化

SMTP および拡張 SMTP アプリケーション検査の設定およびイネーブル化

SMTP 検査はデフォルトではイネーブルです。

SMTP または拡張 SMTP 検査をイネーブルにする手順は、次のとおりです。

ステップ 1　FWSM の後ろの SMTP サーバが SMTP トラフィックを待ち受けるポートを決定します。デフォルト ポートは TCP ポート 25 ですが、他のポートを待ち受けるよう SMTP サーバを設定できます。

ステップ 2　SMTP トラフィックを識別するには、クラス マップを作成するか、または既存のクラス マップを変更します。**class-map** コマンドを次のように使用します。

```
hostname(config)# class-map class_map_name
hostname(config-cmap)#
```

class_map_name は、トラフィック クラスの名前です。**class-map** コマンドを入力すると、CLI はクラス マップ コンフィギュレーション モードを開始します。

ステップ 3　**match** コマンドを使用して、ステップ 1 で決定した SMTP ポートに送信されたトラフィックを識別します。

ポート マッパ プロセスが単一ポートを待ち受ける場合、**match port** コマンドを使用して、ポートに送信されたトラフィックを次のように識別できます。

```
hostname(config-cmap)# match port tcp eq port_number
```

port_number は、ポート マッパ プロセスが待ち受けるポートです。連続したポート範囲を割り当てる必要がある場合、**range** キーワードを使用します。次に例を示します。

```
hostname(config-cmap)# match port tcp range begin_port_number end_port_number
```

ヒント　連続しない複数のポートを識別する必要がある場合、**access-list extended** コマンドを入力して、各ポートと一致するよう ACE を定義します。それから、**match port** コマンドではなく、**match access-list** コマンドを使用して、アクセス リストと SMTP トラフィック クラスを対応付けます。

ステップ 4　SMTP インスペクション エンジンを SMTP トラフィックに適用するのに使用するポリシー マップを作成します。そのためには、**policy-map** コマンドを次のように使用します。

```
hostname(config-cmap)# policy-map policy_map_name
hostname(config-pmap)#
```

policy_map_name は、ポリシー マップの名前です。CLI はポリシー マップ コンフィギュレーション モードを開始し、プロンプトがそれに応じて変わります。

ステップ 5　ステップ 2 で作成したクラス マップを指定します。このクラス マップは SMTP トラフィックを識別します。**class** コマンドを次のように使用します。

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

class_map_name は、ステップ 2 で作成したクラス マップの名前です。CLI はポリシー マップ クラス コンフィギュレーション モードを開始し、プロンプトがそれに応じて変わります。

ステップ 6　次のうち、いずれかの作業を実行します。

a.　拡張 SMTP アプリケーション検査をイネーブルにするには、次のコマンドを入力します。

```
hostname(config-pmap-c)# inspect esmtp
```

b.　SMTP アプリケーション検査をイネーブルにするには、次のコマンドを入力します。

```
hostname(config-pmap-c)# inspect smtp
```

（注）　**inspect smtp** コマンドと **inspect esmtp** コマンドの違いについては、「SMTP および拡張 SMTP 検査の概要」（p.20-67）を参照してください。

ステップ 7　ポリシー マップをグローバルに、または特定のインターフェイスに適用するには、**service-policy** コマンドを次のように使用します。

```
hostname(config-pmap-c)# service-policy policy_map_name [global | interface
interface_ID]
hostname(config)#
```

policy_map_name は、ステップ 4 で設定したポリシー マップです。ポリシー マップをすべてのインターフェイス上のトラフィックに適用する場合、**global** オプションを使用します。ポリシー マップを特定のインターフェイス上のトラフィックに適用する場合、***interface*** *interface_ID* オプションを使用します。*interface_ID* は、**nameif** コマンドでインターフェイスに割り当てられた名前です。

FWSM は、指定のとおりに SMTP トラフィックの検査を開始します。

例 20-14　ESMTP 検査の設定およびイネーブル化

```
hostname(config)# class-map smtp_port
hostname(config-cmap)# match port tcp eq 25
hostname(config-cmap)# policy-map sample_policy
hostname(config-pmap)# class smtp_port
hostname(config-pmap-c)# inspect esmtp
hostname(config-pmap-c)# service-policy sample_policy interface outside
hostname(config)#
```

