ストリーム TCP インスペクタ

ストリーム TCP インスペクタの概要

タイプ	インスペクタ（ストリーム）
使用方法	検査
インスタンス タイプ	マルチトン
その他のインスペクタが必要	なし
有効	`true`

Transmission Control Protocol（TCP）は、コネクション型のステートフルなトランスポート層プロトコルです。TCP は、クライアントとサーバー間で IP ネットワークを介して順序付けされたバイトストリームを確実に送信できます。TCP では、接続パラメータ値が同じ接続は、一度に 1 つしか存在できません。ホストのオペレーティングシステムは、TCP 接続の状態を管理します。

`stream_tcp` インスペクタは、TCP フロートラッキング、ストリームの正規化、およびストリームのリアセンブルを実行します。各ストリーム TCP インスペクタが、ネットワーク内の 1 つ以上のホストの TCP トラフィックを処理できます。さらに、TCP トラフィックをネットワークに送信しているホストに関する十分な情報がある場合は、それらのホストに対して `stream_tcp` インスペクタを設定できます。

ネットワーク分析ポリシー（NAP）では、Snort は設定した `stream_tcp` インスペクタそれぞれを、`binder` インスペクタの設定で定義されている TCP サービスに適用します。

複数のストリームTCPインスペクタを設定して、さまざまなオペレーティングシステムとTCPトラフィックを処理できます。

stream_tcp インスペクター構成には、次のものが含まれます。

- TCP ホストのオペレーティング システム
- オペレーティングシステムのオプション：リアセンブル時の重複の処理方法
- トラフィック処理のオプション：セッションまたは方向のバイトまたはセグメントの最大数
- TCP ストリームリアセンブルのオプション：リアセンブルされた PDU の最大サイズ

（注） インライン IPS モードでは、stream_tcp インスペクタはペイロードストリームを正規化し、重複が常に最初に見つかったコピーに解決されるようにします。各ストリーム TCP インスペクタは、繰り返される SYN、RST 検証、およびタイムスタンプチェックを処理します。

ストリーム TCP インスペクタを設定するためのベストプラクティス

stream_tcp インスペクタをリアセンブルするときは、次のベストプラクティスを考慮してください。

- Snort がフローの片側だけを検査できるように、デバイスにセンシングインターフェイスを展開しないでください。stream_tcp インスペクタで reassemble_async パラメータを有効にして、非対称トラフィックを処理できます。ただし、ストリームTCPインスペクタは、すべての場合で非対称トラフィックを処理できるわけではありません。たとえば、HTTP HEAD要求への応答により、HTTPインスペクタが同期しなくなる可能性があります。IDS モードでは、TCP 確認応答がないため、回避がはるかに簡単になります。

  IPSモードの場合、Snortがフローの両側を検査できる場合にのみデバイスを展開することをお勧めします。

- TCP トラフィックを送受信する予定の TCP ホストのオペレーティングシステムごとに stream_tcp インスペクタを作成します。同じネットワーク分析ポリシーに複数のバージョンの stream_tcp インスペクタを含めることができます。各 stream_tcp インスペクタで定義された TCP ポリシーは、binder インスペクタで指定された TCP ホストに適用されます。
- IPS モードを有効にするには、normalizer インスペクタの normalizer.tcp.ips パラメータを true に設定します。
- ネットワーク分析ポリシー（NAP）の詳細設定では、カスタムターゲットベースの stream_tcp インスペクタで識別するネットワークが親 NAP で処理されるネットワーク、

ゾーン、および VLAN のサブネットと一致するか、またはサブネットであることを確認します。

- システムは、各リーフドメインに個別のネットワークマップを作成します。マルチドメイン展開では、実際の IP アドレスを使用してこの設定を抑制すると、予期しない結果になる可能性があります。 上書き対応オブジェクトを使用すると、子孫ドメインの管理者は、グローバル コンフィギュレーションを自分のローカル環境に調整できます。
- イベントを生成し、インライン展開では、違反パケットをドロップします。 には、`stream_tcp` インスペクタのルール（GID 129）を有効にします。

