Normalizer インスペクタ

ノーマライザインスペクタの概要

タイプ	インスペクタ（パケット）
使用方法	コンテキスト
インスタンス タイプ	ネットワーク
その他のインスペクタが必要	なし
有効	`true`

`normalizer` インスペクタは、パケット内のプロトコルの異常を検出して削除します。`normalizer` インスペクタを使用することで、インライン展開での検出を回避するために攻撃者がパケットを作成する可能性を最小限に抑えることができます。

（注） トラフィックをネットワークから送信する前に、ルーテッド、スイッチド、または透過的なインターフェイスあるいはインライン インターフェイス ペアを使用して、関連する設定を管理対象デバイスに展開する必要があります。

パケット内に IPv4、IPv6、ICMPv4、ICMPv6、および TCP プロトコルの組み合わせの正規化を指定できます。`normalizer` インスペクタは、パケットごとの正規化を実行し、ほとんどの正規化を処理します。`stream_tcp` インスペクタは TCP ペイロードの正規化を含む TCP の状態関連のパケットとストリームの正規化を処理します。

インライン正規化は、復号化の直後で他のインスペクタによる処理の直前に実行されます。正規化は、パケット層の内部から外部への方向で行われます。

normalizer インスペクタはイベントを生成しません。normalizer インスペクタは、他のインスペクタやインライン展開で使用できるようにパケットを作成します。インスペクタは、システムが処理するパケットがネットワーク上のホストで受信されるパケットと同じものになるようにします。

Normalizer インスペクタのパラメータ

設定内で normalizer の範囲を見つけ、normalizer インスペクタのパラメータを設定します。

ip6

IPv6 トラフィックの Reserved フラグをクリアします。

型：ブール値

有効な値：true、false

デフォルト値：false

icmp4

ICMPv4 トラフィックの Reserved フラグをクリアします。

型：ブール値

有効な値：true、false

デフォルト値：false

icmp6

ICMPv6 トラフィックの Reserved フラグをクリアします。

型：ブール値

有効な値：true、false

デフォルト値：false

ip4.base

[IPv4 フラグ（IPv4 Flags）] ヘッダーフィールドの単一ビットの [予約済み（Reserved）] サブフィールドとともにパラメータパディングをクリアします。緊急のポインター/フラグの問題を修正します。ip4.base を有効にすることをお勧めします。

型：ブール値

有効な値：true、false

デフォルト値：false

ip4.df

[IPv4 フラグ（IPv4 Flags）] ヘッダーフィールドの単一ビットの [フラグメント禁止（Don't Fragment）] サブフィールドをクリアします。`ip4.df` を有効にして、ダウンストリームルーターがパケットをドロップするのではなくフラグメント化できるようにします。`ip4.df` パラメータは、ドロップするパケットを作成する回避が行われないようにすることができます。

型：ブール値

有効な値：`true`、`false`

デフォルト値：`false`

ip4.rf

着信パケットの予約ビットをクリアします。

型：ブール値

有効な値：`true`、`false`

デフォルト値：`false`

ip4.tos

1 バイトの [差別化サービス（Differentiated Services）]（旧称 [タイプオブサービス（Type of Service）]）フィールドをクリアします。

型：ブール値

有効な値：`true`、`false`

デフォルト値：`false`

ip4.trim

過剰なペイロードを持つパケットを、IP ヘッダーに指定されたデータグラム長にレイヤ 2（たとえば、イーサネット）ヘッダーを合計した長さにまで切り捨てます。ただし、最小フレーム長より小さく切り捨てることはしません。

型：ブール値

有効な値：`true`、`false`

デフォルト値：`false`

tcp.base

TCP ヘッダーの単一ビットの [予約済み（Reserved）] サブフィールドとオプションのパディングバイトをクリアします。緊急ポインタまたは緊急フラグの問題を修正します。

型：ブール値

有効な値：`true`、`false`

デフォルト値：`false`

tcp.block

TCP 正規化中にパケットをドロップするかどうかを指定します。

有効にすると、Snortは無効になり受信ホストによってブロックされる可能性が高い異常なTCPパケット（正規化されている場合）をブロックします。たとえば、Snortは確立されたセッションの後に送信された SYN パケットをブロックします。

ルールが有効にされているかどうかに関係なく、Snort は次の TCP ストリームインスペクタのいずれかのルールに一致するすべてのパケットをドロップします。

- 129:1
- 129:3
- 129:4
- 129:6
- 129:8
- 129:11
- 129:14 ～129:19

型：ブール値

有効な値：`true`、`false`

デフォルト値：`false`

tcp.ecn

`Explicit Congestion Notification (ECN)` フラグのパケット単位またはストリーム単位の正規化を次のように有効にします。

- [パケット（Packet）] を指定すると、ネゴシエーションに関係なく、パケット単位でECNフラグがクリアされます。
- [ストリーム（Stream）] を指定すると、ECNの使用がネゴシエートされていない場合、ストリーム単位で ECN フラグがクリアされます。[ストリーム（Stream）] を指定した場合に正規化を行うには、TCP ストリーム インスペクタで `tcp.require_3whs` を有効にする必要があります。
- `tcp.ecn` パラメータを無効にするには、`off` を指定します。

