接続設定

この章では、ASA を経由する接続用、または、ASA を宛先とする管理接続用の接続を設定する方法について説明します。

接続設定に関する情報

接続の設定は、ASA を経由する TCP フローなどのトラフィック接続の管理に関連するさまざまな機能で構成されます。一部の機能は、特定のサービスを提供するために設定する名前付きコンポーネントです。

接続の設定には、次が含まれています。

- **さまざまなプロトコルのグローバル タイムアウト**：すべてのグローバル タイムアウトにデフォルト値があるため、早期の接続の切断が発生した場合にのみグローバル タイムアウトを変更する必要があります。
- **トラフィック クラスごとの接続タイムアウト**：サービス ポリシーを使用して、特定のタイプのトラフィックのグローバル タイムアウトを上書きできます。すべてのトラフィック クラスのタイムアウトにデフォルト値があるため、それらの値を設定する必要はありません。
- **接続制限と TCP 代行受信**：デフォルトでは、ASA を経由する（または宛先とする）接続の数に制限はありません。サービス ポリシー ルールを使用して特定のトラフィック クラスに制限を設定することで、サービス妨害（DoS）攻撃からサーバーを保護できます。特に、初期接続（TCP ハンドシェイクを完了していない初期接続）に制限を設定できます。これにより、SYN フラッディング攻撃から保護されます。初期接続の制限を超えると、TCP 代行受信コンポーネントは、プロキシ接続に関与してその攻撃が抑制されていることを確認します。

- **Dead Connection Detection（DCD; デッド接続検出）**：アイドル タイムアウトの設定を超えたために接続が閉じられるように、頻繁にアイドル状態になっても有効な接続を維持する場合、Dead Connection Detection をイネーブルにして、アイドル状態でも有効な接続を識別してそれを維持することができます（接続のアイドル タイマーをリセットすることによって）。アイドル時間を超えるたびに、DCD は接続の両側にプローブを送信して、接続が有効であることを両側で合意しているかどうかを確認します。**show service-policy** コマンド出力には、DCDからのアクティビティ量を示すためのカウンタが含まれています。**show conn detail** コマンドを使用すると、発信側と受信側の情報およびプローブの送信頻度を取得できます。
- **TCP シーケンスのランダム化**：それぞれの TCP 接続には 2 つの ISN（初期シーケンス番号）が割り当てられており、そのうちの 1 つはクライアントで生成され、もう 1 つはサーバーで生成されます。デフォルトでは、ASA は、着信と発信の両方向で通過する TCP SNY の ISN をランダム化します。ランダム化により、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。ただし、TCP シーケンスのランダム化は、TCP SACK（選択的確認応答）を実質的に破棄します。クライアントが認識するシーケンス番号がサーバーが認識するものと異なるためです。必要に応じて、トラフィック クラスごとにランダム化をディセーブルにすることができます。
- **TCP 正規化**：TCP ノーマライザは、異常なパケットから保護します。一部のタイプのパケット異常をトラフィック クラスで処理する方法を設定できます。
- **TCPステート バイパス**：ネットワークで非対称ルーティングを使用するかどうかをチェックする TCP ステートをバイパスできます。
- **SCTP ステート バイパス**：SCTP プロトコル検証が必要なければ、Stream Control Transmission Protocol（SCTP）のステートフル インスペクションをバイパスできます。
- **フローのオフロード**：フローがNIC自体で切り替えられる超高速パスにオフロードされるトラフィックを識別して選択できます。オフロードによって、大容量ファイルの転送など、データ集約型アプリケーションのパフォーマンスを向上させることができます。
- **IPsec フローのオフロード**：IPsec サイト間 VPN またはリモートアクセス VPN セキュリティ アソシエーション（SA）の初期設定後、IPsec接続はデバイスのフィールドプログラマブルゲートアレイ（FPGA）にオフロードされるため、デバイスのパフォーマンスが向上します。この機能をサポートするプラットフォームでは、デフォルトで有効になっています。

接続の設定

接続制限、タイムアウト、TCP正規化、TCPシーケンスのランダム化、存続可能時間（TTL）のデクリメントには、ほとんどのネットワークに適切なデフォルト値があります。これらの接続の設定が必要となるのは、独自の要件があり、ネットワークに特定のタイプの設定がある場合、または早期のアイドル タイムアウトによる異常な接続切断が発生した場合のみです。

その他の接続関連機能は無効になっています。これらのサービスは、一般的なサービスとしてではなく、特定のトラフィック クラスにのみ設定します。これらの機能には次のものが含まれ

ています：TCP 代行受信、TCP ステート バイパス、Dead Connection Detection（DCD; デッド接続検出）、SCTP ステート バイパス、フロー オフロード。

次の一般的な手順では、考えられるすべての接続の設定について説明します。必要に応じて実装する設定を選んでください。

始める前に

パケットが デバイスに入ると、ファイアウォールは最初にアクセス コントロール ルールおよび NAT を評価して、接続エントリを作成する必要があるかどうかを判断します。接続エントリは、ネクスト ホップ用の ARP エントリが存在するかどうかに関係なく作成されます。このため、接続が存在しても、その接続範囲内のパケットがデバイスを通過することを意味するものではありません。アイドル タイムアウトなどの接続の設定は、不要な接続が残り、システム リソースを占有しないようにします。

手順

ステップ 1 グローバル タイムアウトの設定 （4 ページ）。これらの設定は、デバイスを通過するすべてのトラフィックに対してさまざまなプロトコルのデフォルトのアイドル タイムアウトを変更します。早期のタイムアウトによりリセットされる接続に問題がある場合は、まずグローバル タイムアウトを変更してください。

ステップ 2 SYN フラッド DoS 攻撃からのサーバーの保護（TCP 代行受信） （6 ページ）。この手順を使用して、TCP 代行受信を設定します。

ステップ 3 異常な TCP パケット処理のカスタマイズ（TCP マップ、TCP ノーマライザ） （9 ページ）（特定のトラフィック クラスについてデフォルトの TCP 正規化の動作を変更する場合）。

ステップ 4 非対称ルーティングの TCP ステートチェックのバイパス（TCP ステートバイパス） （14 ページ）（このタイプのルーティング環境がある場合）。

ステップ 5 TCP シーケンスのランダム化のディセーブル （17 ページ）（デフォルトのランダム化が特定の接続データをスクランブルしている場合）。

ステップ 6 大規模フローのオフロード （19 ページ）（コンピューティング集約型のデータセンターのパフォーマンスを改善する必要がある場合）。

ステップ 7 特定のトラフィック クラスの接続の設定（すべてのサービス） （28 ページ）。これは、接続の設定用の汎用手順です。これらの設定は、サービス ポリシー ルールを使用して、特定のトラフィック クラスのグローバルのデフォルト値を上書きできます。これらのルールを使用して、TCP ノーマライザのカスタマイズ、TCP シーケンスのランダム化の変更、パケットの存続可能時間のデクリメント、およびその他のオプション機能の実装も行います。

ステップ 8 TCP オプションの構成 （34 ページ）（他の標準的な TCP 動作をリセットまたは変更する必要がある場合）。

グローバル タイムアウトの設定

さまざまなプロトコルの接続スロットと変換スロットのグローバル アイドル タイムアウト期間を設定できます。指定したアイドル時間の間スロットが使用されなかった場合、リソースはフリー プールに戻されます。

グローバル タイムアウトを変更すると、サービス ポリシーによる特定のトラフィック フロー用に上書きできる新しいデフォルトのタイムアウトが設定されます。

特定のタイムアウト設定が構成されていないプロトコル（GRE など）では、アイドル タイムアウトは 2 分です。

手順

timeout コマンドを使用して、グローバル タイムアウトを設定します。

すべてのタイムアウト値の形式は *hh*:*mm*:*ss* で、最大期間はほとんどの場合 1193:0:0 です。すべてのタイムアウトをデフォルト値にリセットするには、**clear configure timeout** コマンドを使用します。単に 1 つのタイマーをデフォルトにリセットする場合は、その設定の **timeout** コマンドをデフォルト値とともに入力します。

タイマーをディセーブルにするには、値に **0** を使用します。

次のグローバル タイムアウトを構成できます。

- **timeout conn** *hh*:*mm*:*ss*：接続を閉じるまでのアイドル時間（0:5:0 ～ 1193:0:0）。デフォルトは 1 時間（1:0:0）です。
- **timeout half-closed** *hh*:*mm*:*ss*：TCP ハーフクローズ接続を閉じるまでのアイドル時間。FIN と FIN-ACK の両方が検出された場合、接続はハーフクローズ状態と見なされます。FIN のみが検出された場合は、通常の **conn** タイムアウトが適用されます。最小は 30 秒です。デフォルトは 10 分です。
- **timeout udp** *hh*:*mm*:*ss*：UDP 接続を閉じるまでのアイドル時間。この期間は 1 分以上にする必要があります。デフォルトは 2 分です。
- **timeout icmp** *hh*:*mm*:*ss*：ICMP のアイドル時間（0:0:2 ～ 1193:0:0）。デフォルトは 2 秒（0:0:2）です。
- **timeout icmp-error** *hh*:*mm*:*ss*：ASA が ICMP エコー応答パケットを受信してから ICMP 接続を削除するまでのアイドル時間で、0:0:0 から 0:1:0 の間、または **timeout icmp** 値のいずれか低い方です。デフォルトは 0（ディセーブル）です。このタイムアウトが無効で、ICMP インスペクションを有効にすると、ASA では、エコー応答が受信されるとすぐに ICMP 接続を削除します。したがってその（すでに閉じられた）接続用に生成されたすべての ICMP エラーは破棄されます。このタイムアウトは ICMP 接続の削除を遅らせるので、重要な ICMP エラーを受信できます。
- **timeout sunrpc** *hh*:*mm*:*ss*：SunRPC スロットが解放されるまでのアイドル時間。この期間は 1 分以上にする必要があります。デフォルトは 10 分です。

