CHAPTER 12

管理アプリケーション プロトコルのインスペクションの設定

この章では、アプリケーション レイヤ プロトコル インスペクションを設定する方法について説明します。インスペクション エンジンは、ユーザのデータ パケット内に IP アドレッシング情報を埋め込むサービスや、ダイナミックに割り当てられるポート上でセカンダリ チャネルを開くサービスに必要です。これらのプロトコルでは、高速パスでパケットを渡すのではなく、ASA で詳細なパケット インスペクションを行う必要があります。そのため、インスペクション エンジンがスループット全体に影響を与えることがあります。

ASA では、デフォルトでいくつかの一般的なインスペクション エンジンがイネーブルになっていますが、ネットワークによっては他のインスペクション エンジンをイネーブルにしなければならない場合があります。

この章は、次の項で構成されています。

DCERPC インスペクション

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

DCERPC の概要

DCERPC は、Microsoft 社の分散クライアント/サーバ アプリケーションで広く使われているプロトコルです。このプロトコルによって、ソフトウェア クライアントがサーバにあるプログラムをリモートで実行できるようになります。

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

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

（注） DCERPC インスペクションでは、ASA にピンホールを開くための EPM とクライアント間の通信だけがサポートされます。EPM を使用しない RPC 通信を使用するクライアントは、DCERPC インスペクションではサポートされません。

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

DCERPC インスペクションの追加のパラメータを指定するには、DCERPC インスペクション ポリシー マップを作成します。作成したインスペクション ポリシー マップは、DCERPC インスペクションをイネーブルにすると適用できます。

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

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

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

policy_map_name には、ポリシー マップの名前を指定します。CLI はポリシー マップ コンフィギュレーション モードに入ります。

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

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

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

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

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

b. DCERPC のピンホールのタイムアウトを設定して、グローバルなシステム ピンホールのタイムアウト（2 分）を上書きするには、次のコマンドを入力します。

```
hostname(config-pmap-p)# timeout pinhole hh:mm:ss
```

hh:mm:ss 引数には、ピンホール接続のタイムアウトを指定します。指定できる値は 0:0:1 ～ 1193:0:0 です。

c. エンドポイント マッパーのトラフィックのオプションを設定するには、次のコマンドを入力します。

```
hostname(config-pmap-p)# endpoint-mapper [epm-service-only] [lookup-operation
[timeout hh:mm:ss]]
```

hh:mm:ss 引数には、ルックアップ操作で生成されたピンホールのタイムアウトを指定します。ルックアップ操作にタイムアウトが設定されていない場合は、timeout pinhole コマンドで指定した値かデフォルトの値が使用されます。**epm-service-only** キーワードを指定すると、バインド中にエンドポイント マッパー サービスを実行し、このサービスのトラフィックだけが処理されるようにします。**lookup-operation** キーワードを指定すると、エンドポイント マッパー サービスのルックアップ操作をイネーブルにします。

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

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

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

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

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

GTP インスペクション

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

- 「GTP インスペクションの概要」（P.12-3）
- 「インスペクション制御を追加するための GTP インスペクション ポリシー マップの設定」（P.12-4）
- 「GTP インスペクションの確認とモニタリング」（P.12-8）

（注）　GTP インスペクションには、特別なライセンスが必要です。必要なライセンスがないときに、ASA で GTP 関連のコマンドを入力した場合、ASA はエラー メッセージを表示します。

GTP インスペクションの概要

GPRS は、モバイル ユーザに対して、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との間で中断しない接続を提供します。GGSN は、GPRS 無線データ ネットワークと他のネットワークとの間のインターフェイスです。SGSN は、モビリティ、データ セッション管理、およびデータ圧縮を実行します。

UMTS は、固定回線テレフォニー、モバイル、インターネット、コンピュータ テクノロジーの商用コンバージェンスです。UTRAN は、このシステムで無線ネットワークを実装するためのネットワーキング プロトコルです。GTP を使用すると、GGSN、SGSN、および UTRAN 間の UMTS/GPRS バックボーンで、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やユーザ データの暗号化は含まれていませんが、ASA で GTP を使用することによって、これらの危険性からネットワークを保護できます。

