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。接続の設定には、次のものが含まれます。

- 最大接続数（TCP および UDP 接続、初期接続、クライアントあたりの接続）
- 接続タイムアウト
- デッド接続検出
- TCP シーケンスのランダム化
- TCP 正規化カスタマイゼーション
- TCP ステート バイパス
- グローバル タイムアウト
- 「接続の設定に関する情報」（P.12-1）
- 「接続設定のライセンス要件」（P.12-5）
- 「ガイドラインと制限事項」（P.12-5）
- 「デフォルト設定」（P.12-6）
- 「接続の設定」（P.12-6）
- 「接続設定のモニタリング」（P.12-14）
- 「接続設定の設定例」（P.12-14）
- 「接続設定の機能履歴」（P.12-16）

接続の設定に関する情報

この項では、接続の制限が必要になる理由を示します。

- 「TCP 代行受信および初期接続の制限」（P.12-2）
- 「クライアントレス SSL 互換での管理パケットの TCP 代行受信のディセーブル化」（P.12-2）
- 「デッド接続検出（DCD）」（P.12-2）
- 「TCP シーケンスのランダム化」（P.12-3）
- 「TCP の正規化」（P.12-3）
- 「TCP ステート バイパス」（P.12-4）

TCP 代行受信および初期接続の制限

初期接続の数を制限することで、DoS 攻撃（サービス拒絶攻撃）から保護されます。ASA では、クライアントあたりの制限値と初期接続の制限を利用して TCP 代行受信を開始します。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。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。TCP 代行受信では、SYN クッキー アルゴリズムを使用して TCP SYN フラッディング攻撃を防ぎます。SYN フラッディング攻撃は、通常はスプーフィングされた IP アドレスから送信されてくる一連の SYN パケットで構成されています。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、接続要求に対してサービスを提供できなくなります。接続の初期接続しきい値を超えると、ASA はサーバのプロキシとして動作し、クライアント SYN 要求に対する SYN-ACK 応答を生成します。ASA がクライアントから ACK を受信すると、クライアントを認証し、サーバへの接続を許可できます。

（注） TCP SYN クッキー保護を使用して SYN 攻撃からサーバを保護する場合、保護するサーバの TCP SYN バックログ キューより低い初期接続制限を設定する必要があります。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 攻撃中にサーバにアクセスできなくなります。

TCP 代行受信に関する統計情報（攻撃を受けた上位 10 サーバなど）を表示する方法については、第 16 章「脅威の検出」を参照してください。

クライアントレス SSL 互換での管理パケットの TCP 代行受信のディセーブル化

デフォルトでは、TCP 管理接続では TCP 代行受信が常にイネーブルになっています。TCP 代行受信をイネーブルにすると、3 ウェイ TCP 接続確立のハンドシェイク パケットが代行受信されるため、ASA ではクライアントレス SSL のパケットを処理できなくなります。クライアントレス SSL では、クライアントレス SSL 接続で selective-ack や他の TCP オプションを提供するために、3 ウェイ ハンドシェイク パケットを処理する機能が必要になります。管理トラフィックの TCP 代行受信をディセーブルにするには、初期接続制限を設定します。初期接続制限に達した後にだけ TCP 代行受信をイネーブルにできます。

デッド接続検出（DCD）

DCD では、デッド接続を検出して、トラフィックをまだ処理できる接続を期限切れにすることなく、そのデッド接続を期限切れにすることができます。DCD は、アイドル状態でも有効な接続を維持する場合に設定します。

DCD をイネーブルにすると、アイドル タイムアウト動作が変化します。アイドル タイムアウトになると、DCD プローブが 2 つのエンドホストそれぞれに送信され、接続の有効性が判断されます。設定された間隔でプローブが送信された後にエンドホストが応答を返さないと、その接続は解放され、リセット値が設定されていれば各エンドホストに送信されます。両方のエンドホストが応答して接続の有効性が確認されると、アクティビティ タイムアウトは現在時刻に更新され、それに応じてアイドル タイムアウトが再スケジュールされます。