- **timeout H323** *hh*:*mm*:*ss*：H.245（TCP）および H.323（UDP）メディア接続を閉じるまでのアイドル時間（0:0:0 ～ 1193:0:0）。デフォルトは 5 分（0:5:0）です。H.245 と H.323 のいずれのメディア接続にも同じ接続フラグが設定されているため、H.245（TCP）接続は H.323（RTP および RTCP）メディア接続とアイドル タイムアウトを共有します。
- **timeout h225** *hh*:*mm*:*ss*：H.225 シグナリングリ接続を閉じるまでのアイドル時間。H.225 のデフォルト タイムアウトは 1 時間（1:0:0）です。すべての呼び出しがクリアされた後に接続をすぐにクローズするには、値を 1 秒（0:0:1）にすることを推奨します。
- **timeout mgcp** *hh*:*mm*:*ss*：MGCP メディア接続を削除するまでのアイドル時間（0:0:0 ～ 1193:0:0）。デフォルトは、5 分（0:5:0）です。
- **timeout mgcp-pat** *hh*:*mm*:*ss*：MGCP PAT 変換を削除するまでの絶対間隔（0:0:0 ～ 1193:0:0）。デフォルトは 5 分（0:5:0）です。最小時間は 30 秒です。
- **timeout sctp** *hh*:*mm*:*ss*：Stream Control Transmission Protocol（SCTP）接続を閉じるまでのアイドル時間（0:1:0 ～ 1193:0:0）。デフォルトは 2 分（0:2:0）です。
- **timeout sip** *hh*:*mm*:*ss*：SIP シグナリング ポート接続を閉じるまでのアイドル時間（0:5:0 ～ 1193:0:0）。デフォルトは、30 分（0:30:0）です。
- **timeout sip_media** *hh*:*mm*:*ss*：SIP メディア ポート接続を閉じるまでのアイドル時間。この期間は 1 分以上にする必要があります。デフォルトは 2 分です。SIP メディア タイマーは、SIP UDP メディア パケットを使用する SIP RTP/RTCP で、UDP 非アクティブ タイムアウトの代わりに使用されます。
- **timeout sip-provisional-media** *hh*:*mm*:*ss*：SIP 暫定メディア接続のタイムアウト値（0:1:0 ～ 0:30:0）。デフォルトは 2 分です。
- **timeout sip-invite** *hh*:*mm*:*ss*：暫定応答のピンホールとメディア xlate を閉じるまでのアイドル時間（0:1:0 ～ 00:30:0）。デフォルトは、3 分（0:3:0）です。
- **timeout sip-disconnect** *hh*:*mm*:*ss*：CANCEL メッセージまたは BYE メッセージで 200 OK を受信しなかった場合に、SIP セッションを削除するまでのアイドル時間（0:0:1 ～ 00:10:0）。デフォルトは 2 分（0:2:0）です。
- **timeout uath** *hh*:*mm*:*ss* {**absolute** | **inactivity**}：認証および認可キャッシュがタイムアウトし、ユーザーが次回接続時に再認証が必要となるまでの継続時間（0:0:0 ～ 1193:0:0）。デフォルトは 5 分（0:5:0）です。デフォルトのタイマーは **absolute** です。**inactivity** キーワードを入力すると、非アクティブになってから一定の期間後にタイムアウトが発生するように設定できます。uauth 継続時間は、xlate 継続時間より短く設定する必要があります。キャッシュをディセーブルにするには、0 に設定します。接続に受動 FTP を使用している場合、または Web 認証に virtual http コマンドを使用している場合は、0 を使用しないでください。
- **timeout xlate** *hh*:*mm*:*ss*：変換スロットが解放されるまでのアイドル時間。この期間は 1 分以上にする必要があります。デフォルトは 3 時間です。
- **timeout pat-xlate** *hh*:*mm*:*ss*：PAT 変換スロットが解放されるまでのアイドル時間（0:0:30 ～ 0:5:0）。デフォルトは 30 秒です。前の接続がアップストリーム デバイスで引き続き開い

ている可能性があるため、開放されたPATポートを使用する新しい接続をアップストリーム ルータが拒否する場合、このタイムアウトを増やすことができます。

- **timeout tcp-proxy-reassembly** *hh*:*mm*:*ss*：リアセンブリのためバッファ内で待機しているパケットをドロップするまでのアイドル タイムアウト（0:0:10 ～ 1193:0:0）。デフォルトは、1 分（0:1:0）です。
- **timeout floating-conn** *hh*:*mm*:*ss*：同じネットワークへの複数のルートが存在し、それぞれメトリックが異なる場合は、ASA は接続確立時点でメトリックが最良のルートを使用します。より適切なルートが使用可能になった場合は、このタイムアウトによって接続が閉じられるので、その適切なルートを使用して接続を再確立できます。デフォルトは 0 です（接続はタイムアウトしません）。より良いルートを使用できるようにするには、タイムアウト値を 0:0:30 ～ 1193:0:0 の間で設定します。
- **timeout conn-holddown** *hh*:*mm*:*ss*：接続で使用されているルートがもう存在していない、または非アクティブになったときに、システムが接続を保持する時間。このホールドダウン期間内にルートがアクティブにならない場合、接続は解放されます。接続ホールドダウンタイマーの目的は、ルートが発生してすぐにダウンする可能性がある場合に、ルート フラッピングの影響を減らすことです。ルートの収束がもっと早く発生するようにホールドダウン タイマーを減らすことができます。デフォルトは 15 秒です。指定できる範囲は 00:00:00 ～ 00:00:15 です。
- **timeout igp stale-route** *hh*:*mm*:*ss*：古いルートをルータの情報ベースから削除する前に保持する時間。これらのルートはOSPFなどの内部ゲートウェイ プロトコル用です。デフォルトは 70 秒（00:01:10）です。指定できる範囲は 00:00:10 ～ 00:01:40 です。

SYN フラッド DoS 攻撃からのサーバーの保護（TCP 代行受信）

攻撃者が一連の SYN パケットをホストに送信すると、SYN フラッディング サービス妨害（DoS）攻撃が発生します。これらのパケットは通常、スプーフィングされたIPアドレスから発信されます。SYN パケットのフラッディングが定常的に生じると、SYN キューが一杯になる状況が続き、正規ユーザーからの接続要求に対してサービスを提供できなくなります。

SYNフラッディング攻撃を防ぐために初期接続数を制限できます。初期接続とは、送信元と宛先の間で必要になるハンドシェイクを完了していない接続要求のことです。

接続の初期接続しきい値を超えると、ASA はサーバーのプロキシとして動作し、その接続がターゲットホストの SYN キューに追加されないように、SYN Cookie 方式を使用してクライアント SYN 要求に対する SYN-ACK 応答を生成します。SYN クッキーは、基本的に秘密を作成するために、MSS、タイムスタンプ、およびその他の項目の数学的ハッシュから構築される SYN-ACK で返される最初のシーケンス番号です。ASA は、正しいシーケンス番号で有効な時間ウィンドウ内にクライアントから返された ACK を受信すると、クライアントが本物であることを認証し、サーバーへの接続を許可できます。プロキシを実行するコンポーネントは、TCP 代行受信と呼ばれます。

SYN フラッド攻撃からサーバーを保護するためのエンドツーエンド プロセスでは、接続制限を設定し、TCP代行受信の統計情報をイネーブルにし、結果をモニターする必要があります。

始める前に

- 保護するサーバーの TCP SYN バックログ キューより低い初期接続制限を設定していることを確認します。これより高い初期接続制限を設定すると、有効なクライアントが、SYN 攻撃中にサーバーにアクセスできなくなります。初期接続制限に適切な値を決定するには、サーバーの容量、ネットワーク、サーバーの使用状況を入念に分析してください。
- ASA モデル上の CPU コア数によっては、同時接続および初期接続の最大数が、各コアによる接続の管理方法が原因で、設定されている数を超える場合があります。最悪の場合、ASA は最大 *n*-1 の追加接続および初期接続を許可します。ここで、*n* はコアの数です。たとえば、モデルに 4 つのコアがあり、6 つの同時接続および 4 つの初期接続を設定した場合は、各タイプで 3 つの追加接続を使用できます。ご使用のモデルのコア数を確認するには、**show cpu core** コマンドを入力します。

手順

ステップ 1 L3/L4 クラスマップを作成して、保護するサーバーを識別します。アクセスリスト一致を使用します。

```
class-map name
match parameter
```

例：

```
hostname(config)# access-list servers extended permit tcp any host 10.1.1.5 eq http
hostname(config)# access-list servers extended permit tcp any host 10.1.1.6 eq http
hostname(config)# class-map protected-servers
hostname(config-cmap)# match access-list servers
```

ステップ 2 クラスマップ トラフィックで実行する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して、クラス マップを指定します。

```
policy-map name
class name
```

例：

```
hostname(config)# policy-map global_policy
hostname(config-pmap)# class protected-servers
```

デフォルト設定では、global_policy ポリシー マップはすべてのインターフェイスにグローバルに割り当てられます。global_policy を編集する場合は、ポリシー名として global_policy を入力します。クラス マップの場合、この手順ですでに作成したクラスを指定します。

ステップ 3 初期接続制限を設定します。

- **set connection embryonic-conn-max** *n*：許可する同時 TCP 初期接続の最大数（0 ～ 2000000）。デフォルトは 0 で、この場合は接続数が制限されません。
- **set connection per-client-embryonic-max** *n*：許可する同時 TCP 初期接続のクライアントごとの最大数（0 ～ 2000000）。デフォルトは 0 で、この場合は接続数が制限されません。
- **set connection syn-cookie-mss** 初期接続数制限に達したときに初期接続の SYN cookie を生成するためのサーバーの最大セグメントサイズ（MSS）（48 ～ 65535）。デフォルトは 1380 です。この設定は、**set connection embryonic-conn-max** または **per-client-embryonic-max** を設定する場合にのみ有効です。

例：

```
hostname(config-pmap-c)# set connection embryonic-conn-max 1000
hostname(config-pmap-c)# set connection per-client-embryonic-max 50
```

ステップ 4 既存のサービス ポリシー（global_policy という名前のデフォルト グローバル ポリシーなど）を編集している場合は、このステップを省略できます。それ以外の場合は、1 つまたは複数のインターフェイスでポリシー マップをアクティブにします。

service-policy *policymap_name* {**global** | **interface** *interface_name*}

例：

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

global キーワードはポリシー マップをすべてのインターフェイスに適用し、**interface** はポリシーを 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。インターフェイスでは、そのインターフェイスへサービス ポリシーを適用することで、グローバル ポリシーを上書きできます。各インターフェイスには、ポリシー マップを 1 つだけ適用できます。

ステップ 5 TCP 代行受信によって代行受信される攻撃の脅威検出統計情報を設定します。

threat-detection statistics tcp-intercept[**rate-interval** *minutes*] [**burst-rate** *attacks_per_sec*] [**average-rate** *attacks_per_sec*]

それぞれの説明は次のとおりです。

- **rate-interval** *minutes* は、履歴モニタリング ウィンドウのサイズを、1 ～ 1440 分の範囲で設定します。デフォルトは 30 分です。この間隔の間に、ASA は攻撃の数を 30 回サンプリングします。
- **burst-rate** *attacks_per_sec* は、syslog メッセージ生成のしきい値を 25 ～ 2147483647 の範囲内で設定します。デフォルトは 1 秒間に 400 です。バースト レートがこれを超えると、syslog メッセージ 733104 が生成されます。

- **average-rate** *attacks_per_sec* は、syslog メッセージ生成の平均レートしきい値を、25 ～ 2147483647 の範囲で設定します。デフォルトは 1 秒間に 200 回です。平均レートがこれを超えると、syslog メッセージ 733105 が生成されます。

例：

```
hostname(config)# threat-detection statistics tcp-intercept
```

ステップ 6 次のコマンドを使用して結果をモニターします。

- **show threat-detection statistics top tcp-intercept** [**all** | **detail**]：攻撃を受けて保護された上位 10 サーバーを表示します。**all** キーワードは、トレースされているすべてのサーバーの履歴データを表示します。**detail** キーワードは、履歴サンプリング データを表示します。ASA はレート間隔の間に攻撃の数を 30 回サンプリングするので、デフォルトの 30 分間隔では、60 秒ごとに統計情報が収集されます。
- **clear threat-detection statistics tcp-intercept**：TCP 代行受信の統計情報を消去します。