SNMP 検査

ここでは、SNMP アプリケーション検査をイネーブルにして、デフォルトのポート設定を変更する手順について説明します。次の内容について説明します。

- SNMP 検査の概要（p.20-70）
- SNMP アプリケーション検査のイネーブル化および設定（p.20-70）

SNMP 検査の概要

SNMP アプリケーション検査は、SNMP の特定のバージョンへの SNMP トラフィックを制限します。SNMP の初期のバージョンはセキュリティが低く、したがって、セキュリティ ポリシーによって所定の SNMP バージョンを拒否する必要があります。FWSM は、SNMP バージョン 1、2、2c、または 3 を拒否できます。SNMP マップ コンフィギュレーション モードで **deny version** コマンドを使用することにより、許可するバージョンを制御します。

SNMP アプリケーション検査のイネーブル化および設定

SNMP 検査のデフォルト設定を変更する手順は、次のとおりです。

ステップ 1　FWSM の後ろの ネットワーク デバイスが SNMP トラフィックを待ち受けるポートを決定します。デフォルト ポートは、TCP ポート 161 および 162 です。

ステップ 2　SNMP トラフィックを識別するには、クラス マップを作成するか、または既存のクラス マップを変更します。**class-map** コマンドを次のように使用します。

```
hostname(config)# class-map class_map_name
hostname(config-cmap)#
```

class_map_name は、トラフィック クラスの名前です。**class-map** コマンドを入力すると、CLI はクラス マップ コンフィギュレーション モードを開始します。

ステップ 3　**match** コマンドを使用して、ステップ 1 で決定した SNMP ポートに送信されたトラフィックを識別します。

連続したポート範囲を割り当てる必要がある場合、**range** キーワードを使用します。次に例を示します。

```
hostname(config-cmap)# match port tcp range begin_port_number end_port_number
```

begin_port_number は、SNMP ポート範囲の最小ポートで、*end_port_number* は最大ポートです。

ヒント　連続しない複数のポートを識別する必要がある場合、**access-list extended** コマンドを入力して、各ポートと一致するよう ACE を定義します。それから、**match port** コマンドではなく、**match access-list** コマンドを使用して、アクセス リストと SNMP トラフィック クラスを対応付けます。

ステップ 4　SNMP 検査のパラメータを含む SNMP マップを作成します。**snmp-map** コマンドを次のように使用します。

```
hostname(config-cmap)# snmp-map map_name
hostname(config-snmp-map)#
```

map_name は、SNMP マップの名前です。CLI は、SNMP マップ コンフィギュレーション コマンドを開始します。

ステップ 5　SNMP マップによって許可された SNMP のバージョンを指定します。そのためには、**deny version** コマンドを使用して、許可しないバージョンを次のように否定します。

```
hostname(config-snmp-map)# deny version version
hostname(config-snmp-map)#
```

version は、制限する SNMP バージョンです。*version* の有効な値は 1、2、2c、3 です。**deny version** コマンドを必要な回数入力できます。

ステップ 6　SNMP インスペクション エンジンを SNMP トラフィックに適用するために使用するポリシー マップを作成するか、または既存のポリシー マップを変更します。そのためには、**policy-map** コマンドを次のように使用します。

```
hostname(config-cmap)# policy-map policy_map_name
hostname(config-pmap)#
```

policy_map_name は、ポリシー マップの名前です。CLI はポリシー マップ コンフィギュレーション モードを開始し、プロンプトがそれに応じて変わります。

ステップ 7　ステップ 2 で作成したクラス マップを指定します。このクラス マップは SNMP トラフィックを識別します。**class** コマンドを次のように使用します。

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

class_map_name は、ステップ 2 で作成したクラス マップの名前です。CLI はポリシー マップ クラス コンフィギュレーション モードを開始し、プロンプトがそれに応じて変わります。

ステップ 8　SNMP アプリケーション検査をイネーブルにします。そのためには、**inspect snmp** コマンドを次のように使用します。

```
hostname(config-pmap-c)# inspect snmp snmp_map_name
hostname(config-pmap-c)#
```

snmp_map_name は、ステップ 4 で作成した SNMP マップです。

ステップ 9　ポリシー マップをグローバルに、または特定のインターフェイスに適用するには、**service-policy** コマンドを次のように使用します。

```
hostname(config-pmap-c)# service-policy policy_map_name [global | interface
interface_ID]
hostname(config)#
```

policy_map_name は、ステップ 6 で設定したポリシー マップです。ポリシー マップをすべてのインターフェイス上のトラフィックに適用する場合、**global** オプションを使用します。ポリシー マップを特定のインターフェイス上のトラフィックに適用する場合、***interface*** *interface_ID* オプションを使用します。*interface_ID* は、**nameif** コマンドでインターフェイスに割り当てられた名前です。