TCP ストリームのリアセンブルのためのベストプラクティス

`stream_tcp` インスペクタは、TCP セッションでのサーバーからクライアントへの通信ストリーム、クライアントからサーバーへの通信ストリーム、またはその両方の通信ストリームに含まれるすべてのパケットを収集してリアセンブルします。TCP ストリームアセンブリでは、Snort は特定のストリームに含まれる個々のパケットだけを検査するのではなく、リアセンブルされた単一のエンティティ（プロトコルデータユニット（PDU））としてストリームを検査できます。PDU が大きい場合、ルールエンジンはその PDU をいくつかの部分に分割します。

ストリームのリアセンブルにより、Snort は、個々のパケットを検査する場合には検出できない可能性のあるストリームベースの攻撃を識別できます。リアセンブルする通信ストリームはネットワークの必要に基づいて指定できます。たとえば、Web サーバー上のトラフィックをモニタする際に、独自の Web サーバーから不正なトラフィックを受信する可能性が低いため、クライアントトラフィックだけを検査するという場合もあります。

`stream_tcp` インスペクタごとに、`binder` の設定で TCP ポートのリストを指定できます。TCP ストリームインスペクタは、トラフィックを識別してリアセンブルするように設定されたポートを自動的かつ透過的に組み込みます。適応型プロファイルの更新が有効になっている場合、リアセンブルするトラフィックを識別するサービスを、ポートの代わりとして、あるいはポートと組み合わせてリストすることもできます。

次の Snort インスペクタの `binder` の設定で TCP ポートを指定します。

- `dnp3`
- `ftp_server`
- `gtp_inspect`（デフォルトで提供されるポート）
- `http_inspect`
- `imap`
- `iec104`（デフォルトで提供されるポート）
- `mms`（デフォルトで提供されるポート）

- modbus（デフォルトで提供されるポート）
- pop
- sip
- smtp
- ssh
- ssl
- telnet

（注） 複数のトラフィックタイプ（クライアント、サーバー、両方）をリアセンブルすると、Snort のリソースの需要が増加する可能性があります。

ストリーム TCP インスペクタのパラメータ

ストリーム TCP リアセンブルの設定

binder インスペクタは、ネットワーク分析ポリシー（NAP）の TCP ストリームリアセンブル設定を定義します。TCP ストリーム リアセンブル ポリシーを適用するホスト IP アドレスを指定します。ストリーム TCP インスペクタは、NAP の binder に設定されているポートに自動的に関連付けられます。詳細については、『バインダインスペクタの概要』を参照してください。

（注） システムは、各リーフドメインに個別のネットワークマップを作成します。マルチドメイン展開では、実際の IP アドレスを使用してこの設定を抑制すると、予期しない結果になる可能性があります。 上書き対応オブジェクトを使用すると、子孫ドメインの管理者は、グローバルコンフィギュレーションを自分のローカル環境に調整できます。

デフォルトポリシーの default 設定では、別のターゲットベースのポリシーでカバーされていないモニタ対象ネットワークセグメントのすべての IP アドレスが指定されます。したがって、デフォルトポリシーの IP アドレスまたは CIDR ブロック/プレフィックス長は指定できず、また指定する必要もありません。また、別のポリシーでこの設定を空白にしたり、any を表すアドレス表記（0.0.0.0/0 または ::/0）を使用したりすることはできません。

policy

ターゲットホスト（複数可）のオペレーティングシステムを指定します。オペレーティングシステムは、適切な TCP リアセンブルポリシーとオペレーティングシステムの特性を決定します。ストリーム TCP インスペクタごとに policy パラメータを 1 つだけ定義できます。

（注） policy パラメータを first に設定すると、Snort はある程度の保護を提供できますが、攻撃は見逃します。TCP ストリームインスペクタの policy パラメータを編集して、適切なオペレーティングシステムを指定する必要があります。