SGSN は、GTP を使用する GGSN に論理的に接続されます。GTP を使用すると、GSN 間の GPRS バックボーンで、マルチプロトコル パケットをトンネリングできます。GTP は、トンネル制御および管理プロトコルを提供します。このプロトコルによって、SGSN は、トンネルの作成、変更、および削除を行い、モバイル ステーションに GPRS ネットワーク アクセスを提供できます。GTP は、トンネリング メカニズムを使用して、ユーザ データ パケットを伝送するためのサービスを提供します。

（注） GTP をフェールオーバーと同時に使用しているとき、GTP 接続が確立され、データがトンネルを超えて伝送される前にアクティブ装置に障害が発生した場合、GTP データ接続（「j」フラグが設定されています）は、スタンバイ装置に複製されません。これは、アクティブ装置が初期接続をスタンバイ装置に複製しないためです。

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

GTP トラフィックに対して追加のパラメータを適用する場合、GTP マップを作成および設定します。**inspect gtp** コマンドにマップを指定しないと、ASA は、次のデフォルト値で事前に設定されたデフォルトの GTP マップを使用します。

- **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**

GTP マップを作成および設定するには、次の手順を実行します。作成した GTP マップは、「アプリケーション レイヤ プロトコル インスペクションの設定」（P.8-7）に従って GTP インスペクションをイネーブルにすると適用できます。

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

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

policy_map_name には、ポリシー マップの名前を指定します。CLI はポリシー マップ コンフィギュレーション モードに入ります。

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

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

ステップ 3　アクセス ポイント名を照合するには、次のコマンドを入力します。

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

ステップ 4　メッセージ ID を照合するには、次のコマンドを入力します。

```
hostname(config-pmap)# match [not] message id [message_id | range lower_range upper_range]
```

message_id には、英数字の ID（1 ～ 255）を指定します。*lower_range* は、メッセージ ID の範囲の下限です。*upper_range* は、メッセージ ID の範囲の上限です。

ステップ 5　メッセージ長を照合するには、次のコマンドを入力します。

```
hostname(config-pmap)# match [not] message length min min_length max max_length
```

min_length と *max_length* には、どちらも 1 ～ 65536 の値を指定します。このコマンドで指定する長さは、GTP ヘッダーとメッセージの残りの部分（UDP パケットのペイロード）の合計です。

ステップ 6　バージョンを照合するには、次のコマンドを入力します。

```
hostname(config-pmap)# match [not] version [version_id | range lower_range upper_range]
```

version_id には、0 ～ 255 を指定します。*lower_range* は、バージョン範囲の下限です。*upper_range* は、バージョン範囲の上限です。

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

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

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

mnc *network_code* 引数には、ネットワーク コードを表す 2 桁または 3 桁の値を指定します。

デフォルトでは、セキュリティ アプライアンスは、MCC と MNC の有効な組み合わせがあるかどうかチェックしません。このコマンドは、IMSI プレフィックス フィルタリングに使用されます。受信パケットの IMSI の MCC および MNC は、このコマンドで設定された MCC および MNC と比較され、一致しない場合はドロップされます。

このコマンドは、IMSI プレフィックス フィルタリングをイネーブルにするために使用する必要があります。複数のインスタンスを設定して許可する MCC と MNC の組み合わせを指定できます。デフォルトでは、ASA は MNC と MCC の組み合わせが有効であるかどうかをチェックしないため、設定した組み合わせが有効であるかどうかを確認する必要があります。MCC および MNC コードの詳細については、ITU E.212 勧告『*Identification Plan for Land Mobile Stations*』を参照してください。

b.　無効な GTP パケット、または、そのままでは解析で失敗してドロップされるパケットを許可するには、次のコマンドを入力します。

```
hostname(config-pmap-p)# permit errors
```

デフォルトでは、無効なパケットまたは解析中に失敗したパケットはすべてドロップされます。

c.　GSN プーリングのサポートをイネーブルにするには、**permit response** コマンドを使用します。

ASA が GTP インスペクションを実行する場合、デフォルトで ASA は、GTP 要求で指定されていない GSN からの GTP 応答をドロップします。これは、GSN のプール間でロードバランシングを使用して、GPRS の効率とスケーラビリティを高めているときに発生します。