DCD をイネーブルにすると、TCP ノーマライザでのアイドルタイムアウト処理の動作が変更されます。DCD プローブにより、**show conn** コマンドで表示される接続でのアイドル タイムアウトがリセットされます。タイムアウト コマンドで設定したタイムアウト値を超過していても、DCD プローブのために存続している接続を判別するため、**show service-policy** コマンドには、DCD からのアクティビティ数を示すカウンタが含まれています。

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 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 つはサーバで生成されます。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。

保護されたホストの ISN をランダム化することにより、攻撃者が新しい接続で次の ISN を予測できないようにして、新規セッションが乗っ取られるのを防ぎます。

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。
- 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。

TCP の正規化

TCP 正規化機能は、検出時に ASA が対処できる異常なパケットを識別します。ASA は、パケットを許可、ドロップ、またはクリアできます。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。

TCP 正規化には、設定できないアクションと設定できるアクションが含まれます。通常、接続をドロップまたはクリアする設定できないアクションは、どのような場合でも不良なパケットに適用されます。設定できるアクション（「TCP マップを使用した TCP ノーマライザのカスタマイズ」（P.12-6）を参照）は、ネットワークのニーズに応じたカスタマイズが必要な場合があります。

TCP 正規化に関する次のガイドラインを参考にしてください。

- ノーマライザは、SYN フラッドからの保護は行いません。ASA には、他の方法による SYN フラッド保護機能が組み込まれています。
- ノーマライザは、ASA がフェールオーバーのためにルーズ モードになっていない限り、SYN パケットを最初のパケットと見なします。

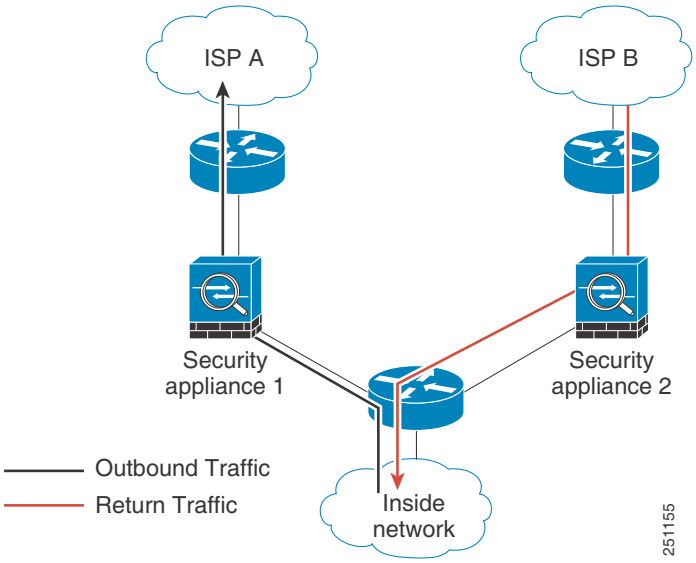
TCP ステート バイパス

デフォルトでは、ASA を通過するトラフィックはすべて、アダプティブ 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、通過を許可されるか、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）、ファスト 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。ステートフル ファイアウォールの詳細については、一般的な操作のコンフィギュレーション ガイドを参照してください。

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 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 ASA 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、接続のエントリが高速パス テーブルに追加されます。この接続の後続のパケットが ASA 1 を通過する場合、パケットは高速パスのエントリと一致して、通過します。しかし、後続のパケットが ASA 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。図 12-1 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 ASA を通過しています。

図 12-1　非対称ルーティング



アップストリーム ルータに設定された非対称ルーティングがあり、トラフィックが 2 つの ASA の間で切り替わる場合は、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ディセーブルにします。この機能は、TCP トラフィックを UDP 接続と同じように処理します。指定されているネットワークに一致する非 SYN パケットが ASA に到着し、高速パスのエントリがない場合は、パケットはセッション管理パスを通過して、高速パスの接続を確立します。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。