型：列挙体

有効な値： policy パラメータのオペレーティングシステムのタイプを設定します。

表 *1 : TCP* オペレーティング システム ポリシー

ポリシー	オペレーティング システム
first	不明な OS
last	Cisco IOS
bsd	AIX FreeBSD OpenBSD
hpux_10	HP-UX 10.2 以前
hpux_11	HP-UX 11.0 以降
irix	SGI Irix
linux	Linux 2.4 カーネル Linux 2.6 カーネル
macos	Mac OS 10（Mac OS X）
old_linux	Linux 2.2 以前のカーネル
solaris	Solaris OS SunOS
vista	Windows Vista
windows	Windows 98 Windows NT Windows 2000 Windows XP
win_2003	Windows 2003

デフォルト値：bsd

max_window

受信側ホストが許可する TCP ウィンドウの最大サイズを指定します。65535 未満の整数を指定するか、または 0 を指定して TCP ウィンドウのサイズの検査を無効にすることができます。

注意 max_window の上限は、RFC 1323 で許可されている最大ウィンドウサイズです。上限を設定して、攻撃者が検出を回避するのを防ぐことができますが、TCP ウィンドウの最大サイズが非常に大きいと、自発的にサービス拒否が発生する可能性があります。

型：整数

有効な範囲：0 ～ 1,073,725,440

デフォルト値：0

overlap_limit

各 TCP セッションで許可される重複セグメントの最大数を指定します。重複セグメントを無制限に許可するには、0 を指定します。0 ～ 255 の数値を設定すると、セッションのセグメントのリアセンブルが停止します。

ルール 129:7 を有効にしてイベントを生成し、インライン展開では、違反パケットをドロップします。。

型：整数

有効な範囲：0 ～ 4,294,967,295 (max32)

デフォルト値：0

max_pdu

リアセンブルされたプロトコルデータユニット（PDU）の最大サイズを指定します。

型：整数

有効な範囲：1460 ～ 32768

デフォルト値：16384

reassemble_async

トラフィックが両方向で確認される前に、データがリアセンブルのキューに入るようにします。モニタ対象ネットワークが非同期ネットワークの場合、reassemble_async パラメータを有効にする必要があります。非同期ネットワークでは、一度に一方向のトラフィックと 1 つのフローのみが許可されます。reassemble_async パラメータが有効になっている場合、Snort はパフォーマンスを向上させるために TCP ストリームをリアセンブルしません。

（注） ストリーム TCP インスペクタは、すべての場合において非対称トラフィックを正しく処理できるわけではありません。たとえば、HTTP HEAD 要求への応答により、HTTP インスペクタが同期しなくなる可能性があります。IDS モードでは、TCP 確認応答がないため、回避がはるかに簡単になります。IPS モードの場合、ルールエンジンでフローの両側を検査できる場合にのみデバイスをデプロイすることをお勧めします。

`reassemble_async` パラメータは Secure Firewall Threat Defenseルーテッドインターフェイスと透過的インターフェイスの場合は無視されます。

型：ブール値

有効な値：`true`、`false`

デフォルト値：`true`

require_3whs

ストリーム TCP インスペクタが中間ストリームセッションの追跡を停止するまでの、起動からの秒数を指定します。いつ発生したかに関係なく、すべての中間 TCP セッションを追跡するには、`-1` を指定します。

Snort はほとんどのプロトコルストリームを同期しません。Snort は、ハンドシェイクオプション（タイムスタンプ、ウィンドウスケール、またはMSS）のいずれかが必要な場合、常にSYNを感知します。通常、中間ピックアップを許可しても IPS の有効性は向上しません。

