CHAPTER 21

拡張接続機能の設定

この章では、接続機能をカスタマイズする手順について説明します。内容は次のとおりです。

接続制限とタイムアウトの設定

ここでは、TCP および UDP の最大接続数、最大接続レート、接続タイムアウトを設定し、TCP シーケンスのランダム化をディセーブルにする方法について説明します。

TCP 接続ごとに ISN を 2 つずつ使用します。1 つはクライアントが作成し、もう 1 つはサーバが作成します。FWSM は、着信方向と発信方向の両方で通過する TCP SYN の ISN をランダム化します。

保護対象のホストの ISN をランダム化することにより、攻撃者が新しい接続に使用される次の ISN を予測して新しいセッションをハイジャックするのを阻止します。

TCP 初期シーケンス番号のランダム化は、必要に応じてディセーブルにできます。次に例を示します。

- 別の直列接続されたファイアウォールでも初期シーケンス番号がランダム化され、トラフィックに影響することはないものの、両方のファイアウォールでこの動作を実行する必要がない場合。
- FWSM で eBGP マルチホップを使用しており、eBGP ピアで MD5 を使用している場合。ランダム化すると、MD5 チェックサムが破損します。
- FWSM で接続のシーケンス番号をランダム化しないようにする必要がある WAAS デバイスを使用する場合。

（注） TCP シーケンス番号のランダム化手法が実装されているため、xlate-bypass コマンドを使用する Xlate バイパスをイネーブルにする場合（「Xlate バイパスの設定」（P.16-20）を参照）、TCP シーケンスのランダム化をディセーブルにすることは制御接続だけで有効で、データ接続では無効です。データ接続に対して TCP シーケンスは継続してランダム化されます。

NAT コンフィギュレーションで最大接続数および TCP シーケンス ランダム化を設定することもできます。両方の方法を使用して同じトラフィックにこれらの設定を行う場合は、FWSM は低い方の制限を

使用します。TCP シーケンスのランダム化がいずれかの方法を使用してディセーブルになっている場合、FWSM は TCP シーケンスのランダム化をディセーブルにします。

NAT では、DoS 攻撃を防ぐ TCP 代行受信をトリガーする、初期接続制限も設定できます。接続制限、TCP シーケンスのランダム化、および初期接続制限を設定する場合は、第 16 章「ネットワーク アドレス変換（NAT）の設定」を参照してください。

接続制限とタイムアウトを設定する手順は、次のとおりです。

ステップ 1　トラフィックを特定するには、**class-map** コマンドを使用してクラス マップを追加します。詳細については、「トラフィックの識別（レイヤ 3/4 クラス マップ）」（P.20-4）を参照してください。

たとえば、次のコマンドを使用してすべてのトラフィックを照合できます。

```
hostname(config)# class-map CONNS
hostname(config-cmap)# match any
```

特定のトラフィックを照合する場合は、次のようにアクセス リストを照合できます。

```
hostname(config)# access list CONNS extended permit ip any 10.1.1.1 255.255.255.255
hostname(config)# class-map CONNS
hostname(config-cmap)# match access-list CONNS
```

（注）　3.x では、**set connection** コマンドをアクセス リスト（**match access-list**）に使用した場合に、接続設定が Access Control Entry（ACE; アクセス コントロール エントリ）ごとに適用されていましたが、4.0 では、接続設定がアクセス リスト全体に適用されるようになりました。

ステップ 2　クラス マップ トラフィックで行うアクションを設定するポリシー マップを追加または編集するには、次のコマンドを入力します。

```
hostname(config)# policy-map name
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

class_map_name はステップ 1 で追加したクラス マップです。

次に例を示します。

```
hostname(config)# policy-map CONNS
hostname(config-pmap)# class CONNS
hostname(config-pmap-c)#
```

ステップ 3　最大接続制限、接続レート制限、または TCP シーケンスのランダム化をイネーブルにするかどうかを設定するには、次のコマンドを入力します。

```
hostname(config-pmap-c)# set connection {[conn-max n] [conn-rate-limit n]
[random-sequence-number {enable | disable}]}
```

conn-max *n* 引数には、同時 TCP 接続または同時 UDP 接続（あるいはこの両方）の許容最大数を 0 ～ 65535 の範囲内で設定します。デフォルト値は 0 です。この場合は、接続数に制限はありません。

conn-rate-limit *n* 引数には、秒当たりの TCP 接続または UDP 接続（あるいはこの両方）の最大数を 0 ～ 65535 の範囲内で設定します。デフォルト値は 0 です。この場合は、接続レートに制限はありません。

random-sequence-number {**enable** | **disable**} キーワードで、TCP シーケンス番号のランダム化をイネーブルまたはディセーブルにします。

このコマンドを 1 行ですべて入力することも（順序は任意）、各属性を別々のコマンドとして入力することもできます。FWSM は、コマンドを実行コンフィギュレーション内で 1 行に結合します。

ステップ 4　TCP 初期接続（ハーフオープン）または TCP ハーフクローズ接続のタイムアウトを設定するには、次のコマンドを入力します。

