ネットワーク攻撃の防止

この章では、TCP 正規化の設定、TCP 接続と UDP 接続の制限、およびその他のさまざまな保護機能によるネットワーク攻撃の防止方法について説明します。

次の事項について説明します。

TCP の正規化の設定

TCP の正規化機能を使用して、異常なパケットを特定する基準を指定できます。このような異常なパケットは、検出時にセキュリティ アプライアンスによってドロップされます。この機能はモジュラ ポリシー フレームワークを使用するため、TCP の正規化の実装は、トラフィックの特定、TCP の正規化基準の指定、およびインターフェイスでの TCP の正規化のアクティブ化で構成されます。詳細については、第 21 章「モジュラ ポリシー フレームワークの使用」を参照してください。

TCP の正規化を設定するには、次の手順を実行します。

ステップ 1　検索する TCP の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指定できます。

ステップ 2　次の 1 つ以上のオプションについて、コマンドを入力して TCP マップ基準を設定します。

- 一貫性のない TCP 再送信を防止する。
  ```
  hostname(config-tcp-map)# check-retransmission
  ```
- チェックサムを確認する。
  ```
  hostname(config-tcp-map)# checksum-verification
  ```
- 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する。デフォルトでは、このようなパケットはドロップされるため、次のコマンドで許可します。
  ```
  hostname(config-tcp-map)# exceed-mss {allow | drop}
  ```
- TCP 接続のキューに入れることができる異常なパケットの最大数を設定する。
  ```
  hostname(config-tcp-map)# queue-limit pkt_num
  ```
  pkt_num には、異常なパケットの最大数を指定します。範囲は 0 ～ 250 で、デフォルトは 0 です。
- TCP　ヘッダーの予約ビットを消去するか、または予約ビットが設定されているパケットをドロップする。デフォルトでは、予約ビットは許可されます。そのため、次のコマンドで予約ビットを消去するか、パケットをドロップします。
  ```
  hostname(config-tcp-map)# reserved-bits {allow | clear | drop}
  ```
  allow 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** 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** 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。
- データが設定されている SYN パケットをドロップする。デフォルトでは、データが設定されている SYN パケットは許可されます。そのため、次のコマンドでパケットをドロップします。
  ```
  hostname(config-tcp-map)# syn-data {allow | drop}
  ```
- selective-ack、timestamp、window-scale の各 TCP オプションを消去するか、または番号で指定された TCP オプションの範囲をドロップする。デフォルトでは、指定されたオプションが設定されているパケットは許可され、範囲内のオプションは消去されます。そのため、次のコマンドでそれらを消去、許可、またはドロップします。
  ```
  hostname(config-tcp-map)# tcp-options {selective-ack | timestamp | window-scale}
  {allow | clear}
  ```
  または
  ```
  hostname(config-tcp-map)# tcp-options range lower upper {allow | clear | drop}
  ```

allow は、指定されたオプションが設定されているパケットを許可します。**clear** は、オプションを消去してパケットを許可します。**drop** は、パケットをドロップします。

selective-ack キーワードは、SACK オプションを許可または消去します。デフォルトでは、SACK オプションは許可されます。

timestamp キーワードは、タイムスタンプ オプションを許可または消去します。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。デフォルトでは、タイムスタンプ オプションは許可されます。

widow-scale キーワードは、ウィンドウ スケール メカニズム オプションを許可または消去します。デフォルトでは、ウィンドウ スケール メカニズム オプションは許可されます。

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。

- TTL 回避保護をディセーブルにする。

```
hostname(config-tcp-map)# ttl-evasion-protection
```

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、セキュリティ アプライアンス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、セキュリティ アプライアンスには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。

- URG ポインタを許可する。

```
hostname(config-tcp-map)# urgent-flag {allow | clear}
```

URG フラグは、ストリーム中の他のデータよりも優先順位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。デフォルトの動作は、URG フラグおよびオフセットの消去です。URB フラグを許可するには、このコマンドを使用します。

- 予想外のウィンドウ サイズの変更が発生した接続をドロップする。デフォルトでは、このような接続は許可されるため、次のコマンドでドロップします。

```
hostname(config-tcp-map)# window-variation {allow | drop}
```

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　TCP 正規化を適用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、P.21-4 の「レイヤ 3/4 クラスマップによるトラフィックの特定」を参照してください。

ステップ 4　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するには、次のコマンドを入力します。

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

ステップ 5　アクションを割り当てるステップ 1 のクラスマップを特定するには、次のコマンドを入力します。

```
hostname(config-pmap)# class class_map_name
```

ステップ 6　次のコマンドを入力して、TCP マップをクラスマップに適用します。

```
hostname(config-pmap-c)# set connection advanced-options tcp-map-name
```