型：整数

有効な範囲：`-1` 〜 `2,147,483,647 (max31)`

デフォルト値：`-1`

queue_limit.max_bytes

TCP接続の一方の側でセッションのキューに入れるバイトの最大数を指定します。バイト数を無制限に許可するには、`0` を指定します。

注意 `queue_limit.max_bytes` パラメータのデフォルト設定を変更する前に、Cisco TAC に連絡することをお勧めします。

型：整数

有効な範囲：`0` 〜 `4,294,967,295 (max32)`

デフォルト値：`4,194,304`

queue_limit.max_segments

TCP接続の一方の側でセッションのキューに入れるデータセグメントの最大数を指定します。データセグメントの数を無制限に許可するには、`0` を指定します。

注意 queue_limit.max_segments パラメータのデフォルト設定を変更する前に、Cisco TAC に連絡することをお勧めします。

型：整数

有効な範囲：0 ～ 4,294,967,295 (max32)

デフォルト値：3072

small_segments.count

連続する小さな TCP セグメントの予想数を超える数を指定します。連続した小さな TCP セグメントのカウントを無視するには、0 を指定します。

small_segments.count パラメータと small_segments.maximum_size パラメータには、同じ型の値を設定する必要があります。両方のパラメータに 0 を指定するか、または各パラメータをゼロ以外の値に設定します。

（注） Snort は、各セグメントの長さが 1 バイトであっても、通常の連続 TCP セグメント数を超える 2,000 の連続セグメントを考慮します。

Snort は、Threat Defense のルーテッドインターフェイスと透過的なインターフェイスの small_segments.count パラメータを無視します。

ルール 129:12 を有効にして、イベントを生成し、インライン展開では、違反パケットをドロップします。。

型：整数

有効な範囲：0 ～ 2048

デフォルト値：0

small_segments.maximum_size

TCP セグメントを小さい TCP セグメントよりも大きいものとして識別するバイト数を指定します。小さい TCP セグメントのサイズは、1 ～ 2048 バイトの範囲です。小さいセグメントの最大サイズを無視するには、0 を指定します。

Snort は、Threat Defense のルーテッドインターフェイスと透過的なインターフェイスの small_segments.maximum_size パラメータを無視します。

small_segments.maximum_size パラメータと small_segments.count パラメータには、同じ型の値を設定する必要があります。両方のパラメータに 0 を指定するか、または各パラメータをゼロ以外の値に設定します。

（注） 2048 バイトの TCP セグメントは、標準的な 1500 バイトのイーサネットフレームよりも大きいセグメントです。

ルール 129:12 を有効にして、イベントを生成し、インライン展開では、違反パケットをドロップします。。

型：整数

有効な範囲：0 ～ 2048

デフォルト値：0

session_timeout

Snort が非アクティブな TCP ストリームを状態テーブルに保持する秒数を指定します。指定した時間内にストリームがリアセンブルされない場合、Snort はそのストリームを状態テーブルから削除します。セッションがまだ動作しており、さらにパケットが表示される場合は、Snort はストリームを中間フローとして処理します。

session_timeout パラメータをホストの TCP セッションタイムアウト以上に設定することをお勧めします。

型：整数

有効な範囲：0 ～ 2,147,483,647 (max31)

デフォルト値：180

ストリーム TCP インスペクタのルール

stream_tcp インスペクタのルールを有効にし、イベントを生成し、インライン展開では、違反パケットをドロップします。。

表 *2:* ストリーム *TCP* インスペクタのルール

GID:SID	ルール メッセージ
129:1	確立済みセッションの SYN（SYN on established session）
129:2	SYN パケットのデータ（data on SYN packet）
129:3	ストリームで送信されたデータがデータを受け入れない（data sent on stream not accepting data）
129:4	TCP タイムスタンプが PAWS ウィンドウ内にない（TCP timestamp is outside of PAWS window）