```
hostname(config-pmap-c)# set connection timeout {[embryonic hh:mm:ss] [half-closed
hh:mm:ss]}
```

embryonic *hh*:*mm*:*ss* キーワード には、TCP 初期（ハーフオープン）接続が閉じられるまでのタイムアウト期間を 0:0:1 ～ 0:4:15 の間で設定します。デフォルトは 0:0:20 です。この値を 0 に設定することもでき、この場合は接続がタイムアウトしないことを意味します。

half-closed *hh*:*mm*:*ss* キーワードには、アイドル タイムアウトを 0:0:1 ～ 0:4:15 の間で設定します。デフォルトは 0:0:20 です。この値を 0 に設定することもでき、この場合は接続がタイムアウトしないことを意味します。FWSM は、ハーフクローズ接続を切断する場合にリセットを送信しません。

このコマンドを 1 行ですべて入力することも（順序は任意）、各属性を別々のコマンドとして入力することもできます。コマンドは実行コンフィギュレーションで 1 行に結合されます。

（注）　このコマンドは、インスペクション エンジンにより作成されるセカンダリ接続には影響を与えません。たとえば、**set connection timeout** コマンドを使用する SQL*Net、FTP データ フローなどのセカンダリ フローの接続設定は変更できません。これらの接続については、global **timeout conn** コマンドを使用して、アイドル時間を変更します。**timeout conn** コマンドは、該当トラフィックに **set connection timeout** コマンドを使用しない限り、すべてのトラフィック フローに影響を与えるので注意してください。

ステップ 5　すべてのプロトコルのアイドル接続のタイムアウトを設定するには、次のコマンドを入力します。

```
hostname(config-pmap-c)# set connection timeout idle hh:mm:0
```

idle *hh:mm*:**0** 引数には、確立されたプロトコル接続が閉じられるまでのアイドル時間を 0:5:0 ～ 1092:15:0 の間で設定します。デフォルト値は 0:60:0 です。値を 0 に設定することもできます。これは、接続がタイムアウトになることはないことを意味します。

（注）　このコマンドでは、秒に設定した値は無視されます。指定できるのは、時間と分だけです。そのため、秒は 0 に設定してください。

set connection timeout コマンドで **tcp** キーワードが **idle** キーワードに置換されましたが、**tcp** コマンド（TCP 接続限定）が設定に含まれている場合、このコマンドはまだ受け付けられます。**idle** コマンドと **tcp** コマンドが両方ともポリシーに含まれている場合は、TCP トラフィックが明示的に指定されているアクセス リストとクラス マップが一致している場合にかぎり、TCP トラフィックに関して **tcp** コマンドが優先されます。詳細については、『*Catalyst 6500 Series Switch and Cisco 7600 Series Router Firewall Services Module Command Reference*』の **set connection timeout** コマンドの説明を参照してください。

このコマンドは、インスペクション エンジンにより作成されるセカンダリ接続には影響を与えません。たとえば、**set connection timeout** コマンドを使用する SQL*Net、FTP データ フローなどのセカンダリ フローの接続設定は変更できません。これらの接続については、global **timeout conn** コマンドを使用して、アイドル時間を変更します。**timeout conn** コマンドは、該当トラフィックに **set connection timeout** コマンドを使用しない限り、すべてのトラフィック フローに影響を与えるので注意してください。

ステップ 6　1 つ以上のインターフェイスでポリシーマップをアクティブにするには、次のコマンドを入力します。

```
hostname(config)# service-policy policymap_name {global | interface interface_name}
```

policy_map_name は、ステップ 2 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は、**global** キーワードを使用します。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は、**interface** *interface_name* オプションを使用します。*interface_name* 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。

グローバル ポリシーは 1 つしか適用できません。インターフェイスのグローバル ポリシーは、そのインターフェイスにサービス ポリシーを適用することで上書きできます。各インターフェイスに適用できるポリシー マップは 1 つだけです。

次の例では、10.1.1.1 へのトラフィックに対して TCP 接続と UDP 接続の最大数を 5000、秒当たりの最大接続数を 500、最長初期接続タイムアウトを 40 秒、ハーフクローズ タイムアウトを 20 分、アイドル タイムアウトを 2 時間に設定します。

```
hostname(config)# access-list CONNS permit ip any host 10.1.1.1

hostname(config)# class-map conns
hostname(config-cmap)# match access-list CONNS

hostname(config-cmap)# policy-map conns
hostname(config-pmap)# class conns
hostname(config-pmap-c)# set connection conn-max 5000 conn-rate-limit 500
hostname(config-pmap-c)# set connection timeout embryonic 0:0:40 half-closed 0:20:0
hostname(config-pmap-c)# set connection timeout idle 2:0:0