例：

```
hostname(config)# show threat-detection statistics top tcp-intercept
Top 10 protected servers under attack (sorted by average rate)
Monitoring window size: 30 mins    Sampling interval: 30 secs
<Rank> <Server IP:Port> <Interface> <Ave Rate> <Cur Rate> <Total> <Source IP (Last Attack
 Time)>
------------------------------------------------------------------------------
1    10.1.1.5:80 inside 1249 9503 2249245 <various> Last: 10.0.0.3 (0 secs ago)
2    10.1.1.6:80 inside 10 10 6080 10.0.0.200 (0 secs ago)
```

異常な TCP パケット処理のカスタマイズ（TCP マップ、TCP ノーマライザ）

TCP ノーマライザは、異常なパケットを識別します。これは、ASA による検出時に処理（パケットを許可、ドロップ、またはクリア）させることができます。TCP 正規化は、攻撃から ASA を保護するのに役立ちます。TCP 正規化は常にイネーブルになっていますが、機能の一部の動作をカスタマイズできます。

デフォルト コンフィギュレーションには、次の設定が含まれます。

```
no check-retransmission
no checksum-verification
exceed-mss allow
queue-limit 0 timeout 4
reserved-bits allow
syn-data allow
synack-data drop
invalid-ack drop
seq-past-window drop
tcp-options range 6 7 clear
```

```
tcp-options range 9 18 clear
tcp-options range 20 255 clear
tcp-options md5 allow
tcp-options mss allow
tcp-options selective-ack allow
tcp-options timestamp allow
tcp-options window-scale allow
ttl-evasion-protection
urgent-flag clear
window-variation allow-connection
```

TCP ノーマライザをカスタマイズするには、まず、TCP マップを使用して設定を定義します。次に、サービス ポリシーを使用して、選択したトラフィック クラスにマップを適用できます。

手順

ステップ 1 確認する TCP 正規化基準を指定するための TCP マップを作成します。**tcp-map** *tcp-map-name*

ステップ 2 次の 1 つ以上のコマンドを入力して TCP マップ基準を設定します。入力しないコマンドにはデフォルトが使用されます。設定を無効化するには、コマンドの **no** 形式を使用します。

- **check-retransmission**：一貫性のない TCP 再送信を防止します。このコマンドは、デフォルトでディセーブルになっています。
- **checksum-verification**：TCP チェックサムを検証し、検証に失敗したパケットをドロップします。このコマンドは、デフォルトでディセーブルになっています。
- **exceed-mss** {**allow** | **drop**}：データ長が TCP 最大セグメント サイズを超えるパケットを許可またはドロップします。デフォルトでは、パケットを許可します。
- **invalid-ack** {**allow** | **drop**}：無効な ACK を含むパケットを許可またはドロップします。デフォルトでは、パケットをドロップします（パケットが許可される WAAS 接続を除く）。次のような場合に無効な ACK が検出される可能性があります。
  - TCP 接続が SYN-ACK-received ステータスでは、受信した TCP パケットの ACK 番号が次の TCP パケット送信のシーケンス番号と同じでない場合、その ACK は無効です。
  - 受信した TCP パケットの ACK 番号が次の TCP パケット送信のシーケンス番号より大きい場合は常に、その ACK は無効です。
- **queue-limit** *pkt_num* [**timeout** *seconds*]：バッファに格納して TCP 接続の正しい順序に設定できる、異常なパケットの最大数を設定します。1 ～ 250 パケットです。デフォルト値の 0 は、この設定がディセーブルであり、トラフィックのタイプに応じたデフォルトのシステム キュー制限が使用されることを意味します。
  - アプリケーション インスペクション（**inspect** コマンド）、、および TCP インスペクション再送信（TCP マップ **check-retransmission** コマンド）のための接続のキュー制限は、3 パケットです。ASA が異なるウィンドウ サイズの TCP パケットを受信した場合は、アドバタイズされた設定と一致するようにキュー制限がダイナミックに変更されます。

- 他の TCP 接続の場合は、異常なパケットはそのまま通過します。

queue-limit コマンドを 1 以上に設定した場合、すべての TCP トラフィックに対して許可される異常なパケットの数は、この設定と一致します。たとえば、アプリケーションインスペクション、および TCP check-retransmission のトラフィックの場合、TCP パケットからアドバタイズされたすべての設定が**キュー制限**設定を優先して、無視されます。その他の TCP トラフィックについては、異常なパケットはバッファに格納されて、そのまま通過するのではなく、正しい順序に設定されます。

timeout *seconds* 引数は、異常なパケットがバッファ内に留まることができる最大時間を設定します。設定できる値は 1 ～ 20 秒です。タイムアウト期間内に正しい順序に設定されて渡されなかったパケットはドロップされます。デフォルトは 4 秒です。*pkt_num* 引数を 0 に設定した場合は、どのトラフィックのタイムアウトも変更できません。**timeout** キーワードを有効にするには、制限を 1 以上に設定する必要があります。

- **reserved-bits** {**allow** | **clear** | **drop**}：TCP ヘッダーの予約ビットに対するアクションを設定します。パケットを**許可**するか（ビットを変更せずに）、ビットを**クリア**してパケットを許可するか、またはパケットを**ドロップ**できます。
- **seq-past-window** {**allow** | **drop**}：パストウィンドウ シーケンス番号を含むパケットに対するアクションを設定します。つまり、受信した TCP パケットのシーケンス番号が、TCP 受信ウィンドウの右端より大きい場合です。**queue-limit** コマンドを 0（ディセーブル）に設定した場合にのみ、パケットを**許可**できます。デフォルトでは、パケットをドロップします。
- **synack-data** {**allow** | **drop**}：データを含む TCP SYNACK パケットを許可またはドロップします。デフォルトは、パケットのドロップです。
- **syn-data** {**allow** | **drop**}：データを含む SYN パケットを許可またはドロップします。デフォルトでは、パケットを許可します。
- **tcp-options** {**md5** |**mss** |**selective-ack** |**timestamp** |**window-scale** | **range** *lowerupper*} *action*：TCP オプションを使用してパケットのアクションを設定します。これらのオプションには**md5**、**mss**、**selective-ack**（選択的確認応答メカニズム）、**timestamp**、および **window-scale**（ウィンドウ スケール メカニズム）という名前が付いています。その他のオプションでは、**range** キーワードで数値を使用してオプションを指定します。範囲の制限は 6 ～ 7、9 ～ 18、20 ～ 255 です。数字別に単一オプションをターゲットにするには、上下の範囲に同じ数字を入力します。マップでコマンドを複数回入力することで、ポリシー全体を定義できます。TCP 接続をインスペクションする場合、設定に関係なく MSS オプションと選択的応答確認（SACK）オプションを除き、すべてのオプションがクリアされます。選択可能なアクションは、次のとおりです。
  - **allow** [**multiple**]：このタイプの単一オプションを含むパケットを許可します。これは、すべての名前付きオプションのデフォルトです。オプションのインスタンスが複数含まれていてもパケットを許可する場合は、**multiple** キーワードを追加します。（**multiple** キーワードは **range** では使用できません。）

- **maximum** *limit*：**mss** のみ。最大セグメント サイズを指示された制限に設定します（68 ～ 65535）。デフォルトの TCP MSS は、**sysopt connection tcpmss** コマンドで定義されます。
- **clear**：このタイプのオプションをヘッダーから削除し、パケットを許可します。これは、すべての番号付きオプションのデフォルトです。タイムスタンプ オプションを消去すると、PAWS と RTT がディセーブルになります。
- **drop**：このオプションを含むパケットをドロップします。このアクションは、**md5** および **range** でのみ使用可能です。

- **ttl-evasion-protection**：接続の最大 TTL を最初のパケットで TTL によって決定させます。後続パケットの TTL は削減できますが、増やすことはできません。システムは、TTL をその接続の以前の最小 TTL にリセットします。これによって、TTL を回避した攻撃から保護します。デフォルトでは、TTL 回避保護がイネーブルになっているため、このコマンドの **no** 形式を入力するだけです。

  たとえば、攻撃者は TTL を非常に短くしてポリシーを通過するパケットを送信できます。TTL がゼロになると、ASA とエンドポイントの間のルータはパケットをドロップします。この時点で、攻撃者は TTL を長くした悪意のあるパケットを送信できます。このパケットは、ASA にとって再送信のように見えるため、通過します。一方、エンドポイント ホストにとっては、このパケットが攻撃者によって受信された最初のパケットになります。この場合、攻撃者はセキュリティによる攻撃の防止を受けず、攻撃に成功します。

- **urgent-flag** {**allow** | **clear**}：URG フラグを含むパケットに対するアクションを設定します。パケットを**許可**するか、フラグを**クリア**してパケットを許可できます。デフォルトでは、フラグをクリアします。

  URG フラグは、ストリーム中の他のデータよりもプライオリティの高い情報がこのパケットに含まれていることを示すために使用します。TCP RFC では、URG フラグの正確な解釈が明確にされていません。そのため、エンドシステムは緊急オフセットをさまざまな方法で処理しており、これが攻撃に対する脆弱性になることがあります。

- **window-variation** {**allow** | **drop**}：予期せずにウィンドウ サイズが変更された接続を許可またはドロップします。デフォルトでは、接続を許可します。

  ウィンドウ サイズ メカニズムによって、TCP は大きなウィンドウをアドバタイズでき、続いて、過剰な量のデータを受け入れずに、はるかに小さなウィンドウをアドバタイズできます。TCP 仕様により、「ウィンドウの縮小」は極力避けることが推奨されています。この条件が検出された場合に、接続をドロップできます。

ステップ 3 サービス ポリシーを使用して、TCP マップをトラフィック クラスに適用します。

a) L3/L4 クラスマップを使用してトラフィック クラスを定義し、そのマップをポリシーマップに追加します。

```
class-map name
match parameter
policy-map name
```

class *name*

例：

```
hostname(config)# class-map normalization
hostname(config-cmap)# match any
hostname(config)# policy-map global_policy
hostname(config-pmap)# class normalization
```

デフォルト設定では、global_policy ポリシー マップはすべてのインターフェイスにグローバルに割り当てられます。global_policy を編集する場合は、ポリシー名として global_policy を入力します。クラスマップの照合ステートメントの詳細については、通過トラフィック用のレイヤ 3/4 クラス マップの作成を参照してください。

b) TCP マップを適用します：**set connection advanced-options** *tcp-map-name*

例：

```
hostname(config-pmap-c)# set connection advanced-options tcp_map1
```

c) 既存のサービス ポリシー（たとえば、global_policy という名前のデフォルト グローバル ポリシー）を編集している場合は、以上で終了です。それ以外の場合は、1 つまたは複数のインターフェイスでポリシー マップをアクティブにします。

service-policy *policymap_name* {**global** | **interface** *interface_name*}

例：

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

global キーワードはポリシー マップをすべてのインターフェイスに適用し、**interface** はポリシーを 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。インターフェイスでは、そのインターフェイスへサービス ポリシーを適用することで、グローバル ポリシーを上書きできます。各インターフェイスには、ポリシー マップを 1 つだけ適用できます。