GID:SID	ルール メッセージ
129:5	不良セグメント、調整済みサイズ <= 0（非推奨）（bad segment, adjusted size <= 0 (deprecated)）
129:6	ポリシーが許可するよりも大きいウィンドウ サイズ（スケーリング後）（window size (after scaling) larger than policy allows）
129:7	重複 TCP パケット数の制限に到達した（limit on number of overlapping TCP packets reached）
129:8	TCP リセット送信後にストリームで送信されたデータ（data sent on stream after TCP reset sent）
129:9	TCP クライアントがハイジャックされた可能性がある、別のイーサネットアドレス（TCP client possibly hijacked, different ethernet address）
129:10	TCP サーバーがハイジャックされた可能性がある、別のイーサネットアドレス（TCP server possibly hijacked, different ethernet address）
129:11	TCP フラグが設定されていない TCP データ（TCP data with no TCP flags set）
129:12	連続した TCP の小さなセグメントがしきい値を超えている（consecutive TCP small segments exceeding threshold）
129:13	4 ウェイハンドシェイクが検出された（4-way handshake detected）
129:14	TCP タイムスタンプがない（TCP timestamp is missing）
129:15	外部ウィンドウをリセット（reset outside window）
129:16	FIN 番号が前の FIN よりも大きい（FIN number is greater than prior FIN）
129:17	ACK 番号が前の FIN より大きい（ACK number is greater than prior FIN）
129:18	TCP リセットの受信後にストリームで送信されたデータ（data sent on stream after TCP reset received）
129:19	データを受信する前に TCP ウィンドウが閉じた（TCP window closed before receiving data）
129:20	3 ウェイハンドシェイクなしの TCP セッション（TCP session without 3-way handshake）

ストリーム TCP インスペクタの侵入ルールのオプション

stream_reassemble

一致するトラフィックでTCPストリームのリアセンブルを有効にするかどうかを指定します。stream_reassemble ルールオプションには、stream_reassemble.action、stream_reassemble.direction、stream_reassemble.noalert、および stream_reassemble.fastpath の 4 つのパラメータが含まれます。

シンタックス：stream_reassemble: <enable|disable>, <server|client|both>, noalert, fastpath;

例：stream_reassemble: disable,client,noalert;

stream_reassemble.action

ストリームのリアセンブルを停止または開始します。

型：列挙体

シンタックス：stream_reassemble: <action>;

有効な値：disable または enable

例： stream_reassemble: enable;

stream_reassemble.direction

指定された方向にアクションが適用されます。

型：列挙体

シンタックス：stream_reassemble: <direction>

有効な値：client、server、both

例： stream_reassemble: both;

stream_reassemble.noalert

ルールが一致したときにアラートを出しません。stream_reassemble.noalert パラメータはオプションです。

シンタックス：stream_reassemble: noalert;

例：stream_reassemble: noalert;

stream_reassemble.fastpath

必要に応じて、セッションの残りを信頼します。stream_reassemble.fastpath パラメータはオプションです。

シンタックス：stream_reassemble: fastpath;

例：`stream_reassemble: fastpath;`

stream_size

ストリームサイズチェックの検出オプション。TCP シーケンス番号によって決定される監視されたバイト数に従って、ルールがトラフィックを照合できるようにします。`stream_size` ルールオプションには、`stream_size.direction` と `stream_size.range` の 2 つのパラメータが含まれます。

シンタックス：`stream_size: <server|client|both|either>, <operator><number>;`

例： `stream_size: client, <6;`

stream_size.direction

比較はフローの方向に適用されます。

型：列挙体

シンタックス：`stream_size: <direction>;`

有効な値は、次のとおりです。

- `either`
- `to_server`
- `to_client`
- `both`

例：`stream_size: to_client;`

stream_size.range

ストリームサイズが指定した範囲内にあるかどうかを確認します。`range` 演算子と 1 つ以上の正の整数を指定します。

型：間隔

構文: stream_size:<range_operator><positive integer> ;または stream_size: ;

有効な値：1 つ以上の一連の正の整数と表 3 : 範囲の形式に指定されている `range_operator` の 1 つ。

例：`stream_size: >6;`

表 *3*: 範囲の形式

範囲の形式	演算子	説明
operator i		
	<	より少ない
	>	右辺と比較して大きい

範囲の形式	演算子	説明
	=	等しい
	≠	等しくない
	≤	以下
	≥	以上
j operator k		
	<>	j よりも大きく、k よりも小さい
	<=>	j 以上で k 以下

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。