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

複数のパラメータを指定して **set connection** コマンドを入力することも、各パラメータを個別のコマンドとして入力することもできます。FWSM は、実行コンフィギュレーションでコマンドを 1 つの行にまとめます。たとえば、次の 2 つのコマンドをクラス コンフィギュレーション モードで入力したとします。

```
hostname(config-pmap-c)# set connection timeout embryonic 0:0:40
hostname(config-pmap-c)# set connection timeout half-closed 0:20:0
```

show running-config policy-map コマンドの出力には、2 つのコマンドの結果が 1 つにまとめられたコマンドで表示されます。

```
set connection timeout embryonic 0:0:40 half-closed 0:20:0
```

PISA 統合でのアプリケーション タイプの許可または拒否

（注） この機能は Cisco IOS Release 12.2(18)ZYA 以降に依存しており、Catalyst 6500 スイッチでだけ使用できます。

スイッチのスーパーバイザ上で Programmable Intelligent Services Accelerator（PISA; プログラマブル インテリジェント サービス アクセラレータ）を使用すると、ディープ パケット検査を実行して特定フローのアプリケーション タイプを迅速に判別できます。この判別は、標準ポートがトラフィックに使用されていない場合でも実行できます。FWSM では、PISA カードの高性能ディープ パケット検査を利用して、アプリケーション タイプに基づいてトラフィックを許可または拒否することができます。コントロール プレーン パスを通過する FWSM の検査機能とは異なり、PISA でタグ付けされたトラフィックは FWSM アクセラレーション パスを通過できます。FWSM と PISA の統合には、PISA が導入されたアップストリーム スイッチを複数設定しなくても、単一の FWSM にセキュリティ設定を統合できるという別の利点もあります。

クリティカル アプリケーション タイプ用に帯域幅を確保する場合は、特定タイプのアプリケーション トラフィックを拒否することができます。たとえば、peer-to-peer（P2P; ピアツーピア）アプリケーションが他のクリティカル アプリケーションに影響を与えている場合に、P2P アプリケーションの使用を拒否できます。

ここでは、次の内容について説明します。

PISA 統合の概要

ここでは、PISA と FWSM との連携方法について説明します。内容は次のとおりです。

PISA 統合の注意事項と制限事項

PISA 統合に適用される注意事項と制限事項は次のとおりです。

- PISA と FWSM は同じスイッチ シャーシに配置できません。ただし、必要な場合は、FWSM のアップストリームとダウンストリームで複数の PISA を使用できます。
- FWSM へのパケットにタグを付ける必要があるため（セクション「GRE によるタギング」を参照）、FWSM に送出されるトラフィックに関して PISA のパフォーマンスにわずかな影響があります。
- FWSM のサービス ポリシーによって UDP パケットが拒否された場合に、対応するセッションがすぐに解除されません。代わりに、タイムアウトを設定し、その間に該当するセッションに到達したパケットを廃棄できます。
- FWSM と PISA 間で使用される特殊な GRE キーをエンド ユーザ アプリケーションで利用できます。この場合、PISA は syslog メッセージを生成し、これらのパケットを廃棄します。
- PISA では、アプリケーション タイプを判別するのに複数のパケットが必要となります。このため、PISA タギングが開始される前に、FWSM でセッションの確立が開始されます。PISA タギングが開始された時点で FWSM セキュリティ ポリシーが適用されるため、フローの拒否がポリシーに含まれている場合は、セッションを完了できません。
- フラグメント化されたパケットの場合、PISA は最初のフラグメントにタグを付け、FWSM はそのパケットを組み立て直してから、最初のフラグメントに含まれているカプセルに基づいてそのパケットを処理します。

「PISA の制限事項」（P.21-8）も参照してください。

GRE によるタギング

PISA は、特定のトラフィック フローで使用されているアプリケーションを判別すると、GRE を使用してすべてのパケットをカプセル化し、そのアプリケーション タイプを FWSM に報告するためのタグを挿入します。また、内部 / 元の IP ヘッダーとほとんど同じ外部 IP ヘッダー（レイヤ 4 プロトコルで

GRE が示されている点が異なる）も追加されます。元のレイヤ 2 ヘッダーはそのまま維持されます。これにより、パケットが変更された場合でも、元のルーティング / スイッチング パスが保持されます。GRE カプセル化では、32 バイト（外部 IP ヘッダー用の 20 バイトと GRE ヘッダー用の 12 バイト）が別途追加されます。

FWSM は、パケットを受信し、情報に従って処理を実行したあと、そのパケットから GRE カプセル化を解除します。

トラフィックが存在する VLAN に対して、PISA カプセル化に基づいてトラフィックを拒否するように FWSM を設定した場合は、PISA ですべてのトラフィック（拒否対象として指定されていないトラフィックを含む）がカプセル化されます。