例

たとえば、既知の FTP データ ポートと Telnet ポートの間の TCP ポート範囲に送信されるすべてのトラフィックで緊急フラグと緊急オフセット 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)# tcp-map tmap
hostname(config-tcp-map)# urgent-flag allow
hostname(config-tcp-map)# class-map urg-class
hostname(config-cmap)# match port tcp range ftp-data telnet
hostname(config-cmap)# policy-map pmap
hostname(config-pmap)# class urg-class
hostname(config-pmap-c)# set connection advanced-options tmap
```

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

非対称ルーティングの TCP ステートチェックのバイパス（TCP ステートバイパス）

ネットワークで非対称ルーティング環境を設定し、特定の接続の発信フローと着信フローが 2 つの異なる ASA デバイスを通過できる場合は、影響を受けるトラフィックに TCP ステートバイパスを実装する必要があります。

ただし、TCPステートバイパスによってネットワークのセキュリティが弱体化するため、非常に詳細に限定されたトラフィック クラスでバイパスを適用する必要があります。

ここでは、問題と解決策についてより詳細に説明します。

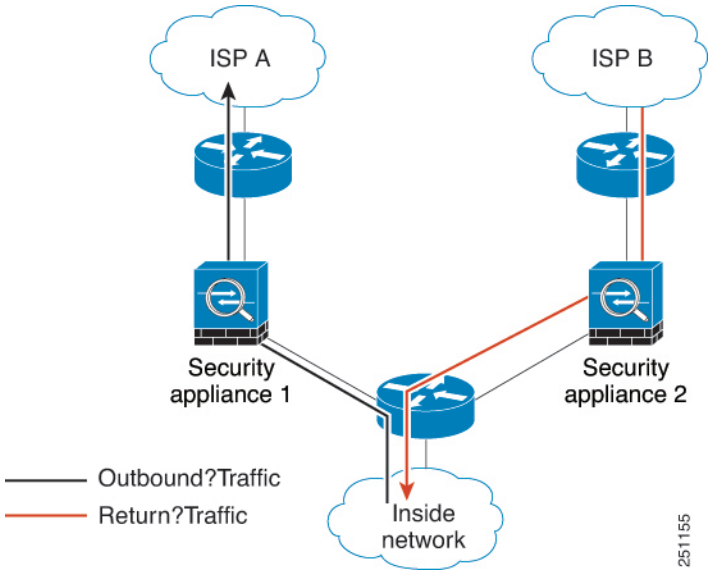
非対称ルーティングの問題

デフォルトで、ASA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。ASA では、各パケットの状態（新規接続であるか、または確立済み接続であるか）がチェックされ、そのパケットをセッション管理パス（新規接続の SYN パケット）、高速パス（確立済みの接続）、またはコントロールプレーンパス（高度なインスペクション）に割り当てることによって、ファイアウォールのパフォーマンスが最大化されます。

高速パスの既存の接続に一致する TCP パケットは、セキュリティ ポリシーのあらゆる面の再検査を受けることなくASAを通過できます。この機能によってパフォーマンスは最大になります。ただし、SYNパケットを使用してファストパスにセッションを確立する方法、およびファストパスで行われるチェック（TCP シーケンス番号など）が、非対称ルーティングソリューションの障害となる場合があります。これは、接続の発信フローと着信フローの両方が同じ ASA デバイスを通過する必要があるためです。

たとえば、ある新しい接続がセキュリティ アプライアンス 1 に到達するとします。SYNパケットはセッション管理パスを通過し、接続のエントリが高速パス テーブルに追加されます。この接続の後続パケットがセキュリティ アプライアンス 1 を通過した場合、高速パス内のエントリに一致するのでこのパケットは送信されます。しかし、後続のパケットがセキュリティ アプライアンス 2 に到着すると、SYNパケットがセッション管理パスを通過していないために、高速パスにはその接続のエントリがなく、パケットはドロップされます。次の図は、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 ASA を通過しています。

図 *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ASAデバイス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 バイパスは、高速パスでのセッションの確立方法を変更し、高速パスのインスペクションを無効化します。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ASAデバイスに入った時点で高速パスエントリが存在しない場合、高速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。いったん高速パスに入ると、トラフィックは高速パスのインスペクションをバイパスします。

TCP ステート バイパスのガイドラインと制限事項

TCP ステート バイパスでサポートされない機能

TCP ステート バイパスを使用するときは、次の機能はサポートされません。

- アプリケーションインスペクション：インスペクションでは、着信トラフィックと発信トラフィックの両方が同じ ASA を通過する必要があるため、インスペクションは TCP ステート バイパス トラフィックに適用されません。
- AAA 認証セッション：ユーザーがある ASA で認証される場合、他の ASA 経由で戻るトラフィックは、その ASA でユーザーが認証されていないため、拒否されます。
- TCP代行受信、最大初期接続制限、TCPシーケンス番号ランダム化：ASAでは接続の状態が追跡されないため、これらの機能は適用されません。
- TCP 正規化：TCP ノーマライザはディセーブルです。
- ステートフル フェールオーバー。

TCP ステート バイパスのガイドライン

変換セッションはASAごとに個別に確立されるため、TCP ステート バイパス トラフィック用に両方のデバイスでスタティック NAT を設定する必要があります。ダイナミック NAT を使用すると、デバイス 1 でのセッションに選択されるアドレスは、デバイス 2 でのセッションに選択されるアドレスとは異なります。

TCP ステート バイパスの設定

非対称ルーティング環境で TCP ステートチェックをバイパスするには、影響を受けるホストまたはネットワークのみに適用するトラフィッククラスを注意深く定義してから、サービスポリシーを使用してトラフィッククラスで TCP ステートバイパスを有効にします。バイパスによってネットワークのセキュリティが低下するため、そのアプリケーションをできるだけ制限します。

始める前に

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 idle** コマンドを TCP ステート バイパス トラフィック クラスに使用するとオーバーライドできます。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。

手順

ステップ 1 L3/L4 クラス マップを作成して、TCPステート バイパスを必要とするホストを識別します。アクセス リスト一致を使用して、送信元と宛先のホストを識別します。

```
class-map name
match parameter
```

例：

```
hostname(config)# access-list bypass extended permit tcp host 10.1.1.1 host 10.2.2.2
hostname(config)# class-map bypass-class
hostname(config-cmap)# match access-list bypass
```

ステップ 2 クラスマップ トラフィックで実行する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して、クラス マップを指定します。

```
policy-map name
class name
```

例：

```
hostname(config)# policy-map global_policy
hostname(config-pmap)# class bypass-class
```

デフォルト設定では、global_policyポリシーマップはすべてのインターフェイスにグローバルに割り当てられます。global_policy を編集する場合は、ポリシー名として global_policy を入力します。クラス マップの場合、この手順ですでに作成したクラスを指定します。

ステップ 3 クラスでTCP ステート バイパスを有効にします：**set connection advanced-options tcp-state-bypass**

ステップ 4 既存のサービス ポリシー（たとえば、global_policy という名前のデフォルト グローバル ポリシー）を編集している場合は、以上で終了です。それ以外の場合は、1つまたは複数のインターフェイスでポリシー マップをアクティブにします。

service-policy *policymap_name* {**global** | **interface** *interface_name*}

例：

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

global キーワードはポリシー マップをすべてのインターフェイスに適用し、**interface** はポリシーを 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。インターフェイスでは、そのインターフェイスへサービス ポリシーを適用することで、グローバル ポリシーを上書きできます。各インターフェイスには、ポリシー マップを 1 つだけ適用できます。

例

TCP ステート バイパスの設定例を次に示します。

```
hostname(config)# access-list tcp_bypass extended permit tcp 10.1.1.0 255.255.255.224
any

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 interface outside
```

TCP シーケンスのランダム化のディセーブル

各 TCP 接続には、クライアントで生成される ISN とサーバーで生成される ISN の 2 つの ISN があります。ASA は、着信と発信の両方向で通過する TCP SNY 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。ただし、TCP シーケンスのランダム化は、TCP SACK（選択的確認応答）を実質的に破棄します。クライアントが認識するシーケンス番号がサーバーが認識するものと異なるためです。

たとえば、データがスクランブルされるため、必要に応じて TCP 初期シーケンス番号ランダム化をディセーブルにすることができます。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。
- ASA 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。ランダム化により、MD5 チェックサムは分解されます。
- ASA で接続のシーケンス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。
- ISA 3000 のハードウェア バイパスを有効にします。ISA 3000 がデータ パスの一部でなくなると、TCP 接続はドロップされます。

（注） クラスタリングを使用する場合は、TCP シーケンスのランダム化を無効にすることは推奨されません。SYN/ACK パケットがドロップされる可能性があるため、一部の TCP セッションが確立されない可能性があります。

手順

ステップ 1 L3/L4 クラスマップを作成して、TCP シーケンス番号をランダム化しないトラフィックを識別します。クラス マップは、TCP トラフィック用にします。TCP ポート一致を行う特定のホストを識別したり（ACL を使用して）、任意のトラフィックと照合したりすることができます。

```
class-map name
match parameter
```

例：

```
hostname(config)# access-list preserve-sq-no extended permit tcp any host 10.2.2.2
hostname(config)# class-map no-tcp-random
hostname(config-cmap)# match access-list preserve-sq-no
```

ステップ 2 クラスマップ トラフィックで実行する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して、クラス マップを指定します。

```
policy-map name
class name
```

例：

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