GSN プーリングのサポートをイネーブルにするには、**permit response** コマンドを使用します。このコマンドは、GTP 要求がどの GSN に送信されたかにかかわらず、指定された GSN のセットの中のいずれかからの応答を許可するように ASA を設定します。ロードバランシング GSN のプールは、ネットワーク オブジェクトとして指定します。同様に、SGSN もネットワーク オブジェクトとして指定します。応答している GSN が GTP 要求の送信先の GSN と同じオブジェクト グループに属している場合、および応答している GSN による GTP 応答の送信が許可されている先のオブジェクト グループに SGSN がある場合、ASA はその応答を許可します。

d.　ロードバランシング GSN のプールを表すオブジェクトを作成するには、次の手順を実行します。

object-group コマンドを使用して、ロードバランシング GSN のプールを表す新しいネットワーク オブジェクト グループを定義します。

```
hostname(config)# object-group network GSN-pool-name
```

```
hostname(config-network)#
```

たとえば、次のコマンドは、gsnpool32 というオブジェクト グループを作成します。

```
hostname(config)# object-group network gsnpool32
hostname(config-network)#
```

e. **network-object** コマンドを使用して、ロードバランシング GSN を指定します。**host** キーワードを使用して、GSN ごとに 1 つの **network-object** コマンドを使用します。**network-object** コマンドを使用して、ロード バランシングを実行する GSN を含むネットワーク全体を指定することもできます。

```
hostname(config-network)# network-object host IP-address
```

たとえば、次のコマンドは、個別のホストを表す 3 つのネットワーク オブジェクトを作成します。

```
hostname(config-network)# network-object host 192.168.100.1
hostname(config-network)# network-object host 192.168.100.2
hostname(config-network)# network-object host 192.168.100.3
hostname(config-network)#
```

f. ロードバランシング GSN による応答が許可される SGSN を表すオブジェクトを作成するには、次の手順を実行します。

   a. **object-group** コマンドを使用して、GTP 要求を GSN プールに送信する SGSN を表す新しいネットワーク オブジェクト グループを定義します。

   ```
   hostname(config)# object-group network SGSN-name
   hostname(config-network)#
   ```

   たとえば、次のコマンドは、sgsn32 というオブジェクト グループを作成します。

   ```
   hostname(config)# object-group network sgsn32
   hostname(config-network)#
   ```

   b. **network-object** コマンドと **host** キーワードを使用して、SGSN を特定します。

   ```
   hostname(config-network)# network-object host IP-address
   ```

   たとえば、次のコマンドは、SGSN を表すネットワーク オブジェクトを作成します。

   ```
   hostname(config-network)# network-object host 192.168.50.100
   hostname(config-network)#
   ```

g. c.、d で定義した GSN プールを表すネットワーク オブジェクトの任意の GSN から、c.、f. で定義した SGSN を表すネットワーク オブジェクトへの GTP 応答を許可するには、次のコマンドを入力します。

```
hostname(config)# gtp-map map_name
hostname(config-gtp-map)# permit response to-object-group SGSN-name from-object-group
GSN-pool-name
```

たとえば、次のコマンドは、オブジェクト グループ gsnpool32 からオブジェクト グループ sgsn32 への GTP 応答を許可します。

```
hostname(config-gtp-map)# permit response to-object-group sgsn32 from-object-group
gsnpool32
```

次の例では、GSN プールと SGSN のネットワーク オブジェクトを定義して GSN プーリングをサポートする方法を示します。クラス C ネットワーク全体が GSN プールとして定義されていますが、ネットワーク全体を指定する代わりに、複数の個別の IP アドレスを **network-object** コマンドで 1 つずつ指定できます。この例では、次に、GSN プールから SGSN への応答を許可するように、GTP マップを変更します。

```
hostname(config)# object-group network gsnpool32
```

```
hostname(config-network)# network-object 192.168.100.0 255.255.255.0
hostname(config)# object-group network sgsn32
hostname(config-network)# network-object host 192.168.50.100
hostname(config)# gtp-map gtp-policy
hostname(config-gtp-map)# permit response to-object-group sgsn32 from-object-group
gsnpool32
```

h. キューで応答待ちができる GTP 要求の最大数を指定するには、次のコマンドを入力します。

```
hostname(config-gtp-map)# request-queue max_requests
```

max_requests 引数には、キューで応答待ちができる GTP 要求の最大数を 1 ～ 4294967295 で設定します。デフォルトは 200 です。

この上限に達した後に新しい要求が到着すると、最も長い時間キューに入っていた要求が削除されます。「Error Indication」、「Version Not Supported」および「SGSN Context Acknowledge」というメッセージは、要求と見なされないため、応答待ち要求のキューに入れられません。

i. GTP セッションの非アクティブ状態タイマーを変更するには、次のコマンドを入力します。

```
hostname(config-gtp-map)# timeout {gsn | pdp-context | request | signaling | tunnel}
hh:mm:ss
```