接続設定のライセンス要件

モデル	ライセンス要件
ASAv	標準または Premium ライセンス
他のすべてのモデル	基本ライセンス

ガイドラインと制限事項

コンテキスト モードのガイドライン

シングル コンテキスト モードとマルチ コンテキスト モードでサポートされています。

ファイアウォール モードのガイドライン

ルーテッド モードとトランスペアレント モードでサポートされます。

フェールオーバーのガイドライン

フェールオーバーはサポートされます。

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：アプリケーション インスペクションではインバウンド トラフィックとアウトバウンド トラフィックの両方が同じ ASA を通過する必要があるので、アプリケーション インスペクションは TCP ステート バイパスではサポートされません。
- AAA 認証済みセッション：ユーザが 1 つの ASA で認証するとき、他の ASA を介して返されるトラフィックは、ユーザがその ASA で認証を受けていないので拒否されます。
- TCP 代行受信、最大初期接続制限、TCP シーケンス番号のランダム化：ASA は接続のステートを追跡しないので、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になります。
- SSM および SSC 機能：TCP ステート バイパスおよび IPS や CSC などの SSM または SSC 上で実行するアプリケーションは使用できません。

TCP ステート バイパスの NAT 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ので、TCP ステート バイパス トラフィック用に両方のASAでスタティック NAT を設定してください。ダイナミック NAT を使用すると、ASA 1 でのセッションに選択されるアドレスが、ASA 2 でのセッションに選択されるアドレスと異なります。

最大同時接続および初期接続のガイドライン

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。

デフォルト設定

TCP ステート バイパス

TCP ステート バイパスは、デフォルトでディセーブルになっています。

TCP ノーマライザ

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

接続の設定

接続の設定のタスク フロー

ステップ 1　TCP 正規化カスタマイゼーションについては、「TCP マップを使用した TCP ノーマライザのカスタマイズ」（P.12-6）に従って TCP マップを作成します。

ステップ 2　すべての接続設定については、第 1 章「モジュラ ポリシー フレームワークを使用したサービス ポリシー」に従ってサービス ポリシーを設定します。

ステップ 3　「接続の設定」（P.12-11）に従って接続を設定します。

TCP マップを使用した TCP ノーマライザのカスタマイズ

TCP ノーマライザをカスタマイズするには、まず、TCP マップを使用する設定を定義します。

手順の詳細

ステップ 1　検索する TCP 正規化基準を指定するには、次のコマンドを入力して TCP マップを作成します。

```
hostname(config)# tcp-map tcp-map-name
```

TCP マップごとに 1 つまたは複数の設定値をカスタマイズできます。

ステップ 2　（任意）次の 1 つ以上のコマンド（表 12-1 を参照）を入力して TCP マップ基準を設定します。一部の設定をカスタマイズする場合、入力しないコマンドにはデフォルトが使用されます。

表 12-1　tcp-map コマンド

コマンド	注
check-retransmission	一貫性のない TCP 再送信を防止します。
checksum-verification	チェックサムを確認します。
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に対するアクションを設定します。 （デフォルト）**allow** キーワードは、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します。 **drop** キーワードは、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ドロップします。
invalid-ack {**allow** | **drop**}	無効な ACK を含むパケットに対するアクションを設定します。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。 **allow** キーワードは、無効な ACK を含むパケットを許可します。 （デフォルト）**drop** キーワードは、無効な ACK を含むパケットをドロップします。 **（注）**　無効な ACK を含む TCP パケットは、WAAS 接続で自動的に許可されます。

表 12-1　tcp-map コマンド（続き）

コマンド	注
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）、IPS（**ips** 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、IPS、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。 （デフォルト）**allow** キーワードは、TCP ヘッダーの予約ビットが設定されているパケットを許可します。 **clear** キーワードは、TCP ヘッダーの予約ビットを消去して、パケットを許可します。 **drop** キーワードは、TCP ヘッダーの予約ビットが設定されているパケットをドロップします。