```
hostname(config-pmap)# class no-tcp-random
```

デフォルト設定では、global_policyポリシーマップはすべてのインターフェイスにグローバルに割り当てられます。global_policy を編集する場合は、ポリシー名として global_policy を入力します。クラス マップの場合、この手順ですでに作成したクラスを指定します。

ステップ 3 クラスで TCP シーケンス番号ランダム化をディセーブルにします。

set connection random-sequence-number disable

後でオンに戻す場合は、「disable」を **enable** に置き換えます。

ステップ 4 既存のサービス ポリシー（たとえば、global_policy という名前のデフォルト グローバル ポリシー）を編集している場合は、以上で終了です。それ以外の場合は、1つまたは複数のインターフェイスでポリシー マップをアクティブにします。

service-policy *policymap_name* {**global** | **interface** *interface_name*}

例：

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

global キーワードはポリシー マップをすべてのインターフェイスに適用し、**interface** はポリシーを 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。インターフェイスでは、そのインターフェイスへサービス ポリシーを適用することで、グローバル ポリシーを上書きできます。各インターフェイスには、ポリシー マップを 1 つだけ適用できます。

大規模フローのオフロード

データセンターのサポートされているデバイス上でCisco ASA を展開する場合は、超高速パスにオフロードするトラフィックを識別して、トラフィックがNIC 自身でスイッチングされるようにできます。オフロードによって、大容量ファイルの転送など、データ集約型アプリケーションのパフォーマンスを向上させることができます。

- ハイパフォーマンス コンピューティング（HPC）調査サイト。ここでは、ASA はストレージと高コンピューティング ステーション間で展開されます。1 つの調査サイトが NFS 経由のFTP ファイル転送またはファイル同期を使用してバックアップを行うと、大量のデータ トラフィックが ASA 上のすべてのコンテキストに影響を与えます。NFS を介する FTP ファイル転送およびファイル同期のオフロードによって、他のトラフィックへの影響が軽減されます。
- 主にコンプライアンス目的で使用される High Frequency Trading（HFT）。ここでは、ASA はワークステーションと Exchange 間で展開されます。セキュリティは通常は問題にはなりませんが、遅延は大きな問題です。

オフロードされる前に、ASA は接続の確立時にアクセス ルールやインスペクションなどの通常のセキュリティ処理を最初に適用します。ASA のセッションも切断されます。ただし、一旦

接続が確立されると、オフロードされる資格があれば、さらなる処理が ASA ではなく NIC で行われます。

オフロードされたフローは、基本的な TCP フラグとオプションのチェック、設定した場合にはチェックサムの確認などの、制限されたステートフルインスペクションを受信し続けます。システムは必要に応じてさらなる処理のためにファイアウォール システムへのパケットを選択的に増やすことができます。

オフロードが可能なフローを識別するには、フロー オフロード サービスを適用するサービス ポリシールールを作成します。一致するフローはその後、次の条件を満たす場合にオフロードされます。

- IPv4 アドレスのみ。
- TCP、UDP、GRE のみ。
- 標準または 802.1Q タグ付きイーサネット フレームのみ。
- （トランスペアレント モードのみ。）インターフェイスを 2 つだけ含むブリッジ グループのマルチキャスト フロー。

オフロードされたフローのリバース フローもオフロードされます。

フロー オフロードの制限事項

すべてのフローをオフロードできるわけではありません。オフロードの後でも、フローを特定の条件下でのオフロードから除外することができます。次に、制限事項の一部を示します。

デバイスによる制限

この機能は、以下のデバイスでサポートされています。

- Cisco Secure Firewall 3100
- Cisco Secure Firewall 4200
- Firepower 4100/9300

オフロードできないフロー

次のタイプのフローはオフロードできません。

- IPv6 アドレッシングなど、IPv4 アドレッシングを使用しないフロー。
- TCP、UDP、GRE 以外のプロトコルに対するフロー。

（注） PPTP GRE 接続はオフロードできません。

- インスペクションが必要なフロー。FTP など場合によっては、コントロール チャネルはオフロードできませんがセカンダリ データ チャネルはオフロードできます。
- デバイスで終端する IPsec および TLS/DTLS VPN 接続。

- ルーテッド モードのマルチキャスト フロー。
- 3 つ以上のインターフェイスがあるブリッジ グループに対するトランスペアレント モードのマルチキャスト フロー。
- TCP インターセプト フロー。
- TCP ステートバイパスフロー。同じトラフィックにフローオフロードと TCP ステートバイパスを設定することはできません。
- AAA カットスループロキシフロー。
- Vpath、VXLAN 関連のフロー。
- セキュリティ グループでタグ付けされたフロー。
- Cisco Secure Firewall 4200 を除く：クラスタで非対称フローが発生した場合に備えて、別のクラスタ ノードから転送されるリバース フロー。
- クラスタ内の一元化されたフロー（フローのオーナーが制御ユニットでない場合）。

その他の制限事項

- フローオフロードとデッド接続検出（DCD）は互換性がありません。オフロードできる接続に DCD を設定しないでください。
- フローオフロード条件に一致する複数のフローがキューイングされて、ハードウェア上の同じ場所に同時にオフロードされる場合、最初のフローのみがオフロードされます。他のフローは通常どおりに処理されます。これをコリジョン（衝突）といいます。この状況の統計を表示するには、CLI で **show flow-offload flow** コマンドを使用します。
- オフロードされたフローはFXOSインターフェイスを通過しますが、それらのフローの統計は論理デバイスインターフェイスには表示されません。したがって、論理デバイスインターフェイスのカウンタとパケットレートには、オフロードされたフローは反映されません。

クラスタのリダイレクト

既存のフローのトラフィックが別のノードに送信されると、そのトラフィックはクラスタ制御リンクを介してオーナー ノードにリダイレクトされます。非対称フローはクラスタ制御リンクに大量のトラフィックを作り出す可能性があるため、これらのフローをオフロードすると、分散型サイト間 VPN モードなどのパフォーマンスを向上させることができます。この機能は、次でサポートされています。

- Cisco Secure Firewall 4200
- ルーテッド モードとトランスペアレント モード
- スパンド EtherChannel モード
- マルチ コンテキスト モード

オフロードを無効にする条件

フローがオフロードされた後、フロー内のパケットは次の条件を満たす場合に ASA に返され、さらに処理されます。

- タイムスタンプ以外の TCP オプションが含まれている。
- フラグメント化されている。
- これらは等コストマルチパス（ECMP）ルーティングの対象であり、入力パケットは 1 つのインターフェイスから別のインターフェイスに移動する。

フロー オフロードの設定

フローオフロードを設定するには、サービスをイネーブルにしてから、オフロードする対象トラフィックを識別するサービス ポリシーを作成する必要があります。Firepower 4100/9300 の場合：最初にサービスを有効にするときは、再起動する必要があります。Secure Firewall 3100/4200 のフロー オフロードはデフォルトで有効です。

手順

ステップ 1 フロー オフロード サービスをイネーブルにします。

flow-offload enable

フロー オフロードは Secure Firewall 3100/4200 に対してデフォルトで有効です。

Firepower 4100/9300 の場合： 最初にサービスを有効にするときは、再起動する必要があります。

リロードが必要な場合、ヒットレスなモード変更を行うには、クラスタまたはフェールオーバー ペアに特別な考慮事項があります。

- [クラスタリング（Clustering）]：最初に制御ユニット上でコマンドを入力しますが、制御ノードをすぐにリブートしないでください。代わりに、クラスタの各ノードを最初にリブートしてから、制御ノードに戻ってリブートします。次に、制御ノード上でオフロード サービス ポリシーを構成します。
- フェールオーバー：最初にアクティブユニット上でコマンドを入力しますが、アクティブ ユニットをすぐにリブートしないでください。代わりに、スタンバイユニットをリブートしてから、アクティブ ユニットをリブートします。次に、アクティブ ユニット上でオフロード サービス ポリシーを設定します。

マルチコンテキスト モードでは、フロー オフロードを有効または無効にすると、すべてのコンテキストのフローオフロードが有効または無効になります。コンテキストごとに異なる設定を使用することはできません。

例：

```
ciscoasa(config)# flow-offload enable
```

```
WARNING: This command will take effect after the running-config is
saved and the system has been rebooted.

ciscoasa(config)# write memory
ciscoasa(config)# reload
```

ステップ 2 （Cisco Secure Firewall 4200）クラスタ リダイレクトを有効にします。

flow-offload cluster-redirect

クラスタ リダイレクトはデフォルトで有効になっています。この機能には、**flow-offload enable** コマンドが必要ですが、このコマンドもデフォルトで有効になっています。

非対称フローの場合、クラスタリダイレクトにより、転送ノードはハードウェアにフローをオフロードできます。既存のフローのトラフィックが別のノードに送信されると、そのトラフィックはクラスタ制御リンクを介してオーナー ノードにリダイレクトされます。非対称フローは、クラスタ制御リンクに大量のトラフィックを作成する可能性があるため、フォワーダにこれらのフローをオフロードさせると、パフォーマンスが向上します。

ステップ 3 オフロードする対象のトラフィックを識別するサービス ポリシー ルールを作成します。

a) フロー オフロードの対象となるトラフィックを識別する L3/L4 クラス マップを作成します。アクセス リストまたはポートによる照合は最も一般的なオプションです。

```
class-map name
match parameter
```

例：

```
hostname(config)# access-list offload permit tcp 10.1.1.0 255.255.255.224 any
hostname(config)# class-map flow_offload
hostname(config-cmap)# match access-list offload
```

b) クラスマップ トラフィックで実行する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して、クラス マップを指定します。

```
policy-map name
class name
```

例：

```
hostname(config)# policy-map offload_policy
hostname(config-pmap)# class flow_offload
```

デフォルト設定では、global_policy ポリシー マップはすべてのインターフェイスにグローバルに割り当てられます。global_policy を編集する場合は、ポリシー名として global_policy を入力します。クラス マップの場合、この手順ですでに作成したクラスを指定します。

c) クラスに対し、フロー オフロードをイネーブルにします。 **set connection advanced-options flow-offload**

d) 既存のサービス ポリシー（たとえば、global_policy という名前のデフォルト グローバル ポリシー）を編集している場合は、以上で終了です。それ以外の場合は、1 つまたは複数のインターフェイスでポリシー マップをアクティブにします。

service-policy *policymap_name* {**global** | **interface** *interface_name*}

例：

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

global キーワードはポリシー マップをすべてのインターフェイスに適用し、**interface** はポリシーを 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。インターフェイスでは、そのインターフェイスへサービス ポリシーを適用することで、グローバル ポリシーを上書きできます。各インターフェイスには、ポリシー マップを 1 つだけ適用できます。

例

次に、10.1.1.0 255.255.255.224 サブネットからのすべての TCP トラフィックをオフロード対象として分類し、ポリシーを外部インターフェイスにアタッチする例を示します。

```
hostname(config)# access-list offload permit tcp 10.1.1.0 255.255.255.224 any
hostname(config)# class-map flow_offload
hostname(config-cmap)# match access-list offload
hostname(config)# policy-map offload_policy
hostname(config-pmap)# class flow_offload
hostname(config-pmap-c)# set connection advanced-options flow-offload
hostname(config)# service-policy offload_policy interface outside
```

IPsec フローのオフロード

IPsec フローのオフロードを使用するように、サポートするデバイスモデルを設定できます。IPsec サイト間 VPN またはリモートアクセス VPN セキュリティ アソシエーション (SA) の初期設定後、IPsec 接続はデバイスのフィールド プログラマブル ゲート アレイ (FPGA) にオフロードされるため、デバイスのパフォーマンスが向上します。Cisco Secure Firewall 1200 シリーズでは、デバイスのパフォーマンスを向上させるために、IPsec 接続が Marvell Cryptographic Accelerator（CPT）にオフロードされます。

オフロード操作は、特に、入力の事前復号および復号処理と出力の事前暗号化および暗号化処理に関連しています。システムソフトウェアは、セキュリティポリシーを適用するための内部フローを処理します。

IPsec フローのオフロードはデフォルトで有効になっており、次のデバイスタイプに適用されます。

- Cisco Secure Firewall 1200

- Cisco Secure Firewall 3100
- Cisco Secure Firewall 4200

IPsec フローオフロードは、デバイスの VTI ループバック インターフェイスが有効になっている場合にも使用されます。

クラスタ分散型サイト間 VPN モードの非対称フローの場合、IPsec フロー オフロードにより、フローオーナーは、クラスタ制御リンクを介して転送されたハードウェア内の IPsec トラフィックを復号できます。この機能は構成可能ではありません。IPsec フローのオフロードを有効にすると常に使用できます。

IPsec フローのオフロードに関する制約事項

次の IPsec フローはオフロードされません。

- IKEv1 トンネル。IKEv2 トンネルのみがオフロードされます。IKEv2 は、より強力な暗号をサポートしています。
- ボリュームベースのキー再生成が設定されているフロー。
- 圧縮が設定されているフロー。
- トランスポートモードのフロー。トンネルモードのフローのみがオフロードされます。
- AH 形式。ESP/NAT-T 形式のみがサポートされます。
- ポストフラグメンテーションが設定されているフロー。
- 64 ビット以外のアンチリプレイ ウィンドウ サイズを持ち、アンチリプレイが無効になっていないフロー。
- ファイアウォールフィルタが有効になっているフロー。
- マルチ コンテキスト モード。

IPsec フローオフロードの設定

IPsec フローのオフロードは、この機能をサポートするハードウェア プラットフォームではデフォルトで有効になっています。ただし、出力最適化はデフォルトでは有効になっていないため、この機能が必要な場合は構成する必要があります。

始める前に

IPsec フロー オフロードはグローバルに構成されます。選択したトラフィック フローに対して設定することはできません。

この機能を無効にするには、このコマンドの **no** 形式を使用します。

現在の設定状態を表示するには、**show flow-offload ipsec info** コマンドを使用します。

手順

ステップ 1 IPsec フロー オフロードを有効にします。

flow-offload-ipsec

ステップ 2 出力最適化を有効にすることで、データパスを最適化して、単一トンネルフローのパフォーマンスを向上させます。

flow-offload-ipsec egress-optimization

出力最適化の構成は、フロー オフロードとは別です。ただし、出力最適化を有効にしても、IPsec フロー オフロードも有効にしないかぎり無意味です。出力最適化はデフォルトでは有効になっていません。

DTLS 暗号化アクセラレーション

ASAは、FPGA および Nitrox V 暗号化アクセラレータを使用して、次のモデルの DTLS 暗号化アクセラレーションをサポートします。

- Cisco Secure Firewall 3100
- Cisco Secure Firewall 4200

この機能により、DTLS で暗号化および復号されたトラフィックのスループットが向上します。IPv4 と IPv6 の両方のトラフィックがサポートされます。

ASAは、遅延を改善するために、出力暗号化パケットの最適化も実行します。データパスを最適化して、単一トンネルフローのパフォーマンスを向上させます。

どちらの機能もデフォルトで有効になっており、DTLS 1.2 でのみ動作します。

DTLS 暗号化アクセラレーション

デフォルトでは、DTLS 暗号化アクセラレーションは有効になっています。必要に応じて無効にできます。

Cisco ASA は、以下の条件下では DTLS 暗号化アクセラレーションを実行しません。

- フローが DTLS 1.0 またはパケット圧縮を使用している。
- DTLS キーが再生成されている。
- クラスタリングまたはマルチコンテキストモード。

手順

ステップ 1 デバイスで DTLS 暗号化アクセラレーションを無効にします。
no flow-offload-dtls

例：