このコマンドは、タイムアウトごとに別々に入力します。

gsn キーワードで指定した時間、非アクティブ状態が続くと、GSN が削除されます。

pdp-context キーワードでは、PDP コンテキストの受信を開始するまでの最大許容時間を指定します。

request キーワードでは、GTP メッセージの受信を開始するまでの最大許容時間を指定します。

signaling キーワードで指定した時間、非アクティブ状態が続くと、GTP シグナリングが削除されます。

tunnel キーワードで指定した時間、非アクティブ状態が続くと、GTP トンネルが切断されます。

hh:*mm*:*ss* 引数にはタイムアウトを指定します。*hh* は時、*mm* は分、*ss* は秒です。値 **0** は、切断しないことを意味します。

j. ASA 上でアクティブな GTP トンネルの最大許容数を指定するには、次のコマンドを入力します。

```
hostname(config-gtp-map)# tunnel-limit max_tunnels
```

max_tunnels 引数には、トンネルの最大許容数を 1 ～ 4294967295 で指定します。デフォルトは 500 です。

このコマンドで指定したトンネル数に達すると、新しい要求はドロップされます。

次の例は、ネットワークのトンネル数を制限する方法を示しています。

```
hostname(config)# policy-map type inspect gtp gmap
hostname(config-pmap)# parameters
hostname(config-pmap-p)# tunnel-limit 3000

hostname(config)# policy-map global_policy
hostname(config-pmap)# class inspection_default
hostname(config-pmap-c)# inspect gtp gmap

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

GTP インスペクションの確認とモニタリング

GTP コンフィギュレーションを表示するには、特権 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、コマンド リファレンスのコマンドのページを参照してください。

show service-policy inspect gtp statistics コマンドを使用して、GTP インスペクションの統計情報を表示します。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
```

縦棒（|）を使用して、表示をフィルタリングできます。**?|** と入力すると、表示フィルタリング オプションが表示されます。

次に、**show service-policy inspect gtp statistics gsn** コマンドの GSN 出力例を示します。

```
hostname# show service-policy inspect gtp statistics gsn 9.9.9.9
1 in use, 1 most used, timeout 0:00:00

GTP GSN Statistics for 9.9.9.9, Idle 0:00:00, restart counter 0
    Tunnels Active 0Tunnels Created 0
    Tunnels Destroyed 0
    Total Messages Received 2
    Signaling Messages Data Messages
    total received 2 0
    dropped 0 0
    forwarded 2 0
```

show service-policy inspect gtp pdp-context コマンドを使用して、PDP コンテキストに関する情報を表示します。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.cisco.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、IMSI と NSAPI の値の組み合わせであるトンネル ID によって識別されます。GTP トンネルは、それぞれ別個の GSN ノードにある、2 つの関連する PDP コンテキストによって定義され、トンネル ID によって識別されます。GTP トンネルは、パケットを外部パケット データ ネットワークと MS ユーザの間で転送するために必要です。

次の例で示すように、縦棒（|）を使用して、表示をフィルタリングできます。

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

RADIUS アカウンティング インスペクション

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

RADIUS アカウンティング インスペクションの概要

よく知られている問題の 1 つに GPRS ネットワークでの過剰請求攻撃があります。過剰請求攻撃では、利用していないサービスについて料金を請求されるため、ユーザが怒りや不満を感じるおそれがあります。この場合、悪意のある攻撃者は、サーバへの接続をセットアップし、SGSN から IP アドレスを取得します。攻撃者がコールを終了しても、攻撃者のサーバはパケットの送信を続けます。このパケットは GGSN によってドロップされますが、サーバからの接続はアクティブなままです。攻撃者に割り当てられていた IP アドレスが解放され、正規ユーザに再割り当てされるので、正規ユーザは、攻撃者が利用するサービスの分まで請求されることになります。