GRE カプセル化を実行すると、パケットのサイズが少し拡張されるため、「より長いパケット長に対応するためのスイッチ上の MTU の変更」（P.21-8）に従って PISA と FWSM 間の MTU を大きくする必要があります。

GRE カプセル化では、FWSM に送出されるトラフィックに関して PISA のパフォーマンスにわずかな影響があります。

フェールオーバー サポート

PISA のフェールオーバーと FWSM のフェールオーバーの間に相互関係はありません。FWSM に対してステートフル フェールオーバーを実施している場合は、フェールオーバー間でセッション情報が維持されます。

PISA トラフィックを拒否するように FWSM を設定

PISA タギングを使用して拒否するトラフィックを特定する手順は、次のとおりです。

ステップ 1　アプリケーション タイプに基づいて拒否するトラフィックを特定するには、**class-map** コマンドを使用してクラス マップを追加します。詳細については、「トラフィックの識別（レイヤ 3/4 クラス マップ）」（P.20-4）を参照してください。

たとえば、次のようにアクセス リストを照合できます。

```
hostname(config)# access list BAD_APPS extended permit any 10.1.1.1 255.255.255.255
hostname(config)# class-map denied_apps
hostname(config-cmap)# match access-list BAD_APPS
```

ステップ 2　クラス マップ トラフィックで行うアクションを設定するポリシー マップを追加または編集するには、次のコマンドを入力します。

```
hostname(config)# policy-map name
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

class_map_name はステップ 1 で追加したクラス マップです。

次に例を示します。

```
hostname(config)# policy-map denied_apps_policy
hostname(config-pmap)# class denied_apps
hostname(config-pmap-c)#
```

ステップ 3　許可または拒否するアプリケーションを決定するには、次のコマンドを入力します。

```
hostname(config-pmap-c)# deny {all | protocol}
hostname(config-pmap-c)# permit protocol
```

protocol 引数はプロトコル名またはプロトコル番号になります。サポートされているプロトコル名を確認するには、**permit ?** または **deny ?** コマンドを使用します。

permit 文と **deny** 文を結合すると、拒否するトラフィックを絞り込むことができます。1 つ以上の **deny** 文を入力する必要があります。末尾に暗黙拒否が配置されるアクセス リストとは異なり、PISA アクションでは末尾に暗黙許可が配置されます。

たとえば、Skype、eDonkey、および Yahoo を除くすべてのトラフィックを許可するには、次のコマンドを入力します。

```
hostname(config-pmap-c)# deny skype
hostname(config-pmap-c)# deny yahoo
hostname(config-pmap-c)# deny eDonkey
```

次に、Kazaa と eDonkey を除くすべてのトラフィックを拒否する例を示します。

```
hostname(config-pmap-c)# deny all
hostname(config-pmap-c)# permit kazaa
hostname(config-pmap-c)# permit eDonkey
```

（注） **permit** コマンドと **deny** コマンドを含むクラス マップには、**inspect** コマンドを含めることができません。

ステップ 4　1 つまたは複数のインターフェイス上でポリシー マップを有効にするには、次のコマンドを入力します。

```
hostname(config)# service-policy policymap_name {global | interface interface_name}
```

ここで、**global** はポリシー マップをすべてのインターフェイスに適用し、**interface** は 1 つのインターフェイスに適用します。グローバル ポリシーは 1 つしか適用できません。インターフェイスのグローバル ポリシーは、そのインターフェイスにサービス ポリシーを適用することで上書きできます。各インターフェイスに適用できるポリシー マップは 1 つだけです。

次に、PISA 統合の設定例を示します。

```
hostname(config)# access-list BAD_APPS extended permit 10.1.1.0 255.255.255.0 10.2.1.0
255.255.255.0

hostname(config)# class-map denied_apps
hostname(config-cmap)# description "Apps to be blocked"
hostname(config-cmap)# match access-list BAD_APPS

hostname(config-cmap)# policy-map denied_apps_policy
hostname(config-pmap)# class denied_apps
hostname(config-pmap-c)# deny skype
hostname(config-pmap-c)# deny yahoo
hostname(config-pmap-c)# deny eDonkey