```
ciscoasa(config)# no flow-offload-dtls
```

再び有効にするには、**flow-offload-dtls** コマンドを使用します。

ステップ 2 出力暗号化パケットの最適化を無効にし、遅延を改善します。
no flow-offload-dtls egress-optimization

例：

```
ciscoasa(config)# flow-offload-dtls egress-optimization
```

再び有効にするには、**flow-offload-dtls egress-optimization** コマンドを使用します。

DTLS 暗号化アクセラレーションのモニタリング

DTLS暗号化アクセラレーションと出力暗号化パケットの最適化を確認してモニターするには、Threat Defense デバイスで以下の CLI コマンドを使用します。

- DTLS 暗号化アクセラレーションと出力暗号化パケットの最適化のステータスを確認するには、以下のコマンドを使用します。

```
ciscoasa# show flow-offload-dtls info
  DTLS offload : Enabled
  Egress Optimization: Enabled
```

- DTLS 暗号化アクセラレーションの統計を表示するには、以下のコマンドを使用します。

```
ciscoasa# show flow-offload-dtls statistics
  Packet stats of Pipe 0
  ----------------------
  Rx Packet count : 975638666
  Tx Packet count : 975638666
  Error Packet count : 0
  Drop Packet count : 0

  CAM stats of Pipe 0
  ----------------------
  Option ID Table CAM Hit Count : 1145314723
  Option ID Table CAM Miss Count : 0
  Tunnel Table CAM Hit Count : 0
  Tunnel Table CAM Miss Count : 0
  6-Tuple CAM Hit Count : 975638666
```

```
  6-Tuple CAM Miss Count : 169676057
   NOTE: The counters displayed are cumulative counters
   for all offload applications and indicates the total packets
  offloaded
```

- デバイスのNitrox V暗号化アクセラレータの統計を表示するには、以下のコマンドを使用します。

```
ciscoasa# show crypto accelerator statistics

Crypto Accelerator Status
-------------------------
<snip>
[Offloaded SSL Input statistics, Pipe 0]
      Input packets: 290593023
      Input bytes: 147049729714
      Decrypted packets: 290593023
      Decrypted bytes: 147049729714
   [Offloaded SSL Output statistics, Pipe 0]
      Output packets: 254271808
      Output bytes: 136352952720
      Encrypted packets: 254271808
      Encrypted bytes: 136352952720
.
.
.
```

特定のトラフィック クラスの接続の設定（すべてのサービス）

サービス ポリシーを使用して、特定のトラフィック クラスに対してさまざまな接続の設定を行うことができます。サービス ポリシーを使用して、次の内容を実行します。

- DoS 攻撃と SYN フラッディング攻撃から保護するのに使用される接続制限と接続タイムアウトをカスタマイズします。
- アイドル状態でも有効な接続を維持するように、Dead Connection Detection（DCD; デッド接続検出）を実装します。
- TCP シーケンス番号ランダム化が不要な場合、それをディセーブルにします。
- TCP ノーマライザが異常な TCP パケットから保護する方法をカスタマイズします。
- 非対称ルーティングの対象であるトラフィックに対して TCP ステートバイパスを導入します。バイパス トラフィックはインスペクションの対象になりません。
- SCTP ステートフル インスペクションをオフにするには、Stream Control Transmission Protocol（SCTP）ステート バイパスを実装します。
- サポート対象のハードウェア プラットフォームのパフォーマンスを向上させるには、フロー オフロードを実装します。
- ASA がトレース ルート出力に表示されるように、パケットの存続可能時間（TTL）をデクリメントします。

（注） 存続可能時間を減らすと、TTL が 1 のパケットはドロップされますが、接続に TTL がもっと長いパケットが含まれている可能性があるという仮定の下に、セッションに対して接続が開かれます。OSPF hello パケットなどの一部のパケットは TTL = 1 で送信されるため、トランスペアレントモードの ASA デバイスでは、パケット存続時間をデクリメントすると予期しない結果が発生する可能性があります。ASA がルーテッドモードで動作している場合は、パケット存続時間の設定をデクリメントしても OSPF のプロセスに影響を与えません。

同時に使用できない TCP ステート バイパスと TCP ノーマライザのカスタマイズを除き、特定のトラフィック クラスに対してこれらの設定の任意の組み合わせを設定できます。

ヒント この手順は、ASA を通過するトラフィックのサービス ポリシーを示します。管理（to the box）トラフィックに対して接続の最大数と初期接続の最大数を設定することもできます。

始める前に

TCP ノーマライザをカスタマイズする場合は、続行する前に必要な TCP マップを作成してください。

ここでは、**set connection** コマンド（接続制限と TCP シーケンス番号ランダム化の）と **set connection timeout** コマンドについてパラメータごとに個別に説明します。ただし、1 つの行にこれらのコマンドを入力できます。これらのコマンドを個別に入力した場合、1 つのコマンドとしてコンフィギュレーションに表示されます。

手順

ステップ 1 L3/L4 クラスマップを作成して、接続の設定をカスタマイズするトラフィックを識別します。

```
class-map name
match parameter
```

例：

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

照合ステートメントについては、通過トラフィック用のレイヤ 3/4 クラス マップの作成を参照してください。

ステップ 2 クラスマップ トラフィックで実行する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して、クラス マップを指定します。

```
policy-map name
class name
```

例：

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

デフォルト設定では、global_policy ポリシーマップはすべてのインターフェイスにグローバルに割り当てられます。global_policy を編集する場合は、ポリシー名として global_policy を入力します。クラス マップの場合、この手順ですでに作成したクラスを指定します。

ステップ 3 接続制限と TCP シーケンス番号ランダム化を設定します。（TCP 代行受信）

デフォルトでは、接続制限はありません。制限を実装すると、システムはそれらの追跡を開始する必要があります。これにより、CPU とメモリの使用率が増加し、特にクラスタでは高負荷がかかったシステムに動作上の問題が発生する可能性があります。

- **set connection conn-max** *n*：（TCP、UDP、SCTP）。クラス全体で許可される同時接続の最大数（0 ～ 2000000）。デフォルトは 0 で、この場合は接続数が制限されません。TCP 接続の場合、これは確立された接続のみに適用されます。
  - 同時接続を許可するように 2 つのサーバーが設定されている場合、接続制限数は、設定されている各サーバーに別々に適用されます。
  - 制限がクラスに適用されるため、1 つの攻撃ホストがすべての接続を使い果たし、クラスに一致する他のホストが使用できる接続がなくなる可能性があります。

- **set connection per-client-max** *n*：（TCP、UDP、SCTP）。クライアントごとに許可する同時接続の最大数（0 ～ 2000000）。デフォルトは 0 で、この場合は接続数が制限されません。この引数では、クラスに一致する各ホストに許可される同時接続最大数が制限されます。TCP 接続の場合、これには確立済み接続、ハーフオープン接続、ハーフクローズ接続が含まれています。

- **set connection embryonic-conn-max** *n*：許可される同時初期 TCP 接続の最大数（0 ～ 2000000）。デフォルトは 0 で、この場合は接続数が制限されません。0 以外の制限を設定することで、TCP 代行受信をイネーブルにします。代行受信によって、TCP SYN パケットを使用してインターフェイスをフラッディングする DoS 攻撃から内部システムを保護します。また、クライアントごとのオプションを設定して、SYN フラッディングから保護します。

- **set connection per-client-embryonic-max** *n*：クライアントごとに許可される同時初期 TCP 接続の最大数（0 ～ 2000000）。デフォルトは 0 で、この場合は接続数が制限されません。

- **set connection syn-cookie-mss** 初期接続数制限に達したときに初期接続の SYN cookie を生成するためのサーバーの最大セグメントサイズ（MSS）（48 ～ 65535）。デフォルトは

1380 です。この設定は、**set connection embryonic-conn-max** または **per-client-embryonic-max** を設定する場合にのみ有効です。

- **set connection random-sequence-number** {**enable** | **disable**}：TCP シーケンス番号ランダム化をイネーブルまたはディセーブルにするかどうか。デフォルトでは、ランダム化がイネーブルになっています。

例：

