CHAPTER 23

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 1000V でパケット 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 1000V 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。ASA 1000V は、セカンダリ接続に対して、必要に応じて、適切なポート番号とネットワーク アドレスを許可し、NAT も適用します。

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 の検査は、ASA 1000V 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
```

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、ASA 1000V 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、ASA 1000V 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。

ASA 1000V 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。共有秘密を設定しないと、ASA 1000V は、メッセージの送信元を検証する必要がなく、その送信元 IP アドレスが、RADIUS メッセージの送信を許可されている設定済みアドレスの 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 1000V 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 1000V 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 1000V 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.19-6）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 1000V 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 1000V 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 1000V 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。