ステップ 7　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}
```

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は、ポリシーマップを 1 つだけ適用できます。

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
```

接続の制限値とタイムアウトの設定

この項では、TCP と UDP の最大接続数、最大初期接続数、クライアントごとの最大接続数、接続タイムアウトを設定する方法、デッド接続検出、および 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。

接続と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。セキュリティ アプライアンスでは、クライアントごとの制限値と初期接続制限値を利用して TCP 代行受信を発生させます。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。

Dead connection detection（DCD; デッド接続検出）はデッド接続を検出し、トラフィックをまだ処理できる接続の期限を切ることなく、デッド接続の期限を切ることができます。DCD タイムアウトがクラスに対して設定されていると、DCD はそのクラスと一致するトラフィックに対してイネーブルになります。DCD タイムアウトが設定されていないと、DCD はそのクラスと一致するトラフィックに対してディセーブルになります。アイドルにする場合に DCD を設定できますが、有効な接続は継続されます。

DCD をイネーブルにすると、アイドルのタイムアウト動作が変化します。アイドル タイムアウトを使用すると、DCD プローブが 2 つのエンドホストに送信され、接続の有効性を判別します。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できない場合、その接続は解放されます。リセット値が設定された場合、各エンドホストに送信されます。両方のエンドホスト により接続が有効であると応答されると、アクティビティ タイムアウトが現在の時間に更新され、タイムアウトがそれに応じて再度スケジュールされます。

TCP シーケンスのランダム化をディセーブルにするのは、別のインライン ファイアウォールもシーケンス番号をランダム化していて、結果としてデータ順序が変わる場合だけにします。各 TCP 接続には、2 つの Initial Sequence Number（ISN; 初期シーケンス番号）があります。1 つはクライアントが生成し、1 つはサーバが生成します。セキュリティ アプライアンスはクライアントおよびホスト / サーバの両方が生成する ISN をランダム化します。少なくとも 1 つの ISN をランダムに生成して、攻撃者が次の ISN を予想してセッションを乗っ取ることができないようにする必要があります。

（注） 最大接続数、最大初期接続数、および TCP シーケンスのランダム化は、NAT コンフィギュレーションでも設定できます。同じトラフィックに対して両方の方法でこれらの設定値を設定した場合、セキュリティ アプライアンスは小さい方の制限値を使用します。TCP シーケンスのランダム化がいずれかの方法でディセーブルにされている場合、セキュリティ アプライアンスは TCP シーケンスのランダム化をディセーブルにします。

接続制限値を設定するには、次の手順を実行します。

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、P.21-4 の「レイヤ 3/4 クラスマップによるトラフィックの特定」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するには、次のコマンドを入力します。

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

ステップ 3　アクションを割り当てるステップ 1 のクラスマップを特定するには、次のコマンドを入力します。

```
hostname(config-pmap)# class class_map_name
```

ステップ 4　最大接続制限値または TCP シーケンスのランダム化のイネーブル / ディセーブルを設定するには、次のコマンドを入力します。

```
hostname(config-pmap-c)# set connection {conn-max number | embryonic-conn-max number |
per-client-embryonic-max number | per-client-max number | random-sequence-number
{enable | disable}}. . .
```

number は、0 ～ 65535 の整数です。デフォルトは 0 で、接続を制限しないことを意味します。

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）、各アトリビュートを別々のコマンドとして入力することもできます。セキュリティ アプライアンスにより、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。

ステップ 5　接続、初期接続（ハーフオープン接続）、およびハーフクローズ接続、およびデッド接続検出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 {tcp <value> [reset]] [half-close
<value>] [embryonic <value>] [dcd [<retry-interval> [max-retries]]]}
```

half-close および **tcp** 値は 0:5:0 ～ 1192:59:59 間の時間で、*hh*:*mm*:*ss* の形式になります。**half-close** のデフォルトは 0:10:0 で、**tcp** のデフォルトは 1:0:0 です。また、これらの値を 0 に設定することもできます。この場合、接続はタイムアウトになりません。

embryonic <*value*> は 0:0:5 と 1192:59:59 の間の時間で、*hh*:*mm*:*ss* の形式になります。デフォルトは 0:0:30 です。また、この値を 0 に設定することもできます。この場合、接続はタイムアウトになりません。

dcd <*retry-interval*> は、<hh:mm:ss> 形式の各未応答 DCD プローブ間で待機する時間です。最小値は 1 秒で、最大値は 24 時間です。デフォルト値は、15 秒です。

dcd <*max-retries*> は、接続が「デッド」であると宣言される前に連続して失敗したリトライの数です。最小値は 1 秒で、最大値は 255 です。デフォルトは 5 です。

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）、各アトリビュート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。

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}
```