```
hostname(config-pmap-c)# set connection conn-max 256 random-sequence-number disable
```

ステップ 4 接続タイムアウトと Dead Connection Detection（DCD; デッド接続検出）を設定します。

次に説明するデフォルト値は、**timeout** コマンドを使用してこれらの動作のグローバルのデフォルト値を変更していないことを前提としています。グローバルのデフォルト値はここで説明する値を上書きします。接続がタイムアウトしないように、**0** を入力してタイマーをディセーブルにします。

- **set connection timeout embryonic** *hh*:*mm*:*ss*：TCP 初期（ハーフオープン）接続を閉じるまでのタイムアウト期間（0:0:5 ～ 1193:00:00）。デフォルト値は 0:0:30 です。
- **set connection timeout idle** *hh*:*mm*:*ss* [**reset**]：いずれかのプロトコルの確立された接続が閉じてからのアイドル タイムアウト期間（0:0:1 から 1193:0:0）。デフォルト値は 1:0:0 です。TCP トラフィックの場合、**reset** キーワードを指定すると、接続のタイムアウト時にリセット パケットが TCP エンドポイントに送信されます。

  デフォルトの **udp** アイドル タイムアウトは 2 分です。デフォルトの **icmp** アイドル タイムアウトは 2 秒です。デフォルトの **esp** および **ha** アイドル タイムアウトは 30 秒です。その他すべてのプロトコルでは、デフォルトのアイドル タイムアウトは 2 分です。
- **set connection timeout half-closed** *hh*:*mm*:*ss*：ハーフクローズ接続を閉じるまでのアイドル タイムアウト期間（9.1(1) 以前の場合は 0:5:0 ～ 1193:0:0、9.1(2) 以降の場合は 0:0:30 ～ 1193:0: 0）。デフォルト値は 0:10:0 です。ハーフクローズの接続は DCD の影響を受けません。また、ASA は、ハーフクローズ接続を切断するときにリセットを送信しません。
- **set connection timeout dcd** [*retry-interval* [*max_retries*]]：Dead Connection Detection（DCD; デッド接続検出）をイネーブルにします。アイドル接続の期限が切れる前に、ASA はエンドホストにプローブを送信して接続が有効であるかどうかを判断します。両方のホストが応答した場合は、接続が維持されます。それ以外の場合は、接続が解放されます。トランスペアレント ファイアウォール モードで動作している場合、エンドポイントにスタティック ルートを設定する必要があります。オフロードも行われる接続には DCD を設定できないため、DCD とフローオフロードのトラフィック クラスが重複しないようにしてください。発信側と受信側で送信された DCD プローブの個数を追跡するには、**show conn detail** コマンドを使用します。

  retry-interval には、DCD プローブに応答がない場合に別のプローブを送信するまで待機する時間を、*hh*:*mm*:*ss* 形式で、0:0:1 から 24:0:0 の範囲で設定します。デフォルト値は 0:0:15 です。max-retries には、接続が無活動状態であると宣言するまでに失敗する DCD の連続再試行回数を設定します。最小値は 1、最大値は 255 です。デフォルトは 5 分です。

クラスタまたは高可用性構成で動作しているシステムでは、間隔を1分（0:1:0）未満に設定しないことを推奨します。接続をシステム間で移動する必要がある場合、必要な変更には 30 秒以上かかり、変更が行われる前に接続が削除される場合があります。

例：

```
hostname(config-pmap-c)# set connection timeout idle 2:0:0 embryonic 0:40:0
half-closed 0:20:0 dcd
```

ステップ 5 クラスに一致するパケットの存続可能時間（TTL）をデクリメントします：**set connection decrement-ttl**

このコマンド、および **icmp unreachable** コマンドは、ASA をホップの 1 つとして表示する ASA 経由の traceroute を可能とするために必要です。

例：

```
hostname(config)# class-map global-policy
hostname(config-cmap)# match any
hostname(config-cmap)# exit
hostname(config)# policy-map global_policy
hostname(config-pmap)# class global-policy
hostname(config-pmap-c)# set connection decrement-ttl
hostname(config-pmap-c)# exit
hostname(config)# icmp unreachable rate-limit 50 burst-size 6
```

ステップ 6 接続詳細オプションを設定します。

詳細オプションは、通常の状況では不要な特別な用途の設定です。これらのオプションは、**set connection advanced-options** コマンドを使用して設定します。

- **set connection advanced-options** *tcp_map_name*：TCP マップを適用することで、TCP ノーマライザの動作をカスタマイズします。詳細については、異常な TCP パケット処理のカスタマイズ（TCP マップ、TCP ノーマライザ） （9 ページ）を参照してください。
- **set connection advanced-options tcp-state-bypass**：TCP ステート バイパスを実装します。詳細については、非対称ルーティングの TCP ステートチェックのバイパス（TCP ステート バイパス） （14 ページ）を参照してください。
- **set connection advanced-options sctp-state-bypass**：SCTP ステート バイパスを実装して、SCTP ステートフル インスペクションを無効にします。詳細については、SCTP ステートフル インスペクションを参照してください。
- **set connection advanced-options flow-offload**：（Firepower 4100/9300 シャーシ の ASA、FXOS 1.1.3 以降のみ。）フローのオフロードを実装します。フローが NIC 自体で切り替えられる超高速パスにオフロードされる適切なトラフィック。**flow-offload enable** コマンド（これはサービス ポリシーの一部ではありません）も入力する必要があります。

例：

```
hostname(config-pmap-c)# set connection advanced-options tcp_map1
```

ステップ 7 既存のサービス ポリシー（たとえば、global_policy という名前のデフォルト グローバル ポリシー）を編集している場合は、以上で終了です。それ以外の場合は、1つまたは複数のインターフェイスでポリシー マップをアクティブにします。

service-policy *policymap_name* {**global** | **interface** *interface_name*}

例：

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

global キーワードはポリシー マップをすべてのインターフェイスに適用し、**interface** はポリシーを 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。インターフェイスでは、そのインターフェイスへサービス ポリシーを適用することで、グローバル ポリシーを上書きできます。各インターフェイスには、ポリシー マップを 1 つだけ適用できます。

例

次の例では、すべてのトラフィックに対して接続の制限値とタイムアウトを設定しています。

```
hostname(config)# class-map CONNS
hostname(config-cmap)# match any
hostname(config-cmap)# policy-map CONNS
hostname(config-pmap)# class CONNS
hostname(config-pmap-c)# set connection conn-max 1000 embryonic-conn-max 3000
hostname(config-pmap-c)# set connection timeout idle 2:0:0 embryonic 0:40:0
half-closed 0:20:0 dcd
hostname(config-pmap-c)# service-policy CONNS interface outside
```

複数のパラメータを使用して **set connection** コマンドを入力するか、各パラメータを別々のコマンドとして入力できます。ASA 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。たとえば、クラス コンフィギュレーション モードで次の 2 つのコマンドを入力するとします。

```
hostname(config-pmap-c)# set connection conn-max 600
hostname(config-pmap-c)# set connection embryonic-conn-max 50
```

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単一の結合コマンドとして表示されます。

```
set connection conn-max 600 embryonic-conn-max 50
```

TCP オプションの構成

各種オプションを構成して、TCP 動作のいくつかの側面を制御できます。これらの設定のデフォルト値は、ほとんどのネットワークに適しています。

手順

ステップ 1 （CLI）TCP リセット動作を構成します。

service { **resetinbound** [**interface** *interface_name*] | **resetoutbound** [**interface** *interface_name*] | **resetoutside** }

- **resetinbound**を使用して無効にすることができます。ASA の通過を試み、アクセスリストまたはAAA 設定に基づいてASAによって拒否されたすべての着信TCPセッションにTCPリセットを送信します。ASA は、アクセスリストまたは AAA によって許可されても、既存の接続に属しておらず、ステートフルファイアウォールによって拒否されたパケットのリセットも送信します。同じセキュリティ レベルのインターフェイス間のトラフィックも影響を受けます。このオプションをイネーブルにしなかった場合、ASA は拒否されたパケットを何も通知せずに廃棄します。インターフェイスを指定しない場合、この設定はすべてのインターフェイスに適用されます。
- **resetoutbound**を使用して無効にすることができます。ASA の通過を試み、アクセスリストまたは AAA 設定に基づいて ASA によって拒否されたすべての発信 TCP セッションに TCP リセットを送信します。ASA は、アクセスリストまたは AAA によって許可されても、既存の接続に属しておらず、ステートフル ファイアウォールによって拒否されたパケットのリセットも送信します。同じセキュリティ レベルのインターフェイス間のトラフィックも影響を受けます。このオプションをイネーブルにしなかった場合、ASAは拒否されたパケットを何も通知せずに廃棄します。このオプションは、デフォルトで有効です。たとえば、トラフィック ストーム時に CPU の負荷を軽減するためなどに発信リセットをディセーブルにできます。
- **resetoutside**を使用して無効にすることができます。最もセキュリティレベルの低いインターフェイスで終端し、アクセスリストまたはAAA設定に基づいてASAによって拒否された TCP パケットのリセットをイネーブルにします。ASA は、アクセスリストまたは AAA によって許可されても、既存の接続に属しておらず、ステートフル ファイアウォールによって拒否されたパケットのリセットも送信します。このオプションをイネーブルにしなかった場合、ASA は拒否されたパケットを何も通知せずに廃棄します。

  インターフェイスPATでは、このオプションを使用することを推奨します。このオプションを使用すると、外部SMTPまたはFTPサーバーからのIDENTをASAで終端できます。これらの接続をアクティブにリセットすることによって、30秒のタイムアウト遅延を回避できます。

ステップ 2 通過トラフィックの最大 TCP セグメントサイズが設定した値を超えないようにし、指定したサイズ未満にならないようにするには、TCP MSS を設定します。

sysopt connection tcpmss [**minimum**] *bytes*

minimum キーワードなし。最大 TCP セグメントサイズをバイト単位で設定します（48 ～任意の最大値）。デフォルト値は 1380 バイトです。この機能をディセーブルにするには、bytes を 0 に設定します。

minimumを使用して無効にすることができます。最大セグメントサイズを上書きし、指定したバイト（48 ～ 65535 バイト）未満にならないようにします。この機能は、デフォルトでディセーブルです（0 に設定）。

ステップ 3 TCP 接続の確立待機時間を設定します。

sysopt connection timewait

このコマンドを使用すると、各 TCP 接続において、最後の通常の TCP クローズダウンシーケンスの後に、少なくとも 15 秒の短い TIME_WAIT 状態が強制的に維持されます。エンド ホスト アプリケーションのデフォルト TCP 終了シーケンスが同時クローズである場合に、この機能を使用することを推奨します。

ステップ 4 TCP 未処理セグメントの最大数を設定します。

sysopt connection tcp-max-unprocessed-seg *segments*

TCP 未処理セグメントの最大数を 6 ～ 24 に設定します。デフォルト値は 6 です。SIP 電話機が Call Manager に接続していないことを確認したら、未処理の TCP セグメントの最大数を増やすことができます。

接続のモニタリング

次のコマンドを使用して、接続をモニターできます。

- **show conn** [**detail**]

  接続情報を表示します。詳細情報は、フラグを使用して特別な接続の特性を示します。たとえば、「b」フラグは、TCPステート バイパスの対象であるトラフィックを示します。

  detail キーワードを使用すると、デッド接続検出（DCD）プローブの情報が表示されます。この情報は、発信側と応答側で接続がプローブされた頻度を示します。たとえば、DCD 対応接続の接続詳細は次のようになります。