hostname(config-pmap-c)# service-policy denied_apps_policy inside
```

PISA と FWSM の統合用のスイッチの設定

ここでは、PISA と FWSM の統合用のスイッチを設定する手順について説明します。内容は次のとおりです。

PISA の制限事項

PISA に適用される制限事項は次のとおりです。

- Network Based Application Recognition（NBAR）はレイヤ 3 EtherChannel 上で動作しません。レイヤ 2 EtherChannel がサポートされています。
- PISA 上の RP では、プロトコル タギングがサポートされていません。このため、RP から FWSM へのパケットにはタグが付けられません。
- NBAR 実装では、IPv6 がサポートされていません。このため、プロトコル検出とプロトコル タギングは IPv4 にしか適用されません。NBAR によって課せられたこの制限に加えて、基本的な PISA インフラストラクチャでも IPv6 パケットのアクセラレーションがサポートされていません。
- 現在、レイヤ 2 ポート（トランクなど）上で PISA アクセラレートされた VLAN 用の L2 PISA 実装では、アクセラレートされたレイヤ 2 ポートを通過する VLAN 用の Switch Virtual Interface（SVI; スイッチ仮想インターフェイス）が稼動状態にならない（管理停止状態になる）という警告が出されています。
- 複数 VLAN アクセス ポートはサポートされていません。

「PISA 統合の注意事項と制限事項」（P.21-5）も参照してください。

より長いパケット長に対応するためのスイッチ上の MTU の変更

GRE カプセル化を実行する場合は、PISA と FWSM 間で使用される VLAN の MTU サイズを拡張する必要があります。GRE カプセル化では、32 バイト（外部 IP ヘッダー用の 20 バイトと GRE ヘッダー用の 12 バイト）が別途追加されます。

- ルーテッド スイッチ ポートまたはレイヤ 3 インターフェイス（SVI）の MTU を変更するには、次のコマンドを入力します。

```
Router(config-if)# mtu mtu_size
```

SVI の場合、*mtu_size* は 64 ～ 9216 バイトになります。ルーテッド スイッチ ポートの場合、*mtu_size* は 1500 ～ 9216 バイトになります。デフォルトの MTU サイズは 1500 バイトです。

- レイヤ 2 ポートのグローバル LAN ポート MTU サイズを設定するには、次のコマンドを入力します。

```
Router(config)# system jumbomtu mtu_size
```

mtu_size には、1500 ～ 9216 バイトを指定できます。デフォルトのサイズは 9216 バイトです。

PISA での分類の設定

- レイヤ 2 スイッチ ポート（物理ポート上に設定されたアクセス、トランク、または EtherChannel）またはレイヤ 3 インターフェイス（SVI、ルーテッド ポート、またはサブインターフェイス）で分類をイネーブルにするには、次のコマンドをインターフェイス コンフィギュレーション モードで入力します。

```
Router(config-if)# ip nbar protocol-discovery
```

- レイヤ 2 またはレイヤ 3 インターフェイスでプロトコル検出統計を表示するには、次のコマンドを入力します。

```
Router# show ip nbar protocol-discovery interface ifname
```

PISA でのタギングの設定

プロトコル検出をイネーブルにしたら、次に示された各コマンドを入力して出力パケット タギングをイネーブルにします。

（注） 分類とタギングは同一のポートでイネーブルにする必要があります。たとえば、アクセス ポートで分類をイネーブルにしてトランク ポートでタギングをイネーブルにすることはできません。

- スイッチ ポート（アクセス ポート）またはレイヤ 3 インターフェイス（SVI、ルーテッド ポート、またはサブインターフェイス）でタギングをイネーブルにするには、次のコマンドをインターフェイス コンフィギュレーション モードで入力します。

```
Router(config-if)# ip nbar protocol-tagging
```

- トランク ポートでタギングをイネーブルにするには、次のコマンドをインターフェイス コンフィギュレーション モードで入力します。

```
Router(config-if)# ip nbar protocol-tagging [vlan-list vlan-list]
```

vlan-list *vlan-list* 引数には、タグ付けする VLAN のリストを指定します。指定しないと、アクティブな VLAN がすべてタグ付けされます。

次の各コマンドは、PISA でのタギングのモニタに役立ちます。

- タギング設定情報を表示するには、次のコマンドを入力します。

```
Router# show ip nbar protocol-tagging {key | interface ifname | summary}
```

key キーワードは、タギングに使用される GRE キーを示します。

interface *ifname* 引数は、インターフェイスでタギングがイネーブルになっているかどうかを示します。

summary キーワードは、タギングがイネーブルになっているすべてのインターフェイスを示します。

- プロトコル名から ID へのマッピングを表示するには、次のコマンドを入力します。

```
Router# show ip nbar protocol-id [protocol_name]
```

protocol_name を入力すると、マッピングされた ID が表示されます。省略した場合は、プロトコル名と ID の完全なリストが表示されます。

- PISA でタグ付けされたパケットの数を表示するには、次のコマンドを入力します。

```
Router# show platform pisa np tx counters
```

次に例を示します。

```
Router# show platform pisa np tx counters

TX Statistics(ME1)
------------------
Errors:  0
……
```

```
TX NBAR Protocol tagged pkt:  9869
```

PISA 統合に対応したスイッチ設定の例

例 21-1　　レイヤ 3 モード（インターフェイスベース、ルーテッド ポート/SVI）

```
Router(config)# interface vlan 100
Router(config-if)# ip nbar protocol-discovery
! enables discovery
Router(config-if)# ip nbar protocol-tagging
! enables tagging
Router(config-if)# mtu 9216
! Allows packet sizes up to 9216 bytes without fragmenting
```

例 21-2　　レイヤ 2 モード（インターフェイスベース、アップリンク ポートでのプロトコル検出）

