



バインダインスペクタ

- [バインダインスペクタの概要 \(1 ページ\)](#)
- [ポートレス設定でのサービスの自動検出 \(2 ページ\)](#)
- [バインダインスペクタを設定するためのベストプラクティス \(3 ページ\)](#)
- [バインダインスペクタのパラメータ \(5 ページ\)](#)
- [バインダインスペクタのルール \(6 ページ\)](#)
- [バインダインスペクタの侵入ルールのオプション \(7 ページ\)](#)

バインダインスペクタの概要

タイプ	インスペクタ (パッシブ)
使用方法	検査
インスタンス タイプ	単一
その他のインスペクタが必要	確立されたバインディングに依存
有効	true

各ネットワーク分析ポリシー (NAP) には、binder インスペクタが1つあります。binder は、トラフィックを検査するために特定のサービスインスペクタを使用するタイミングを決定します。binder インスペクタの設定には、同じ NAP 内の別のインスペクタがトラフィックを検査する必要がある場合に定義するポート、ホスト、CIDR、およびサービスが含まれます。binder ルールが新しいフローに一致すると、対象のインスペクタがフローにバインドされます。

binder インスペクタは、自動検出ウィザードと連携して、ポートに依存しないサービスの設定と、マルウェア コマンドおよび制御チャネルの検出を実行できます。詳細については、[Snort 3 のプロトコルとサービスの識別](#)を参照してください。

バインディングは、セッションの開始時に評価され、適切なサービスがセッションで識別された場合に再度評価されます。バインディングは、上から下に評価される使用時ルールのリストです。Snort は、最初に一致したネットワークおよびサービス設定を使用してトラフィックを検査します。

例

たとえば、CIP トラフィックを検査するように NAP を設定する場合は、次の手順を実行します。

- NAP の binder インスペクタで、検査するトラフィックの正しいポート、ロール、およびプロトコル情報を使用して、"type":"cip" セクションを更新します。
- 同じ NAP の cip インスペクタでデフォルト値を確認し、CIP トラフィックを検査するために必要な調整を行います。

次に、cip の設定とバインディングの例を示します。この例では、[バインディングスペクタのパラメータ \(5 ページ\)](#) で説明したオプションを使用します。

```
{
  "use": {
    "type": "cip"
  },
  "when": {
    "proto": "udp",
    "ports": "22222 33333",
    "role": "server"
  }
},
{
  "use": {
    "type": "cip"
  },
  "when": {
    "role": "server",
    "ports": "44818",
    "proto": "tcp"
  }
},
}
```

ポートレス設定でのサービスの自動検出

自動検出ウィザードにより、ポートに依存しないサービスの設定と、マルウェアコマンドおよび制御チャンネルの検出が可能になります。トラフィックが着信すると、binder インスペクタは最初に自動検出 wizard をフローに付加して最初のペイロードを確認し、トラフィックが使用しているサービスを特定します。たとえば、GET は HTTP を示し、HELO は SMTP を示します。サービスが決定されると、Snort は適切なサービスインスペクタをフローにバインドし、自動検出 wizard をフローから切り離します。



(注) Secure Firewall Management Center Web インターフェイスから自動検出 wizard を設定することはできません。

ルールエンジンと自動検出 wizard がトラフィックを理解して識別できない場合、binder インスペクタでポートを設定しても検査は強制されません。

自動検出とバインダの設定

binder インスペクタは侵入ルールを上から順に照合し、トラフィックに一致する最初のルールを適用します。フローで検出されたサービスに binder インスペクタを設定していない場合でも、自動検出ウィザードはフローを関連するインスペクタにバインドできます。次に例を示します。

- ペイロードが GET で、自動検出ウィザードがトラフィックタイプを HTTP として識別した場合、binder インスペクタは HTTP インスペクタをそのフローにバインドします。
- トラフィックタイプを識別できない場合、ルールエンジンは非プロトコル固有の検査を実行します。

ポートを正しく設定しないと、binder インスペクタはそのフローのサービスを自動検出できず、インスペクタをバインドできません。たとえば、ポート 88 をバインダに HTTP ポートとして設定すると、binder インスペクタは HTTP インスペクタをそのポートのすべてのフローにバインドします。ただし、フローが HTTP ではない場合、ルールエンジンはフローを HTTP として検査しません。代わりに、検査と検出がタイムアウトします。

ネットワーク分析ポリシーでの自動検出とインスペクタの有効化または無効化

自動検出の動作は、対象のインスペクタがネットワーク分析ポリシーで有効化されているか無効化されているかによって異なります。対象のインスペクタがネットワーク分析ポリシーで有効になっている場合、自動検出は期待どおりに機能します。