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は、ポリシーマップを 1 つだけ適用できます。

IP スプーフィングの防止

この項では、インターフェイスで Unicast Reverse Path Forwarding（Unicast RPF; ユニキャスト逆経路転送）をイネーブルにします。Unicast RPF は、ルーティング テーブルに従い、すべてのパケットが正しい発信元インターフェイスと一致する送信元 IP アドレスを持っていることを確認して、IP スプーフィング（パケットが不正な送信元 IP アドレスを使用し、実際の送信元を隠蔽すること）から保護します。

通常、セキュリティ アプライアンスは、パケットの転送先を判定するときに、宛先アドレスだけを調べます。Unicast RPF は、送信元アドレスも調べるようにセキュリティ アプライアンスに指示します。そのため、逆経路転送（Reverse Path Forwarding）と呼ばれます。セキュリティ アプライアンスの通過を許可するすべてのトラフィックについて、送信元アドレスに戻るルートをセキュリティ アプライアンスのルーティング テーブルに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、セキュリティ アプライアンスはデフォルト ルートを使用して Unicast RPF 保護を満たすことができます。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 テーブルにない場合、セキュリティ アプライアンスはデフォルト ルートを使用して、外部インターフェイスを発信元インターフェイスとして正しく識別します。

ルーティング テーブルにあるアドレスから外部インターフェイスにトラフィックが入り、このアドレスが内部インターフェイスに関連付けられている場合、セキュリティ アプライアンスはパケットをドロップします。同様に、未知の送信元アドレスから内部インターフェイスにトラフィックが入った場合は、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、セキュリティ アプライアンスはパケットをドロップします。

Unicast RPF は、次のように実装されます。

- ICMP パケットにはセッションがないため、個々のパケットはチェックされません。
- UDP と TCP にはセッションがあるため、最初のパケットは逆ルート ルックアップが必要です。セッション中に到着する後続のパケットは、セッションの一部として保持されている既存の状態を使用してチェックされます。最初のパケット以外のパケットは、最初のパケットと同じインターフェイスに到着したことを保証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、セキュリティ アプライアンスは 1 つの IP パケットにつき最大 24 のフラグメントを許可し、最大 200 のフラグメントのリアセンブリ待ちを許可します。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、フラグメントがセキュリティ アプライアンス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。

不要な接続のブロック

ある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た場合（たとえば、システム ログ メッセージで攻撃が示された場合）、送信元 IP アドレスおよびその他の識別パラメータに基づいて、接続をブロック（排除）できます。排除を無効するまで、新しい接続は作成できません。

（注）　トラフィックを監視する IPS（AIP SSM など）がある場合は、IPS で自動的に接続を排除できます。

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
```

セキュリティ アプライアンスは、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]
```

送信元 IP アドレスだけを入力した場合、以後のすべての接続が排除されます。既存の接続はアクティブのままです。

送信元 IP アドレスからの以後の接続をブロックするだけでなく、既存の接続もドロップするには、宛先 IP アドレス、送信元と宛先のポート、およびプロトコルを入力します。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]
```

基本 IPS をサポートする IP 監査の設定

IP 監査機能は、AIP SSM を使用しないセキュリティ アプライアンスに基本 IPS サポートを提供します。シグニチャの基本リストをサポートし、シグニチャと一致するトラフィックに対して 1 つ以上のアクションを実行するようにセキュリティ アプライアンスを設定できます。

IP 監査をイネーブルにするには、次の手順を実行します。

ステップ 1　情報シグニチャに対する IP 監査ポリシーを定義するには、次のコマンドを入力します。

```
hostname(config)# ip audit name name info [action [alarm] [drop] [reset]]
```

ここで、**alarm** はパケットがシグニチャと一致したことを示すシステム メッセージを生成し、**drop** はパケットをドロップし、**reset** はパケットをドロップして接続を閉じます。アクションを定義しない場合、デフォルト アクションはアラームの生成です。

ステップ 2　攻撃シグニチャに対する IP 監査ポリシーを定義するには、次のコマンドを入力します。

```
hostname(config)# ip audit name name attack [action [alarm] [drop] [reset]]
```

ここで、**alarm** はパケットがシグニチャと一致したことを示すシステム メッセージを生成し、**drop** はパケットをドロップし、**reset** はパケットをドロップして接続を閉じます。アクションを定義しない場合、デフォルト アクションはアラームの生成です。

ステップ 3　ポリシーをインターフェイスに割り当てるには、次のコマンドを入力します。

```
ip audit interface interface_name policy_name
```

ステップ 4　シグニチャをディセーブルにする方法およびシグニチャの詳細については、『*Cisco Security Appliance Command Reference*』の **ip audit signature** コマンドを参照してください。