```
Router(config)# interface gigabitethernet 6/1
Router(config-if)# ip nbar protocol-discovery
! Classification
Router(config-if)# ip nbar protocol-tagging vlan-list 100
! Tagging
Router(config-if)# mtu 9216
! Allow packet size up to 9216 bytes without fragmenting
Router(config)# system jumbomtu 9216
! Set global LAN port MTU to 9216 bytes
```

PISA 接続のモニタリング

ここでは、次の内容について説明します。

廃棄された接続に関する Syslog メッセージ

PISA 接続が拒否された時点で Syslog メッセージ 302014（TCP の場合）および 302016（UDP の場合）が表示されます。次に例を示します。

```
%FWSM-6-302014: Teardown TCP connection 144547133155839947 for inside:10.1.1.12/33407 to
outside:209.165.201.10/21 duration 0:00:00 bytes 160 PISA denied protocol
```

FWSM での PISA 接続の表示

PISA からの接続をモニタするには、**show conn** コマンドを使用します。PISA でタグ付けされた接続が「p」フラグ付きの出力に一覧表示されます。次に、**show conn** コマンドの出力例を示します。

```
hostname# show conn
2 in use, 3 most used
 Network Processor 1 connections
TCP out 10.1.1.10:21 in 209.165.201.12:33406 idle 0:00:04 Bytes 1668 FLAGS - UOIp
 Network Processor 2 connections
UDP out 10.1.1.255:137 in 10.1.1.11:137 idle 0:00:48 Bytes 288 FLAGS -
Multicast sessions:
```

```
 Network Processor 1 connections
 Network Processor 2 connections
IPv6 connections:
...
```

TCP ステート バイパスの設定

ここでは、TCP ステート バイパスを設定する手順について説明します。内容は次のとおりです。

TCP ステート バイパスの概要

ここでは、TCP ステート バイパスの使用方法について説明します。内容は次のとおりです。

別々の FWSM を通過する発信フローと着信フローの許可

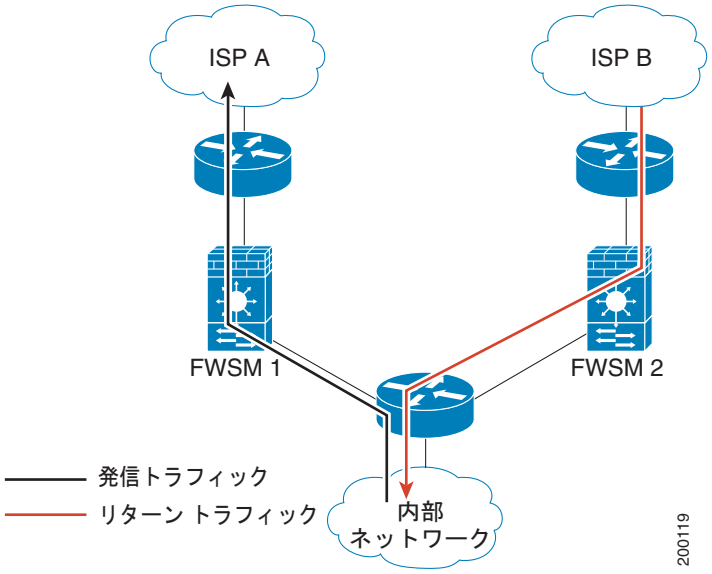
デフォルトで、FWSM を通過するすべてのトラフィックは、適応型セキュリティ アルゴリズムを使用して検査され、セキュリティ ポリシーに基づいて許可またはドロップされます。FWSM では、ファイアウォールの性能を最大限に活用するために、各パケットのステートをチェックし（新規の接続か確立済みの接続かを調べる）、セッション管理パス（新規の接続の SYN パケット）、アクセラレーション パス（確立済みの接続）、またはコントロール プレーン パス（拡張検査）のいずれかに割り当てます。ステートフル ファイアウォールの詳細については、「ステートフル インスペクションの概要」（P.1-8）を参照してください。

アクセラレーション パス内の既存の接続に一致した TCP パケットは、セキュリティ ポリシーのすべての面の再チェックを行わないで FWSM を通過できます。この機能によって、パフォーマンスが最大になります。ただし、SYN パケットを使用してアクセラレーション パスでセッションを確立する方法とアクセラレーション パスで実施されるチェック（TCP シーケンス番号など）は、非対称ルーティング ソリューションの障害となることがあります。接続の発信フローと着信フローはいずれも同じ FWSM を通過する必要があります。

たとえば、ある新しい接続が FWSM 1 に開始されるとします。SYN パケットはセッション管理パスを通過し、接続のエントリがアクセラレーション パス テーブルに追加されます。この接続の後続のパケットが FWSM 1 を通過する場合、これらのパケットはアクセラレーション パス テーブル内のエントリと照合され、一致した場合に通過できます。ただし、セッション管理パスを通過した SYN パケットが含まれていない FWSM 2 に後続のパケットが向かった場合、この接続のエントリがアクセラレー

ション パス テーブル内に存在しないため、これらのパケットはドロップされます。図 21-1 は非対称ルーティングの例を示したもので、アウトバウンド トラフィックはインバウンド トラフィックとは異なる FWSM を通過しています。

図 21-1　非対称ルーティング



アップストリーム ルータに非対称ルーティングが設定されており、トラフィックが 2 つの FWSM を通過することがある場合は、特定のトラフィックに対して TCP ステート バイパスを設定できます。TCP ステート バイパスにより、アクセラレーション パスでのセッションの確立方法が変更され、アクセラレーション パス チェックがディセーブルになります。この機能では、UDP 接続の処理と同様の方法で TCP トラフィックが処理されます。指定されたネットワークと一致した非 SYN パケットが FWSM に入った時点でアクセラレーション パス エントリが存在しない場合、アクセラレーション パスで接続を確立するために、そのパケットはセッション管理パスを通過します。一度アクセラレーション パスに入ったトラフィックでは、アクセラレーション パス チェックが回避されます。

サポートされていない機能

TCP ステート バイパスを使用する場合、次の機能はサポートされません。

- アプリケーション検査：アプリケーション検査では、着信および発信トラフィックの両方が同じ FWSM を通過する必要があるため、TCP ステート バイパスではアプリケーション検査はサポートされません。
- AAA 認証セッション：ユーザがある FWSM で認証される場合、他の FWSM 経由で戻るトラフィックは、その FWSM でユーザが認証されていないため、拒否されます。

NAT との互換性

変換セッションは各 FWSM に個別に確立されるため、TCP ステート バイパス トラフィックの両方の FWSM にスタティック NAT を設定する必要があります。ダイナミック NAT を使用する場合、FWSM 1 でセッションに選択されるアドレスは、FWSM 2 でセッションに選択されるアドレスとは異なります。

接続タイムアウト

特定の接続に 2 分間トラフィックがない場合、接続はタイムアウトします。このデフォルトは、**set connection timeout tcp** コマンドを使用して上書きできます。通常の TCP 接続は、デフォルトで 60 分後にタイムアウトします。

TCP ステート バイパスのイネーブル化

TCP ステート バイパスをイネーブルにする手順は、次のとおりです。

ステップ 1　ステートフル ファイアウォール検査をディセーブルにするトラフィックを特定するには、**class-map** コマンドを使用してクラス マップを追加します。詳細については、「トラフィックの識別（レイヤ 3/4 クラス マップ）」（P.20-4）を参照してください。

たとえば、次のようにアクセス リストを照合できます。