対象のインスペクタがネットワーク分析ポリシーで無効になっている場合でも、通常、自動検出は引き続きストリームインスペクタ（ストリーム TCP やストリーム UDP など）をフローにバインドします。ただし、ルールエンジンはサービスの検査も検出も実行しません。TCP フローの場合、ストリーム TCP インスペクタはリアセンブルを実行します。

バインディングスペクタを設定するためのベストプラクティス

バインディング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。

- そのインスペクタに必要な場合を除き、バインディングスペクタでポートを設定しないでください。ルールエンジンがトラフィックを自動検出できる場合、ポートの設定は有効性を改善しません。ただし、ポート設定が正しくないと、回避の検出に失敗する可能性があります。
- 1 つのインスペクタのみにポートを設定します。異なるプロトコルとインスペクタのバインダでポートが 2 回設定されている場合、最初のインスペクタが自動的にトリガーされません。
- デフォルトの binder インスペクタの設定に表示されない場合は、サービスインスペクタの設定を binder インスペクタに追加します。たとえば、cip インスペクタを使用する場合は、その cip インスペクタの use オプションと when オプションをバインダに追加します。

- ストリーム TCP インスペクタの場合、オペレーティングシステムの設定をカスタムバインドするようにネットワークを設定します。ネットワークの設定はすべてのポートに適用されます。
- サービスインスペクタの場合、バイндаがフロー内のプロトコルを自動検出できる場合は、ハードポートバインディングを回避します。プロトコルが検出可能でない場合、ハードポートバインディングは検出と検査を保証しません。

ポートの設定が必要なインスペクタ

関連するプロトコルでは自動検出が機能しないため、次のインスペクタのバイндаインスペクタでポートを設定します。

- `cip`
- `gtp_inspect`
- `iecl104`
- `modbus`
- `s7commplus`

ポートの設定を必要としないインスペクタ

関連するプロトコルに対して自動検出が機能するため、次のインスペクタのバイндаインスペクタでポートを設定しないでください。

- `arp_spoof`
- `dce_smb`
- `dce_tcp`
- `dnp3`
- `ftp_client`
- `ftp_server`
- `http_inspect`
- `imap`
- `normalizer`
- `pop`
- `port_scan`
- `sip`
- `smtp`
- `ssh`
- `stream_icmp`

- stream_ip
- stream_tcp
- stream_udp
- telnet

バイндаインスペクタのパラメータ

binder[]

バインダには、when オブジェクトと use オブジェクトのペアとして定義されたルールの配列が含まれています。

型：配列

例：

```
{
  binder: {
    rules: [
      {
        "when": {
          ...
        },
        "use": {
          ...
        }
      },
      {
        "when": {
          ...
        },
        "use": {
          ...
        }
      }
    ]
  }
}
```

binder[].use.type

when パラメータの条件が一致したときにデータフローにバインドするインスペクタを指定します。たとえば、CIP トラフィックを検査するには、値 `cip` を使用して `use.type` を追加します。

型：文字列

有効な値：このドキュメントで説明されている Snort 3 インスペクタの名前。

デフォルト値：binder インスペクタには、サポートされている各インスペクタの `use.type` パラメータが含まれています。

binder[].when.proto

`use.type` で指定されたインスペクタにデータフローをバインドするためにトラフィックが一致する必要があるプロトコルを指定します。たとえば、ネットワーク分析ポリシーが TCP トラフィックを検査するように設定されている場合、`binder` インスペクタはこのパラメータを `tcp` に設定する必要があります。

型：列挙体

有効な値：any、ip、icmp、tcp、udp、user、file

デフォルト値：`binder` インスペクタには、各プロトコルの `when.proto` パラメータが含まれています。

binder[].when.ports

`use.type` で指定されたインスペクタにデータフローをバインドするためにトラフィックが一致する必要があるポートを指定します。たとえば、TCP ポート 80 のトラフィックを検査するには、`when.proto` を `tcp` に設定し、`when.ports` を 80 に設定します。

10 進数または 16 進数の整数で表される 1 つ以上のポートのリストを指定します。複数のポートはスペースで区切り、リストを二重引用符で囲みます。

型：文字列

有効な範囲：1 ~ 65535

デフォルト値：65535（この値は `when.proto` の値によって異なる場合があります。）

binder[].when.role

`use.type` で指定したインスペクタにフローをバインドするためにトラフィックが一致する必要があるロールを指定します。

型：列挙体

有効な値：client、server、any

デフォルト値：any

`use.type` で指定されたインスペクタにフローをバインドするためにトラフィックが一致する必要があるサービスを指定します。

型：文字列

有効な値：着信データをカプセル化する可能性のあるサービスの名前（例：netbios-ssn または dcerpc）。

デフォルト値：なし

バイндаインスペクタのルール

`binder` インスペクタに関連付けられたルールはありません。

バインダインスペクタの侵入ルールのオプション

`binder` 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。