表 12-1　tcp-map コマンド（続き）

コマンド	注
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。 **allow** キーワードは、パストウィンドウ シーケンス番号を含むパケットを許可します。このアクションは、**queue-limit** コマンドが 0（ディセーブル）に設定されている場合に限り許可されます。 （デフォルト）**drop** キーワードは、パストウィンドウ シーケンス番号を含むパケットをドロップします。
synack-data {**allow** | **drop**}	データを含む TCP SYNACK パケットに対するアクションを設定します。 **allow** キーワードは、データを含む TCP SYNACK パケットを許可します。 （デフォルト）**drop** キーワードは、データを含む TCP SYNACK パケットをドロップします。
syn-data {**allow** | **drop**}	データを含む SYN パケットに対するアクションを設定します。 （デフォルト）**allow** キーワードは、データを含む SYN パケットを許可します。 **drop** キーワードは、データを含む SYN パケットをドロップします。
tcp-options {**selective-ack** | **timestamp** | **window-scale**} {**allow** | **clear**} または **tcp-options range** *lower upper* {**allow** | **clear** | **drop**}	selective-ack、timestamp、window-scale などの TCP オプションを含むパケットに対するアクションを設定します。 （デフォルト）**allow** キーワードは、指定したオプションを含むパケットを許可します。 （**range** の場合のデフォルト）**clear** キーワードは、オプションを消去して、パケットを許可します。 **drop** キーワードは、指定したオプションを含むパケットをドロップします。 **selective-ack** キーワードは、SACK オプションに対するアクションを設定します。 **timestamp** キーワードは、タイムスタンプ オプションに対するアクションを設定します。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。 **widow-scale** キーワードは、ウィンドウ スケール メカニズム オプションに対するアクションを設定します。 **range** キーワードは、オプションの範囲を指定します。*lower* 引数は、範囲の下限を設定します。6、7、または 9 ～ 255 です。 *upper* 引数は、範囲の上限を設定します。6、7、または 9 ～ 255 です。

表 12-1　tcp-map コマンド（続き）

コマンド	注
ttl-evasion-protection	TTL 回避保護をディセーブルにします。セキュリティ ポリシーを回避しようとする攻撃を防ぐ場合は、このコマンドを入力しないでください。 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。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。 **allow** キーワードは、URG フラグを含むパケットを許可します。 （デフォルト）**clear** キーワードは、URG フラグを消去してパケットを許可します。
window-variation {**allow** | **drop**}	予想外のウィンドウ サイズの変更が発生した接続に対するアクションを設定します。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。 （デフォルト）**allow** キーワードは、ウィンドウが変化した接続を許可します。 **drop** キーワードは、ウィンドウが変化した接続をドロップします。

接続の設定

接続を設定するには、次の手順を実行します。

手順の詳細

	コマンド	目的
ステップ 1	**class-map** *name* **例：** hostname(config)# class-map bypass_traffic	ステートフル ファイアウォール インスペクションをディセーブルにするトラフィックを識別するためのクラス マップを作成します。
ステップ 2	**match** *parameter* **例：** hostname(config-cmap)# match access-list bypass	クラス マップのトラフィックを指定します。詳細については、「トラフィックの特定（レイヤ 3/4 クラス マップ）」（P.1-14）を参照してください。
ステップ 3	**policy-map** *name* **例：** hostname(config)# policy-map tcp_bypass_policy	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ます。
ステップ 4	**class** *name* **例：** hostname(config-pmap)# class bypass_traffic	ステップ 1 で作成したクラス マップを識別します。
ステップ 5	次のいずれかまたは複数の作業を実行します。	

コマンド	目的
set connection {[**conn-max** *n*] [**embryonic-conn-max** *n*] [**per-client-embryonic-max** *n*] [**per-client-max** *n*] [**random-sequence-number** {**enable** | **disable**}]} **例：** `hostname(config-pmap-c)# set connection conn-max 256 random-sequence-number disable`	最大接続数を設定するか、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します。 **conn-max** *n* 引数には、許可される同時 TCP/UDP 接続の最大数を 0 ～ 2000000 の範囲で設定します。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 または UDP の同時接続を許可するように 2 つのサーバ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に別々に適用されます。 クラスに設定された場合、この引数では、クラス全体で許可される同時接続最大数が制限されます。この場合、1 つの攻撃ホストがすべての接続を使い果たし、クラスにおいて ACL 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。 **embryonic-conn-max** *n* 引数には、許可される同時初期接続の最大数を 0 ～ 2000000 の範囲で設定します。デフォルトは 0 で、この場合は接続数が制限されません。 **per-client-embryonic-max** *n* 引数には、クライアントごとに許可される同時初期接続の最大数を 0 ～ 2000000 の範囲で設定します。デフォルトは 0 で、この場合は接続数が制限されません。 **per-client-max** *n* 引数には、クライアントごとに許可される同時接続の最大数を 0 ～ 2000000 の範囲で設定します。デフォルトは 0 で、この場合は接続数が制限されません。クラスに設定された場合、この引数では、クラスにおいて ACL 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。 **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。詳細については、「TCP シーケンスのランダム化」（P.12-3）を参照してください。 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。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。 **（注）** 管理トラフィックの場合は、**conn-max** キーワードと **embryonic-conn-max** キーワードだけを設定できます。