```
hostname(config)# access list bypass extended permit tcp any 10.1.1.1 255.255.255.255
hostname(config)# class-map bypass_traffic
hostname(config-cmap)# match access-list bypass
```

ステップ 2　クラス マップ トラフィックで行うアクションを設定するポリシー マップを追加または編集するには、次のコマンドを入力します。

```
hostname(config)# policy-map name
hostname(config-pmap)# class class_map_name
hostname(config-pmap-c)#
```

class_map_name はステップ 1 で追加したクラス マップです。

次に例を示します。

```
hostname(config)# policy-map tcp_bypass_policy
hostname(config-pmap)# class bypass_traffic
hostname(config-pmap-c)#
```

ステップ 3　TCP ステート バイパス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)# set connection advanced-options tcp-state-bypass
```

ステップ 4　1 つまたは複数のインターフェイス上でポリシー マップを有効にするには、次のコマンドを入力します。

```
hostname(config)# service-policy policymap_name {global | interface interface_name}
```

ここで、**global** はポリシー マップをすべてのインターフェイスに適用し、**interface** は 1 つのインターフェイスに適用します。グローバル ポリシーは 1 つしか適用できません。インターフェイスのグローバル ポリシーは、そのインターフェイスにサービス ポリシーを適用することで上書きできます。各インターフェイスに適用できるポリシー マップは 1 つだけです。