FWSM は、指定のとおりに SNMP トラフィックの検査を開始します。

例 20-15 は、外部インターフェイスから TCP ポート 161 および 162 に送信されたトラフィック上で SNMP アプリケーション検査をイネーブルにします。

例 20-15　SNMP アプリケーション検査の設定

```
hostname(config)# class-map snmp_port
hostname(config-cmap)# match port tcp range 161 162
hostname(config-cmap)# snmp-map sample_map
hostname(config-snmp-map)# deny version 1
hostname(config-snmp-map)# deny version 2
hostname(config-snmp-map)# policy-map sample_policy
hostname(config-pmap)# class snmp_port
hostname(config-pmap-c)# inspect snmp sample_map
hostname(config-pmap-c)# service-policy sample_policy interface outside
hostname(config)#
```

SQL*Net 検査

SQL*Net 検査はデフォルトではイネーブルです。

SQL*Net 検査の詳細については、『*Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*』の **inspect sqlnet** コマンド ページを参照してください。

Sun RPC 検査

ここでは、Sun RPC アプリケーション検査をイネーブルにして、デフォルトのポート設定を変更し、Sun RPC サービス テーブルを管理する手順について説明します。次の内容について説明します。

Sun RPC 検査の概要

Sun RPC アプリケーション検査をイネーブルにする、または FWSM が待ち受けるポートを変更するには、ポリシー マップ クラス コンフィギュレーション モードで **inspect sunrpc** コマンドを使用します。このモードは、ポリシー マップ コンフィギュレーション モード内で **class** コマンドを使用すると利用できます。コンフィギュレー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。

inspect sunrpc コマンドは、Sun RPC プロトコルのアプリケーション検査をイネーブルまたはディセーブルにします。Sun RPC は NFS および Network Information Service（NIS; ネットワーク情報サービス）によって使用されます。Sun RPC サービスは、任意のポート上で実行できます。クライアントからサーバ上の Sun RPC サーバにアクセスする場合には、サービスを実行しているポートを学習する必要があります。そのためには、well-known ポート 111 上のポート マッパ プロセス（通常は rpcbind）にクエリーを送信します。

クライアントからサービスの Sun RPC プログラム番号を送信すると、ポート マッパ プロセスはサービスのポート番号を戻します。クライアントは Sun RPC クエリーをサーバに送信し、ポート マッパ プロセスによって特定されたポートを指定します。サーバから応答が送信されると、FWSM はこのパケットを代行受信し、そのポート上で、TCP/UDP の両方の初期接続をオープンします。

（注）　Sun RPC ペイロード情報の NAT または PAT はサポートされません。

Sun RPC 検査のイネーブル化および設定

Sun RPC 検査はデフォルトではイネーブルです。

（注）　UDP 上で Sun RPC 検査をイネーブルにしたり設定したりするには、別のトラフィック クラスまたは新しいポリシー マップを定義する必要はありません。単に **inspect sunrpc** コマンドをポリシー マップに追加します。このポリシー マップのトラフィック クラスはデフォルトのトラフィック クラスによって定義されます。この設定の例を 例 20-17（p.20-75）に示します。

Sun RPC 検査をイネーブルにする、または TCP を使用して Sun RPC トラフィックの受信に使用するデフォルト ポートを変更する手順は、次のとおりです。

ステップ 1　ポート マッパ プロセスが待ち受けるポートを決定します。これはほとんどの場合、ポート 111 で、オペレーティング システムと実装状態に応じて異なります。

ステップ 2　Sun RP トラフィックを識別するには、クラス マップを作成するか、または既存のクラス マップを変更します。**class-map** コマンドを次のように使用します。

```
hostname(config)# class-map class_map_name
hostname(config-cmap)#
```

class_map_name は、トラフィック クラスの名前です。**class-map** コマンドを入力すると、CLI はクラス マップ コンフィギュレーション モードを開始します。

ステップ 3　**match** コマンドを使用して、ステップ 1 で決定したポートに送信されたトラフィックを識別します。

ポート マッパ プロセスが単一ポートを待ち受ける場合、**match port** コマンドを使用して、ポートに送信されたトラフィックを次のように識別できます。

```
hostname(config-cmap)# match port tcp eq port_number
```

port_number は、ポート マッパ プロセスが待ち受けるポートです。連続したポート範囲を割り当てる必要がある場合、**range** キーワードを使用します。次に例を示します。

```
hostname(config-cmap)# match port tcp range begin_port_number end_port_number
```