型：列挙体

有効な値：`off`、`packet`、`stream`

デフォルト値：`off`

tcp.ips

再送信されるデータの一貫性が確保されるように[TCPデータ（TCP Data）]フィールドの正規化を有効にします。正しく再構成できないセグメントはすべてドロップされます。

型：ブール値

有効な値：`true`、`false`

デフォルト値：`true`

tcp.opts

トラフィックで許可する特定の TCP オプションを正規化するかどうかを指定します。Snort では、明示的に許可したオプションは正規化されません。Snort では、明示的に許可していないオプションが正規化されます。

Snort では、最適な TCP パフォーマンスを実現するために一般的に使用されている次の TCP のオプションが常に許可されます。

- 最大セグメント サイズ（MSS）（Maximum Segment Size (MSS)）
- ウィンドウ スケール（Window Scale）
- タイム スタンプ TCP（Time Stamp TCP）

他のそれほど一般的に使用されないオプションについては、Snort は自動的に許可しません。

`tcp.opts` が有効になっている場合、TCP トラフィックの正規化には次のものが含まれます。

- MSS、ウィンドウスケール、タイムスタンプ、および明示的に許可されたすべてのオプションを除き、すべてのオプションのバイトを [操作なし（No Operation）]（TCP オプション 1）に設定します。
- タイムスタンプは存在していても無効な場合、あるいは有効であってもネゴシエートされない場合、タイムスタンプオクテットを [操作なし（No Operation）] に設定します。
- タイムスタンプがネゴシエートされるものの、存在しない場合、パケットをブロックします。
- 確認応答（ACK）制御ビットが設定されていない場合、[タイムスタンプエコー応答（TSecr）（Time Stamp Echo Reply (TSecr)）] オプションフィールドをクリアします。
- SYN 制御ビットが設定されていない場合、[MSS] オプションと [ウィンドウスケール（Window Scale）] オプションを [操作なし（No Operation）]（TCP オプション 1）に設定します。

型：ブール値

有効な値：`true`、`false`

デフォルト値：`false`

tcp.pad

オプションのパディングバイトをクリアします。

型：ブール値

有効な値：`true`、`false`

デフォルト値：`false`

tcp.req_pay

ペイロードがない場合、TCP ヘッダー [緊急ポインタ（Urgent Pointer）] フィールドと緊急（URG）制御ビットをクリアします。

型：ブール値

有効な値：`true`、`false`

デフォルト値：`false`

tcp.req_urg

TCP ヘッダーの緊急（URG）制御ビットが設定されていない場合は、16 ビットの TCP ヘッダーの [緊急ポインタ（Urgent Pointer）] フィールドをクリアします。

型：ブール値

有効な値：`true`、`false`

デフォルト値：`false`

tcp.req_urp

TCP ヘッダーの [緊急ポインタ（Urgent Pointer）] フィールドが設定されていない場合は、TCP ヘッダーの `urgent (URG)` 制御ビットをクリアします。

型：ブール値

有効な値：`true`、`false`

デフォルト値：`false`

tcp.resv

TCP ヘッダーの `Reserved` ビットをクリアします。

型：ブール値

有効な値：`true`、`false`

デフォルト値：`false`

tcp.trim_mss

ペイロードが MSS より長い場合、TCP の [データ（Data）] フィールドを最大セグメントサイズ（MSS）にまで切り捨てます。

型：ブール値

有効な値：`true`、`false`

デフォルト値：`false`

tcp.trim_rst

RST パケットからデータをクリアします。

型：ブール値

有効な値：`true`、`false`

デフォルト値：`false`

tcp.trim_syn

TCP 同期（SYN）パケット内のデータを削除します。

型：ブール値

有効な値：`true`、`false`

デフォルト値：`false`

tcp.trim_win

TCP の [データ（Data）] フィールドを [ウィンドウ（Window）] フィールドに指定されたサイズにまで切り捨てます。

型：ブール値

有効な値：`true`、`false`

デフォルト値：`false`

tcp.urp

ポインタがペイロード長を上回る場合、2 バイトの TCP ヘッダーの [緊急ポインタ（Urgent Pointer）] フィールドをペイロード長に設定します。

型：ブール値

有効な値：`true`、`false`

デフォルト値：`false`

Normalizer インスペクタのルール

`normalizer` インスペクタには、関連付けられたルールがありません。

Normalizer インスペクタの侵入ルールのオプション

`normalizer` インスペクタには、侵入ルールのオプションはありません。

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。