（注）　**show conn** コマンドを使用した場合、TCP ステート バイパスを使用する接続にはフラグ「b」が表示されます。

次に、TCP ステート バイパスのコンフィギュレーション例を示します。

```
hostname(config)# access-list tcp_bypass extended permit tcp 10.1.1.0 255.255.255.0
10.2.1.0 255.255.255.0
```

```
hostname(config)# class-map tcp_bypass
hostname(config-cmap)# description "TCP traffic that bypasses stateful firewall"
hostname(config-cmap)# match access-list tcp_bypass

hostname(config-cmap)# policy-map tcp_bypass_policy
hostname(config-pmap)# class tcp_bypass
hostname(config-pmap-c)# set connection advanced-options tcp-state-bypass

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

TCP 正規化のディセーブル化

レイヤ 7 検査を必要とするパケットや管理トラフィックなどのコントロール プレーン パスを通過するトラフィックの場合、FWSM は TCP 接続のキューに挿入できる不連続パケットの最大数を 2 に設定します。ユーザはこの値を設定できません。PIX プラットフォームと ASA プラットフォームでサポートされている他の TCP 正規化機能は FWSM に対してイネーブルになっていません。FWSM に限定された TCP 正規化のサポートをディセーブルにするには、**no control-point tcp-normalizer** コマンドを使用します。

IP スプーフィングの回避

Unicast Reverse Path Forwarding（uRPF; ユニキャスト RPF）をインターフェイス上でイネーブルにすることができます。ユニキャスト RPF は、ルーティング テーブルに従ってすべてのパケットの送信元 IP アドレスが正しい送信元インターフェイスに一致することを確認して、IP スプーフィング（パケットで不正な送信元 IP アドレスを使用して実際の送信元を伏せる行為）に対処します。

通常、FWSM はパケットの転送先を決定するときに宛先アドレスだけを参照します。ユニキャスト RPF は、FWSM に送信元アドレスも参照するように指示します。そのため、Reverse Path Forwarding と呼ばれています。FWSM を通過できるようにするトラフィックについて、FWSM のルーティング テーブルに、送信元アドレスに戻るルートを含める必要があります。詳細については、RFC 2267 を参照してください。

たとえば、外部トラフィックの場合、FWSM はデフォルト ルートを使用してユニキャスト RPF 保護を実現します。トラフィックが外部インターフェイスから入り、送信元アドレスがルーティング テーブルに認識されていない場合、FWSM はデフォルト ルートを使用してその外部インターフェイスを送信元インターフェイスとして正しく特定します。

ルーティング テーブルに認識されているものの、内部インターフェイスに関連付けられているアドレスからトラフィックが外部インターフェイスに入った場合、FWSM はパケットをドロップします。また、不明な送信元アドレスからトラフィックが内部インターフェイスに入った場合、一致するルート（デフォルト ルート）が外部インターフェイスを示しているため、FWSM はパケットをドロップします。

ユニキャスト RPF は、次のように実行されます。

- ICMP パケットにはセッションがないため、各パケットがチェックされます。
- UDP および TCP にはセッションがあるため、初期パケットには逆ルート検索が必要です。このセッション中に到着する以降のパケットは、セッションの一部として維持されている既存の状態を使用してチェックされます。非初期パケットは、初期パケットが使用したのと同じインターフェイス上に到着したことを確認するためにチェックされます。

Unicast RPF 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)# ip verify reverse-path interface interface_name
```

フラグメント サイズの設定

デフォルトでは、FWSM は、IP パケット当たりのフラグメントを 24 個まで許可し、リアセンブリを待機するフラグメントを 200 個まで許可します。NFS over UDP など、アプリケーションが日常的にパケットをフラグメント化する場合は、ネットワークでフラグメント化を許可する必要があります。ただし、トラフィックをフラグメント化するアプリケーションがない場合は、フラグメントが FWSM を通過できないようにすることをお勧めします。フラグメント化されたパケットは、DoS 攻撃によく使われます。フラグメントの禁止を設定するには、次のコマンドを入力します。

```
hostname(config)# fragment chain 1 [interface_name]
```

特定のインターフェイスでフラグメント化を禁止する場合は、インターフェイス名を入力します。デフォルトでは、このコマンドはすべてのインターフェイスに適用されます。

不正な接続のブロック

ホストがネットワークを攻撃しようとしていることがわかっている（たとえば、システム ログ メッセージに攻撃が表示されている）場合は、送信元 IP アドレスとその他の識別パラメータに基づいて接続をブロック（または排除）できます。排除を無効にするまで、新しい接続は作成できません。

（注） トラフィックをモニタする IPS がある場合、IPS は自動的に接続を排除します。

接続を手動で排除するには、次の手順を実行します。

ステップ 1　必要に応じて、次のコマンドを入力し、接続に関する情報を表示します。

```
hostname# show conn
```

FWSM は、各接続に関する情報を次のように表示します。

```
TCP out 64.101.68.161:4300 in 10.86.194.60:23 idle 0:00:00 bytes 1297 flags UIO
```

ステップ 2　この送信元 IP アドレスからの接続を排除するには、次のコマンドを入力します。

```
hostname(config)# shun src_ip [dst_ip src_port dest_port [protocol]] [vlan vlan_id]
```

このコマンドは既存の接続を廃棄するだけでなく、後続の接続をブロックします。デフォルトでは、プロトコルは IP を表す 0 です。

マルチ コンテキスト モードでは、このコマンドは管理コンテキストで入力できます。また、他のコンテキストのインターフェイスに割り当てられている VLAN ID を指定することで、他のコンテキストの接続を排除できます。

ステップ 3　排除を無効にするには、次のコマンドを入力します。

```
hostname(config)# no shun src_ip [vlan vlan_id]
```