RADIUS アカウンティング インスペクションは、GGSN へのトラフィックが正規のものかどうかを確認することにより、このような攻撃を防ぎます。RADIUS アカウンティングの機能を正しく設定しておくと、セキュリティ アプライアンスは、RADIUS アカウンティング要求の開始メッセージと終了メッセージに含まれる Framed IP 属性との照合結果に基づいて接続を切断します。終了メッセージの Framed IP 属性の IP アドレスが一致している場合、セキュリティ アプライアンスは、一致する IP アドレスを持つ送信元との接続をすべて検索します。

セキュリティ アプライアンスでメッセージを検証できるように、RADIUS サーバとの事前共有秘密キーを設定することもできます。事前共有秘密キーを設定しないと、セキュリティ アプライアンスは、メッセージの送信元を検証する必要がなく、その IP アドレスが、RADIUS メッセージの送信を許可されているアドレスの 1 つかどうかだけをチェックします。

（注）　GPRS をイネーブルにして RADIUS アカウンティング インスペクションを使用すると、ASA はアカウンティング要求の終了メッセージで 3GPP-Session-Stop-Indicator をチェックして、セカンダリ PDP コンテキストを正しく処理します。具体的には、ASA では、アカウンティング要求の終了メッセージがユーザ セッションおよび関連するすべての接続を終了する前に、メッセージに 3GPP-SGSN-Address 属性が含まれる必要があります。一部のサードパーティの GGSN は、この属性をデフォルトでは送信しない場合があります。

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

この機能を使用するには、**policy-map type management** で **radius-accounting-map** を指定してから、新しい **control-plane** キーワードを使用して service-policy に適用し、トラフィックが to-the-box インスペクションの対象であることを指定します。

次の例では、この機能を正しく設定するのに必要なコマンドをすべて使用しています。

ステップ 1　クラス マップとポートを設定します。

```
class-map type management c1
  match port udp eq 1888
```

ステップ 2　ポリシー マップを作成し、属性、ホスト、およびキー設定用の正しいモードにアクセスするための parameter コマンドを使用して、RADIUS アカウンティング インスペクションのパラメータを設定します。

```
policy-map type inspect radius-accounting radius_accounting_map
  parameters
    host 10.1.1.1 inside key 123456789
    send response
    enable gprs
    validate-attribute 22
```

ステップ 3　サービス ポリシーおよび control-plane キーワードを設定します。

```
policy-map type management global_policy
  class c1
     inspect radius-accounting radius_accounting_map

service-policy global_policy control-plane abc global
```

RSH インスペクション

RSH インスペクションはデフォルトでイネーブルになっています。RSH プロトコルは、TCP ポート 514 で RSH クライアントから RSH サーバへの TCP 接続を使用します。クライアントとサーバは、クライアントが STDERR 出力ストリームを受信する TCP ポート番号をネゴシエートします。RSH インスペクションは、必要に応じて、ネゴシエートされたポート番号の NAT をサポートします。

SNMP インスペクション

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

SNMP インスペクションの概要

SNMP アプリケーション インスペクションでは、SNMP トラフィックを特定のバージョンの SNMP に制限できます。以前のバージョンの SNMP は安全性が低いため、セキュリティ ポリシーを使用して特定の SNMP バージョンを拒否する必要が生じる場合もあります。ASA は、SNMP バージョン 1、2、2c、または 3 を拒否できます。許可するバージョンは、SNMP マップを作成して制御します。

作成した SNMP マップは、「アプリケーション レイヤ プロトコル インスペクションの設定」（P.8-7）に従って SNMP インスペクションをイネーブルにすると適用できます。

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

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

ステップ 1　SNMP マップを作成するには、次のコマンドを入力します。

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

map_name には、SNMP マップの名前を指定します。CLI は SNMP マップ コンフィギュレーション モードに入ります。

ステップ 2　拒否する SNMP のバージョンを指定するには、バージョンごとに次のコマンドを入力します。

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

version には、1、2、2c、3 のいずれかを指定します。

次の例では、SNMP バージョン 1 および 2 を拒否しています。

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

XDMCP インスペクション

XDMCP インスペクションはデフォルトでイネーブルになっていますが、XDMCP インスペクション エンジンは、**established** コマンドが適切に構成されていないと使用できません。

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

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

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

```
setenv DISPLAY Xserver:n
```

n はディスプレイ番号です。

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