CIP インスペクタ

CIP インスペクタの概要

タイプ	インスペクタ（サービス）
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	`stream_tcp`
有効	`false`

Common Industrial Protocol（CIP）は、産業用自動化アプリケーションをサポートするためのアプリケーションプロトコルです。EtherNet/IP（ENIP）は、イーサネットベースのネットワークで使用される CIP の実装です。

`cip` インスペクタは、TCP または UDP で実行されている CIP トラフィックと ENIP トラフィックを検出し、それを侵入ルールエンジンに送信します。カスタム侵入ルールで CIP および ENIP のキーワードを使用すると、CIP および ENIP トラフィックで攻撃を検出できます。

（注） Snort 3 では、`cip` インスペクタは CIP アプリケーションディテクタをサポートしていません。CIP アプリケーション検出を実装するには、カスタム CIP 侵入ルールを作成してインポートし、適切な IPS ルールを有効にします。詳細については、管理アプリケーションの Snort 3 設定に関するドキュメントを参照してください。

CIP インスペクタを設定するためのベストプラクティス

cip インスペクタを設定するときは、次のベストプラクティスを考慮してください。

- デフォルトの CIP 検出ポート 44818 とその他の CIP ポートを binder インスペクタに追加する必要があります。
- 侵入防御アクションをアクセス コントロール ポリシーのデフォルトのアクションとして使用することをお勧めします。
- CIP アプリケーションと ENIP アプリケーションを検出するには、対応するカスタム ネットワーク分析ポリシーで cip インスペクタを有効にする必要があります。
- アクセス コントロール ルールを使用して CIP アプリケーションまたは ENIP アプリケーションのトラフィックをブロックするには、対応するネットワーク分析ポリシーで Normalizer インスペクタとそのインラインモードが有効になっている（デフォルト設定）ことを確認してください。
- cip インスペクタのルールと CIP 侵入ルールをトリガーするトラフィックをドロップするには、対応する侵入ポリシーの **Drop when Inline** が有効になっていることを確認します。
- cip インスペクタは、次のいずれかのアクセス コントロール ポリシーのデフォルトアクションをサポートしていません。
  - **アクセス コントロール：すべてのトラフィックを信頼**
  - **アクセス コントロール：すべてのトラフィックをブロック**
- cip インスペクタは、CIP アプリケーションのアプリケーション可視性（ネットワーク検出を含む）をサポートしていません。

CIP インスペクタのパラメータ

CIP TCP ポートの設定

binder インスペクタは、CIP TCP ポートの設定を定義します。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。

例：

```
[
    {
        "when": {
                    "role": "server",
                    "proto": "tcp",
                    "ports": "44818"
        },
        "use": {
                    "type": "cip"
        }
```

```
    }
]
```

embedded_cip_path

埋め込まれた CIP 接続パスをインスペクタで確認するかどうかを決定します。

型：文字列

有効な値は、次のとおりです。

- `"false"`
- 二重引用符で囲まれた CIP パス（`"0x2 0x36"` など）。

デフォルト値：`"false"`

unconnected_timeout

デフォルトの未接続タイムアウトを秒単位で設定します。CIP 要求メッセージにプロトコル固有のタイムアウト値が含まれておらず、TCP 接続あたりの未接続な同時要求の最大数に達した場合は、このパラメータで指定した秒数の間、システムがメッセージの時間を測定します。タイマーが満了すると、他の要求用のスペースを確保するために、メッセージが削除されます。

0 を指定すると、プロトコル固有のタイムアウトが設定されていないすべてのトラフィックが最初にタイムアウトになります。

型：整数

有効な範囲：`0` 〜 `360`

デフォルト値：`300`

max_unconnected_messages

TCP 接続あたりの同時未接続 CIP メッセージの最大数を設定します。無応答のままになる可能性がある同時要求の最大数に到達すると、システムはその接続を閉じます。

型：整数

有効な範囲：`1` 〜 `10000`

デフォルト値：`100`

max_cip_connections

システムが TCP 接続ごとに許可する同時 CIP 接続の最大数を設定します。

型：整数

有効な範囲：`1` 〜 `10000`

デフォルト値：`100`

CIP インスペクタのルール

cip 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 *1: CIP* インスペクタのルール

GID:SID	ルール メッセージ
148:1	CIP データの形式が誤っている（CIP data is malformed）
148:2	CIP データは ODVA 基準に準拠していない（CIP data is non-conforming to ODVA standard）
148:3	CIP 接続の制限を超えている。最も長い間使用されていない接続が削除された（CIP connection limit exceeded. Least recently used connection removed）
148:4	CIP 未接続要求の制限を超えている。最も古い要求が削除された（CIP unconnected request limit exceeded. Oldest request removed）

CIP インスペクタの侵入ルールのオプション

cip_attribute

CIP 属性を照合する検出パラメータ。

型：間隔

シンタックス：`cip_attribute: <range_operator><positive integer>;` または `cip_attribute: <positive integer><range_operator><positive integer>;`

有効な値：0 ～ 65535 の 1 つ以上の一連の整数と表 2 : 範囲の形式に指定されている `range_operator` の 1 つ。

例：`cip_attribute: <100;`

cip_class

CIP クラスを照合する検出パラメータ。

型：間隔

シンタックス：`cip_class: <range_operator><positive integer>;` または `cip_class: <positive integer><range_operator><positive integer>;`

有効な値：0 ～ 65535 の 1 つ以上の一連の整数と表 2 : 範囲の形式に指定されている `range_operator` の 1 つ。

例：`cip_class: <25;`

cip_conn_path_class

CIP 接続パスクラスを照合する検出パラメータ。

型：間隔

シンタックス：`cip_conn_path_class: <range_operator><positive integer>;` または `cip_conn_path_class: <positive integer><range_operator><positive integer>;`

有効な値：0 ～ 65535 の 1 つ以上の一連の整数と表 2 : 範囲の形式に指定されている `range_operator` の 1 つ。

例：`cip_conn_path_class: <85;`

cip_instance

CIP インスタンスを照合する検出パラメータ。

型：間隔

シンタックス：`cip_instance: <range_operator><positive integer>;` または `cip_instance: <positive integer><range_operator><positive integer>;`

有効な値：0 ～ 65535 の 1 つ以上の一連の整数と表 2 : 範囲の形式に指定されている `range_operator` の 1 つ。

例：`cip_instance: <15;`

cip_req

CIP 要求を照合する検出パラメータ。

シンタックス：`cip_req;`

例：`cip_req;`

cip_rsp

CIP 応答を照合する検出パラメータ。

シンタックス：`cip_rsp;`

例：`cip_rsp;`

cip_service

CIP サービスを照合する検出パラメータ。

型：間隔

シンタックス：`cip_service: <range_operator><positive integer>;` or `cip_service: <positive integer><range_operator><positive integer>;`

有効な値：0 ～ 127 の 1 つ以上の一連の整数と表 2 : 範囲の形式に指定されている `range_operator` の 1 つ。

例：`cip_service: <50;`

cip_status

CIP 応答ステータスを照合する検出パラメータ。

型：間隔

シンタックス：`cip_status: <range_operator><positive integer>;` または `cip_status: <positive integer><range_operator><positive integer>;`

有効な値：0 ～ 255 の 1 つ以上の一連の整数と表 2：範囲の形式に指定されている `range_operator` の 1 つ。

例：`cip_status: <250;`

表 *2*：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
	=	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。