Binder Inspector

Binder Inspector Overview

Type	Inspector (passive)
Usage	Inspect
Instance Type	Singleton
Other Inspectors Required	Depends upon bindings established
Enabled	`true`

Each Network Analysis Policy (NAP) has one `binder` inspector. The `binder` determines when to use a certain service inspector to inspect traffic. The configurations in the `binder` inspector include the ports, hosts, CIDRs, and services that define when another inspector in the same NAP needs to inspect traffic. When a `binder` rule matches a new flow, the targeted inspector is bound to the flow.

The `binder` inspector can work with the autodetection `wizard` to perform port-independent configuration of services and detection of malware command and control channels. For more information, see Protocol and Service Identification in Snort 3.

Bindings are evaluated when a session starts and then again if and when an appropriate service is identified in the session. The bindings are a list of when-use rules evaluated from top to bottom. Snort uses the first matching network and service configurations to inspect traffic.

Example

For example, if you want to configure a NAP to inspect CIP traffic:

- In the binder inspector for the NAP, update the "type":"cip" section with the correct ports, role, and protocol information for the traffic that you want to inspect.
- Review the default values in the cip inspector for that same NAP and make any adjustments required to inspect the CIP traffic.

The following is an example of the cip configuration and binding. This example uses options described in Binder Inspector Parameters, on page 4.

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

Autodetection of Services for Portless Configuration

The autodetection wizard enables port-independent configuration of services and the detection of malware command and control channels. When traffic arrives, the binder inspector attaches the autodetection wizard to the flow at the outset and it checks the initial payload to determine the service the traffic is using. For example, GET would indicate HTTP and HELO would indicate SMTP. After the service is determined, Snort bounds the the appropriate service inspector to the flow and detaches the autodetection wizard from the flow.

Note You cannot configure the autodetection wizard through the Secure Firewall Management Center web interface.

If the rules engine and autodetection wizard cannot understand and identify the traffic, configuring a port in the binder inspector does not force inspection.

Autodetection and Binder Configuration

The binder inspector matches intrusion rules in order, from the top down, and applies the first rule to match the traffic. If you haven't configured the binder inspector for the service detected in the flow, the autodetection wizard can still bind the flow to the relevant inspector. For example:

- If the payload is GET and the autodetection wizard identifies the traffic type as HTTP, the binder inspector binds the HTTP inspector to that flow.
- If the traffic type cannot be identified, the rules engine performs a non-protocol specific inspection.

If you configure a port incorrectly, the `binder` inspector cannot autodetect the service for that flow nor can it bind an inspector to it. For example, if you configure port 88 into the binder as an HTTP port, the `binder` inspector will bind the HTTP inspector to any flow on that port. However, if the flow is not HTTP, the rules engine will not inspect it as HTTP. Instead, the inspection and detection will timeout.

Autodetection and Enable or Disable of Inspectors in the Network Analysis Policy

The behavior of autodetection changes, depending upon whether the targeted inspector is enabled or disabled in the network analysis policy. If the targeted inspector is enabled in the network analysis policy, autodetection works as expected.

If the targeted inspector is disabled in the network analysis policy, typically, autodetection still binds a stream inspector, such as stream TCP or stream UDP, to the flow. However, the rules engine does not perform service inspection or detection. For a TCP flow, the stream TCP inspector performs reassembly.

Best Practices for Configuring the Binder Inspector

Consider the following best practices when you configure the binder inspector:

- Do not configure ports in the binder inspector unless it's required for that inspector. The port configuration does not improve efficacy if the rules engine can autodetect the traffic. However, an incorrect port configuration can lead to failure to detect evasions.
- Configure a port for only one inspector. If a port is configured twice in the binder for different protocols and inspectors, it will automatically trigger the first inspector.
- Add the configuration for a service inspector to the `binder` inspector if you do not see it in the default `binder` inspector configuration. For example, if you want to use the `cip` inspector, add the `use` and `when` options for the `cip` inspector to the binder.
- For the stream TCP inspector, configure networks to custom bind operating system configurations. The network configurations apply to all ports.
- For service inspectors, avoid hard port bindings if the binder can autodetect the protocol in the flow. If the protocol is not detectable, a hard port binding does not ensure detection and inspection.

Inspectors that Require Port Configuration

Configure ports in the binder inspector for the following inspectors, because autodetection does not work for the related protocols:

- `cip`
- `gtp_inspect`
- `iec104`
- `modbus`
- `s7commplus`

Inspectors that Do Not Require Port Configuration

Do not configure ports in the binder inspector for the following inspectors, because autodetection does work for the related protocols:

- `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`

Binder Inspector Parameters

binder[]

A binder includes an array of rules defined as a pair of `when` and `use` objects.

Type: array

Example:

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

binder[].use.type

Specifies the inspector to bind to the data flow when the criteria in the `when` parameter matches. For example, to inspect CIP traffic, add `use.type` with a value of `cip`.

Type: string

Valid values: The name of any Snort 3 inspector described in this document.

Default value: The `binder` inspector includes a `use.type` parameter for each supported inspector.

binder[].when.proto

Specifies the protocol that the traffic must match to bind the data flow to the inspector specified in `use.type`. For example, if the network analysis policy is configured to inspect TCP traffic, the `binder` inspector must have this parameter set to `tcp`.

Type: enum

Valid values: `any`, `ip`, `icmp`, `tcp`, `udp`, `user`, `file`

Default value: The `binder` inspector includes a `when.proto` parameter for each protocol.

binder[].when.ports

Specifies the ports that the traffic must match to bind the data flow to the inspector specified in `use.type`. For example, to inspect traffic on TCP port 80, set `when.proto` to `tcp` and `when.ports` to `80`.

Specify a list of one or more ports represented as decimal or hex integers. Separate multiple ports with a space and enclose the list with double quotes.

Type: string

Valid range: `1 - 65535`

Default value: `65535` (This value may vary depending upon the value of `when.proto`.)

binder[].when.role

Specifies the roles that the traffic must match to bind the flow to the inspector specified in `use.type`.

Type: enum

Valid values: `client`, `server`, `any`

Default value: `any`

Specifies the service that the traffic must match to bind the flow to the inspector specified in `use.type`.

Type: string

Valid values: A name of a service that may encapsulate incoming data, for example: `netbios-ssn` or `dcerpc`.

Default value: None

Binder Inspector Rules

The `binder` inspector does not have any associated rules.

Binder Inspector Intrusion Rule Options

The `binder` inspector does not have any intrusion rule options.