```
TCP dmz: 10.5.4.11/5555 inside: 10.5.4.10/40299,
    flags UO , idle 1s, uptime 32m10s, timeout 1m0s, bytes 11828,
cluster sent/rcvd bytes 0/0, owners (0,255)
  Traffic received at interface dmz
        Locally received: 0 (0 byte/s)
  Traffic received at interface inside
        Locally received: 11828 (6 byte/s)
  Initiator: 10.5.4.10, Responder: 10.5.4.11
  DCD probes sent: Initiator 5, Responder 5
```

- **show flow-offload**{**info** [**detail**] | **cpu** | **flow** [**count** | **detail**] | **statistics**}

全般的なステータス情報、オフロードの CPU 使用率、オフロードされたフローの数と詳細、オフロードされたフロー統計情報を含む、フローのオフロードに関する情報を示します。

- **show service-policy**

  Dead Connection Detection（DCD; デッド接続検出）の統計情報を含むサービス ポリシーの統計情報を表示します。

- **show threat-detection statistics top tcp-intercept** [**all** | **detail**]

  攻撃を受けて保護された上位 10 サーバーを表示します。**all** キーワードは、トレースされているすべてのサーバーの履歴データを表示します。**detail** キーワードは、履歴サンプリング データを表示します。ASA はレート間隔の間に攻撃の数を 30 回サンプリングするので、デフォルトの 30 分間隔では、60 秒ごとに統計情報が収集されます。

（注） Cisco ASA 設定では、初期接続（3 ウェイ ハンドシェイク プロセスがまだ完了していない接続要求）はすぐに閉じられ、アクティブデバイスとスタンバイデバイス間で同期されません。この設計により、HA システムの効率とセキュリティが確保されます。このため、両方の Cisco ASA で接続数に違いが生じる可能性がありますが、これは予想されることです。

接続設定の履歴

機能名	プラットフォーム リリース	説明
TCP ステート バイパス	8.2(1)	この機能が導入されました。**set connection advanced-options tcp-state-bypass** コマンドが導入されました。
すべてのプロトコルの接続タイムアウト	8.2(2)	アイドル タイムアウトは、TCP だけでなく、すべてのプロトコルに適用するように変更されました。 **set connection timeout** コマンドが変更されました。
バックアップ スタティック ルートを使用する接続のタイムアウト	8.2(5)/8.4(2)	同じネットワークへの複数のスタティック ルートが存在しており、それぞれメトリックが異なる場合は、ASA は接続確立時点でメトリックが最良のルートを使用します。より適切なルートが使用可能になった場合は、このタイムアウトによって接続が閉じられるので、その適切なルートを使用して接続を再確立できます。デフォルトは 0 です（接続はタイムアウトしません）。この機能を使用するには、タイムアウトを新しい値に変更します。 **timeout floating-conn** コマンドが変更されました。

機能名	プラットフォーム リリース	説明
PAT xlate に対する設定可能なタイムアウト	8.4(3)	PAT xlate がタイムアウトし（デフォルトでは 30 秒後）、ASA が新しい変換用にポートを再使用すると、一部のアップストリーム ルータは、前の接続がアップストリーム デバイスで依然として開いている可能性があるため、この新しい接続を拒否する場合があります。PAT xlate のタイムアウトを、30 秒～ 5 分の範囲内の値に設定できるようになりました。 **timeout pat-xlate** コマンドが導入されました。 この機能は、*8.5(1)* または *8.6(1)* では使用できません。
サービス ポリシー ルールの最大接続数の引き上げ	9.0(1)	サービス ポリシー ルールの最大接続数が 65535 から 2000000 に引き上げられました。 **set connection conn-max**、**set connection embryonic-conn-max**、**set connection per-client-embryonic-max**、**set connection per-client-max** の各コマンドが変更されました。
ハーフ クローズ タイムアウト最小値を 30 秒に削減	9.1(2)	グローバル タイムアウトおよび接続タイムアウトの両方のハーフ クローズド タイムアウトの最小値は、より優れた DoS 保護を提供するために 5 分から 30 秒に短縮されました。 **set connection timeout half-closed**、**timeout half-closed** の各コマンドが変更されました。
ルートの収束に対する接続ホールドダウン タイムアウト。	9.4(3) 9.6(2)	接続で使用されているルートがもう存在していない、または非アクティブになったときに、システムが接続を保持する時間を設定できるようになりました。このホールドダウン期間内にルートがアクティブにならない場合、接続は解放されます。ルートの収束がさらに迅速に行われるようにホールドダウン タイマーを短縮することができます。ただし、ほとんどのネットワークでは、ルートのフラッピングを防止するためにデフォルトの 15 秒が適切です。 **timeout conn-holddown** コマンドが追加されました。
SCTP アイドル タイムアウトおよび SCTP ステート バイパス	9.5(2)	SCTP 接続のアイドル タイムアウトを設定できます。また、SCTP ステート バイパスを有効にして、トラフィックのクラスで SCTP ステートフル インスペクションをオフにできます。 次のコマンドが追加または変更されました。**timeout sctp**、**set connection advanced-options sctp-state-bypass**。

機能名	プラットフォーム リリース	説明
Firepower 9300 上の ASA のフローオフロード。	9.5(2.1)	ASA からオフロードされ、（Firepower 9300 上の）NIC に直接切り替えられる必要があるフローを特定できます。これにより、データセンターのより大きなデータ フローのパフォーマンスが向上します。 この機能には、FXOS 1.1.3 が必要です。 次のコマンドが追加または変更されました。**clear flow-offload**、**flow-offload enable**、**set-connection advanced-options flow-offload**、**show conn detail**、**show flow-offload**。
Firepower 4100 シリーズ 上の ASA のフロー オフロードのサポート。	9.6(1)	ASA からオフロードされ、Firepower 4100 シリーズ の NIC で直接切り替える必要があるフローを特定できます。 この機能では、FXOS 1.1.4 が必要です。 この機能には、新規のコマンドまたは ASDM 画面はありません。
トランスペアレント モードでのマルチキャスト接続のフローオフロードのサポート。	9.6(2)	トランスペアレント モードの Firepower 4100 および 9300 シリーズ デバイスで、NIC に直接切り替えられるマルチキャスト接続をオフロードできるようになりました。マルチキャスト オフロードは、インターフェイスを 2 つだけ含むブリッジ グループに使用できます。 この機能には、新規のコマンドまたは ASDM 画面はありません。

機能名	プラットフォーム リリース	説明
TCP オプション処理の変更。	9.6(2)	TCP マップを設定する際にパケットの TCP ヘッダー内の TCP MSS および MD5 オプションに対するアクションを指定できるようになりました。さらに、MSS、タイムスタンプ、ウィンドウ サイズ、および選択的確認応答オプションのデフォルトの処理が変更されました。以前は、これらのオプションは、ヘッダーに特定のタイプのオプションが 2 つ以上ある場合でも許可されていました。現在は、パケットに特定のタイプのオプションが 2 つ以上含まれている場合、そのパケットはデフォルトでドロップされます。たとえば、以前は 2 つのタイムスタンプ オプションがあるパケットは許可されていましたが、現在はドロップされます。 MD5、MSS、選択的確認応答、タイムスタンプ、およびウィンドウ サイズに対し、同じタイプの複数のオプションを有効にするための TCP マップを設定できます。MD5 オプションの場合、以前のデフォルトではオプションがクリアされたのに対し、現在のデフォルトでは許可されます。また、MD5 オプションを含むパケットをドロップすることもできます。MSS オプションの場合は、TCP マップで最大セグメント サイズを設定できます（トラフィック クラスごとに）。他のすべての TCP オプションのデフォルトに変更はありません。これらはクリアされます。 次のコマンドが変更されました。**timeout igp stale-route**。
内部ゲートウェイ プロトコルの古いルートのタイムアウト	9.7(1)	OSPF などの内部ゲートウェイ プロトコルの古いルートを削除するためのタイムアウトを設定できるようになりました。 **timeout igp stale-route** コマンドが追加されました。
ICMP エラーのグローバル タイムアウト	9.8(1)	ASA が ICMP エコー応答パケットを受信してから ICMP 接続を削除するまでのアイドル時間を設定できるようになりました。このタイムアウトが無効（デフォルト）で、ICMP インスペクションが有効に設定されている場合、ASA はエコー応答を受信するとすぐに ICMP 接続を削除します。したがって、終了しているその接続に対して生成されたすべての ICMP エラーは破棄されます。このタイムアウトは ICMP 接続の削除を遅らせるので、重要な ICMP エラーを受信することが可能になります。 次のコマンドが追加されました。 **timeout icmp-error**
TCP ステート バイパスのデフォルトのアイドル タイムアウト	9.10(1)	TCP ステート バイパス接続のデフォルトのアイドル タイムアウトは 1 時間ではなく、2 分になりました。

機能名	プラットフォーム リリース	説明
デッド接続検出（DCD）の発信側および応答側の情報、およびクラスタ内の DCD のサポート。	9.13(1)	デッド接続検出（DCD）を有効にした場合は、**show conn detail** コマンドを使用して発信側と応答側に関する情報を取得できます。デッド接続検出を使用すると、非アクティブな接続を維持できます。**show conn** の出力は、エンドポイントがプローブされた頻度が示されます。さらに、DCD がクラスタでサポートされるようになりました。 新しい/変更されたコマンド：**show conn**（出力のみ）
初期接続の最大セグメントサイズ（MSS）を設定します。	9.16(1)	サービスポリシーを設定して、初期接続制限に達したときに初期接続の SYN cookie を生成するためのサーバーの最大セグメントサイズ（MSS）を設定できます。これは、最大初期接続数も設定するサービスポリシーの場合に意味があります。 追加または変更されたコマンド：**set connection syn-cookie-mss**。
IPsec フローがオフロードされます。	9.18(1)	Cisco Secure Firewall 3100 では、IPsec フローはデフォルトでオフロードされます。IPsec サイト間 VPN またはリモートアクセス VPN セキュリティ アソシエーション (SA) の初期設定後、IPsec 接続はデバイスのフィールドプログラマブル ゲートアレイ (FPGA) にオフロードされるため、デバイスのパフォーマンスが向上します。 次のコマンドが追加されました。**clear flow-offload-ipsec**、**flow-offload-ipsec**、**show flow-offload-ipsec**
DTLS 暗号化アクセラレーション	9.22(1)	Cisco Secure Firewall 4200 および 3100 シリーズは、DTLS 暗号化アクセラレーションをサポートします。ハードウェアはDTLS 暗号化と復号化を実行し、DTLS 暗号化トラフィックと DTLS 復号化トラフィックのスループットを向上させます。ハードウェアは、遅延を改善するために、出力暗号化パケットの最適化も実行します。 新規/変更されたコマンド：**flow-offload-dtls**、**flow-offload-dtls egress-optimization**
フロー オフロードは Secure Firewall 3100/4200 に対してデフォルトで有効です	9.23(1)	フロー オフロードはデフォルトで有効です。 追加/変更されたコマンド： **flow-offload enable**

機能名	プラットフォーム リリース	説明
クラスタ リダイレクト： Cisco Secure Firewall 4200 非対称クラスタ トラフィックのフロー オフロードのサポート	9.23(1)	非対称フローの場合、クラスタリダイレクトにより、転送ノードはハードウェアにフローをオフロードできます。この機能はデフォルトで有効になっています。 既存のフローのトラフィックが別のノードに送信されると、そのトラフィックはクラスタ制御リンクを介してオーナー ノードにリダイレクトされます。非対称フローは、クラスタ制御リンクに大量のトラフィックを作成する可能性があるため、フォワーダにこれらのフローをオフロードさせると、パフォーマンスが向上します。 追加/変更されたコマンド： **flow-offload cluster-redirectshow connshow flow-offload flowshow flow-offload flow protocolshow flow-offload info**
分散型サイト間 VPN モードの Cisco Secure Firewall 4200 のクラスタ制御リンクのトラフィックの IPsec フローのオフロード	9.23(1)	分散型サイト間 VPN モードの非対称フローの場合、IPsec フロー オフロードにより、フロー オーナーは、クラスタ制御リンクを介して転送されたハードウェア内の IPsec トラフィックを復号できます。この機能は構成可能ではありません。IPsec フローのオフロードを有効にすると常に使用できます。 追加/変更されたコマンド： **flow-offload-ipsec**、**show crypto ipsec sa detail**

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