ヒント　連続しない複数のポートを識別する必要がある場合、**access-list extended** コマンドを入力して、各ポートと一致するよう ACE を定義します。それから、**match port** コマンドではなく、**match access-list** コマンドを使用して、アクセス リストと Sun RPC トラフィック クラスを対応付けます。

ステップ 4　Sun RPC インスペクション エンジンを Sun RPC トラフィックに適用するために使用するポリシー マップを作成するか、または既存のポリシー マップを変更します。そのためには、**policy-map** コマンドを次のように使用します。

```
hostname(config-cmap)# policy-map policy_map_name
hostname(config-pmap)#
```

policy_map_name は、ポリシー マップの名前です。CLI はポリシー マップ コンフィギュレーション モードを開始し、プロンプトがそれに応じて変わります。

ステップ 5　ステップ 2 で作成したクラス マップを指定します。このクラス マップは Sun RPC トラフィックを識別します。**class** コマンドを次のように使用します。

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

class_map_name は、ステップ 2 で作成したクラス マップの名前です。CLI はポリシー マップ クラス コンフィギュレーション モードを開始し、プロンプトがそれに応じて変わります。

ステップ 6　Sun RPC アプリケーション検査をイネーブルにします。そのためには、次のコマンドを入力します。

```
hostname(config-pmap-c)# inspect sunrpc
hostname(config-pmap-c)#
```

ステップ 7　ポリシー マップをグローバルに、または特定のインターフェイスに適用するには、**service-policy** コマンドを次のように使用します。

```
hostname(config-pmap-c)# service-policy policy_map_name [global | interface
interface_ID]
hostname(config)#
```

policy_map_name は、ステップ 4 で設定したポリシー マップです。ポリシー マップをすべてのインターフェイス上のトラフィックに適用する場合、**global** オプションを使用します。ポリシー マップを特定のインターフェイス上のトラフィックに適用する場合、***interface*** *interface_ID* オプションを使用します。*interface_ID* は、**nameif** コマンドでインターフェイスに割り当てられた名前です。

FWSM は、指定のとおりに Sun RPC トラフィックの検査を開始します。

例 20-16　TCP ベースの Sun RPC 検査のイネーブル化および設定

次に、外部インターフェイスから TCP ポート 111 に送信されたトラフィック上で Sun RPC アプリケーション検査をイネーブルにする例を示します。

```
hostname(config)# class-map sunrpc_port
hostname(config-cmap)# match port tcp eq 111
hostname(config-cmap)# policy-map sample_policy
hostname(config-pmap)# class sunrpc_port
hostname(config-pmap-c)# inspect sunrpc
hostname(config-pmap-c)# service-policy sample_policy interface outside
hostname(config)#
```

例 20-17 に、Sun RPC over UDP をイネーブルにする例を示します。これを行うには、アクションをデフォルト トラフィック クラスに適用するポリシー マップに、**inspect sunrpc** コマンドを追加します。

例 20-17　UDP ベースの Sun RPC 検査のイネーブル化および設定

```
hostname(config)# policy-map asa_global_fw_policy
hostname(config-pmap)# class inspection_default
hostname(config-pmap-c)# inspect sunrpc
hostname(config-pmap-c)#
```

Sun RPC サービスの管理

FWSM は、確立された Sun RPC セッションを制御するための Sun RPC サービス テーブルを保持します。Sun RPC サービス テーブルにエントリを作成するには、グローバル コンフィギュレーション モードで **sunrpc-server** コマンドを使用します。

sunrpc-server コマンドを使用して、FWSM が Sun RPC アプリケーション検査によってオープンしたピンホールを閉じるまでのタイムアウトを指定します。たとえば、IP アドレス 192.168.100.2 を持つ Sun RPC サーバに 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。

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

このコマンドは、Sun RPC アプリケーション検査によってオープンしたピンホールが 30 分後に閉じるよう指定します。この例では、Sun RPC サーバは TCP ポート 111 を使用する内部インターフェイス上にあります。UDP、異なるポート番号、またはポート範囲も指定できます。ポート範囲を指定するには、開始ポート番号と終了ポート番号の間にハイフンを使用して区切ります（例、111-113）。