コマンド	目的
set connection timeout {[**embryonic** *hh*:*mm*:*ss*] {**idle** *hh*:*mm*:*ss* [**reset**]] [**half-closed** *hh*:*mm*:*ss*] [dcd *hh*:*mm*:*ss* [*max_retries*]]} **例：** hostname(config-pmap-c)# set connection timeout idle 2:0:0 embryonic 0:40:0 half-closed 0:20:0 dcd	接続タイムアウトを設定します。グローバル タイムアウトについては、コマンド リファレンスの **timeout** コマンドを参照してください。次に説明するデフォルト値は、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。 **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を 0:0:5 ～ 1193:00:00 の間で設定します。デフォルトは 0:0:30 です。この値を 0 に設置することもでき、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。 **idle** *hh*:*mm*:*ss* キーワードは、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 タイムアウト期間を 0:0:1 から 1193:0:0 の間で設定します。デフォルトは 1:0:0 です。この値を 0 に設置することもでき、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。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。 The **half-closed** *hh*:*mm*:*ss* キーワードは、ハーフ クローズ接続が閉じられるまでのアイドル タイムアウト期間を 0:5:0（9.1(1) 以前の場合）または 0:0:30（9.1(2) 以降の場合）から 1193:0:0 の間で設定します。デフォルトは 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 パケットを送信しません。 **dcd** キーワードは、DCD をイネーブルにします。DCD では、デッド接続を検出して、トラフィックをまだ処理できる接続を期限切れにすることなく、そのデッド接続を期限切れにすることができます。DCD は、アイドル状態でも有効な接続を維持する場合に設定します。TCP 接続がタイムアウトすると、ASA は、エンドホストに DCD プローブを送信して接続の有効性を判断します。最大再試行回数を超えてもエンドホストの一方が応答しない場合、ASA はその接続を解放します。両方のエンドホストが応答して接続の有効性が確認されると、ASA はアクティビティ タイムアウトを現在時刻に更新し、それに応じてアイドル タイムアウトを再スケジュールします。*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 です。 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。 タイムアウトにならないようにするには、0:0:0 を入力します。 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。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。このコマンドは、管理トラフィックでは使用できません。

	コマンド	目的
	set connection advanced-options *tcp-map-name* **例：** hostname(config-pmap-c)# set connection advanced-options tcp_map1	TCP ノーマライザをカスタマイズします。TCP マップを作成するには、「TCP マップを使用した TCP ノーマライザのカスタマイズ」（P.12-6）を参照してください。
	set connection advanced-options tcp-state-bypass **例：** hostname(config-pmap-c)# set connection advanced-options tcp-state-bypass	TCP ステート バイパスをイネーブルにします。
ステップ 6	**service-policy** *policymap_name* {**global** | **interface** *interface_name*} **例：** hostname(config)# service-policy tcp_bypass_policy outside	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。**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は、ポリシー マップを 1 つだけ適用できます。

接続設定のモニタリング

TCP ステート バイパスをモニタするには、次のいずれかのタスクを実行します。

コマンド	目的
show conn	**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。

接続設定の設定例

- 「接続の制限値とタイムアウトの設定例」（P.12-15）
- 「TCP ステート バイパスの設定例」（P.12-15）
- 「TCP 正規化の設定例」（P.12-15）

接続の制限値とタイムアウトの設定例

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 ステート バイパスの設定例

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 outside

hostname(config-pmap-c)# static (inside,outside) 209.165.200.224 10.1.1.0 netmask
255.255.255.224
```

TCP 正規化の設定例

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
```

接続設定の機能履歴

表 12-2 に、各機能変更と、それが実装されたプラットフォーム リリースを示します。

表 12-2　　接続設定の機能履歴

機能名	プラットフォーム リリース	機能情報
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。
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。*

表 12-2　接続設定の機能履歴（続き）

機能名	プラットフォームリリース	機能情報
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。