サービス タイプは、特定のサービス タイプとサービスに使用するポート番号の間にマッピングを指定します。サービス タイプ（この例では 100003）を決定するには、Sun RPC サーバ マシン上の UNIX または Linux コマンドラインで **sunrpcinfo** コマンドを使用します。

Sun RPC コンフィギュレーションを消去するには、次のコマンドを入力します。

```
hostname(config)# clear configure sunrpc-server
```

これは、**sunrpc-server** コマンドを使用して、実行されたコンフィギュレーションを削除します。**sunrpc-server** コマンドを使用すると、指定されたタイムアウトでピンホールを作成できます。

アクティブ Sun RPC サービスを消去するには、次のコマンドを入力します。

```
hostname(config)# clear sunrpc-server active
```

Sun RPC アプリケーション検査はポート マッパ サービスへのサービス要求に基づいてトラフィックをイネーブルにしたので、これはオープンしたピンホールを消去します。

Sun RPC 検査の確認およびモニタ

ここでの出力例は、内部インターフェイス上で IP アドレス 192.168.100.2 を持つ Sun RPC サーバ、および外部インターフェイス上で IP アドレス 209.168.200.5 を持つ Sun RPC クライアントが対象です。

現在の Sun RPC 接続に関する情報を表示するには、**show conn** コマンドを入力します。次に、**show conn** コマンドの出力例を示します。

```
hostname# show conn
15 in use, 21 most used
UDP out 209.165.200.5:800 in 192.168.100.2:2049 idle 0:00:04 flags -
UDP out 209.165.200.5:714 in 192.168.100.2:111 idle 0:00:04 flags -
UDP out 209.165.200.5:712 in 192.168.100.2:647 idle 0:00:05 flags -
UDP out 192.168.100.2:0 in 209.165.200.5:714 idle 0:00:05 flags i
hostname(config)#
```

Sun RPC サービス テーブル コンフィギュレーションに関する情報を表示するには、**show running-config sunrpc-server** コマンドを入力します。次に、**show running-config sunrpc-server** コマンドの出力例を示します。

```
hostname(config)# show running-config sunrpc-server
sunrpc-server inside 192.168.100.2 255.255.255.255 service 100003 protocol UDP port
111 timeout 0:30:00
sunrpc-server inside 192.168.100.2 255.255.255.255 service 100005 protocol UDP port
111 timeout 0:30:00
```

この出力では、30 分のタイムアウト間隔が、内部インターフェイス上で IP アドレス 192.168.100.2 を持つ Sun RPC サーバの UDP ポート 111 に設定されたことを示します。

Sun RPC サービスのオープンした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、**show sunrpc-server active** コマンドの出力例を示します。

```
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上のクライアントまたはサーバの 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上のクライアントまたはサーバの IP アドレスを示します。

Sun RPC サーバ上で実行する Sun RPC サービスに関する情報を表示するには、Linux または UNIX サーバのコマンドラインから **rpcinfo -p** コマンドを入力します。次に、**rpcinfo -p** コマンドの出力例を示します。

```
sunrpcserver:~ # rpcinfo -p
program vers proto port
100000 2 tcp 111 portmapper
100000 2 udp 111 portmapper
100024 1 udp 632 status
100024 1 tcp 635 status
100003 2 udp 2049 nfs
100003 3 udp 2049 nfs
100003 2 tcp 2049 nfs
100003 3 tcp 2049 nfs
100021 1 udp 32771 nlockmgr
100021 3 udp 32771 nlockmgr
100021 4 udp 32771 nlockmgr
100021 1 tcp 32852 nlockmgr
100021 3 tcp 32852 nlockmgr
100021 4 tcp 32852 nlockmgr
100005 1 udp 647 mountd
100005 1 tcp 650 mountd
100005 2 udp 647 mountd
100005 2 tcp 650 mountd
100005 3 udp 647 mountd
100005 3 tcp 650 mountd
```

この出力では、ポート 647 は UDP 上で動作する mountd デーモンに相当します。mountd プロセスでは、一般的にポート 32780 がよく使用されますが、この例では TCP ポート 650 を使用します。

TFTP 検査

TFTP 検査はデフォルトではイネーブルです。

TFTP 検査の詳細については、『*Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*』の **inspect tftp** コマンド ページを参照してください。

XDMCP 検査

XDMCP 検査はデフォルトではイネーブルです。ただし、XDMCP インスペクション エンジンは **established** コマンドの設定が正しくないと動作しません。

XDMCP 検査の詳細については、『*Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*』の **established** および **inspect pptp** コマンド ページを参照してください。