Introduction

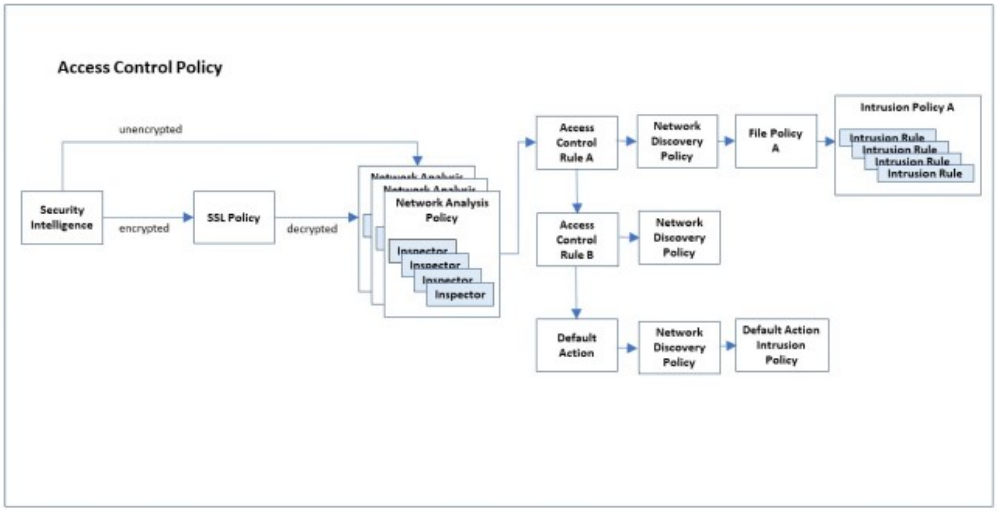
About Snort 3 Inspection

The Snort Intrusion Prevention System (IPS) analyzes network traffic in real time to provide deep packet inspection. Snort can detect and block traffic anomalies, and network probes and attacks. Snort 3 is the latest version of Snort. For more information, see https://snort.org/snort3.

Snort is designed for high performance and scalability. Snort includes a set of configurable plugins called *inspectors*. A Snort inspector can detect and analyze traffic for a certain type of network protocol or probe, normalize messages to enhance packet analysis, and inspect specific types of files embedded in a message. You configure the Snort inspectors in a Network Analysis Policy (NAP) and enable intrusion rules in an Intrusion policy.

Access Control Policies

Access control policies process traffic in several stages. The following diagram represents an example of a policy deployment. The elements addressed in this document are the Snort 3 inspectors and rule options used in intrusion rules, both highlighted in blue.



Network analysis policies enable you to configure Snort 3 inspectors to determine the traffic protocol and extract and normalize data. You can configure multiple network analysis policies, each using a uniquely configured collection of Snort 3 inspectors to normalize the data. Inspectors can alert when they detect irregularities in the data stream, but their main purpose is to prepare the data for the intrusion rules. The intrusion policies apply their configured intrusion rules to examine the data for signs of evasions, intrusions, or attacks.

Within a network analysis policy, you can customize inspection behavior for data using a given protocol by setting configuration parameters specific to the inspector that handles that protocol. For example, to configure inspection behavior for POP data, set the configuration parameters for the `pop` inspector.

You can also customize the intrusion policy for some protocols by writing custom intrusion rules using rule options specific to those protocols.

If you establish a complex configuration using multiple network analysis policies and multiple intrusion policies, the system first chooses the network analysis policy to handle the data. After the network analysis policy has applied the appropriate inspectors to perform its analysis, the data does not automatically get handed off to the corresponding intrusion policy for that protocol. The access control policy performs additional tests to determine which intrusion policy gets the data. For this reason, when configuring your access control, network analysis, and intrusion policies, ensure that data is analyzed by the correct network analysis and intrusion policy pair. For more information, see the Cisco Secure Firewall Management Center Snort 3 Configuration Guide.

Intrusion Rule Updates

Cisco periodically issues intrusion rule updates in the form of Lightweight Security Packages (LSPs). These updates may change the default values of a Snort 3 inspector's configuration parameters and intrusion rule options.

Inspector Configuration

You can enable and disable Snort inspectors as well as view and change their configurations through the Secure Firewall Management Center web interface. The Secure Firewall Management Center web interface uses the JSON format to describe the inspector configurations. For more information, see the Cisco Secure Firewall Management Center Snort 3 Configuration Guide.

To use an inspector, you must enable it through the management center web interface. In addition, for service inspectors, you must configure an entry for the service inspector in the `binder` inspector. For more information, see Binder Inspector Overview.

The Snort 3 Inspector Reference reflects the default settings for Snort 3 inspector parameters and built-in intrusion rule options. Your system may use different defaults depending upon LSP updates, or the base network access policies provided with the system. To get the most accurate understanding of inspector settings for your network access policies, view the settings in the management center web interface.

Introduction to Snort 3 Inspectors

Snort 3 inspectors are plugins that analyze and normalize packets, similar to the Snort 2 preprocessors. The list of inspectors and settings in Snort 3 does not directly correspond to the list of preprocessors and settings in Snort 2.

Snort 3 Inspectors

- ARP Spoof Inspector
- Binder Inspector
- CIP Inspector
- DCE SMB Inspector
- DCE TCP Inspector
- DNP3 Inspector
- FTP Client Inspector
- FTP Server Inspector
- GTP Inspect Inspector
- HTTP Inspect Inspector
- IEC104 Inspector
- IMAP Inspector
- MMS Inspector
- Modbus Inspector
- Normalizer Inspector
- POP Inspector
- Port Scan Inspector
- Rate Filter
- S7CommPlus Inspector
- SIP Inspector
- SMTP Inspector

- SSH Inspector
- Stream ICMP Inspector
- Stream IP Inspector
- Stream TCP Inspector
- Stream UDP Inspector
- Telnet Inspector

For each Snort 3 inspector, this document describes:

- General information about the purpose and function of the inspector.
- The type of inspector:
  - Service: Inspectors that analyze protocol data units (PDUs) used in internet service protocols (HTTP, FTP, TCP, or UDP). Examples include: `http_inspect`, `ftp_server`.
  - Passive: Inspectors that provide only configuration (`ftp_client`, `ftp_server`) or facilitate other processing (`binder`).
  - Packet: Inspectors that perform processing on raw packets before other inspectors do their processing. Examples include: `normalizer`.
  - Probe: Inspectors that perform processing on all packets after all detection has completed. Examples include: `port_scan`.
  - Stream: Inspectors that perform flow tracking, internet protocol defragmentation, and TCP reassembly. Examples include: `stream_tcp`, `stream_ip`.
  - Basic module: A configurable, built-in Snort 3 component which provides functionality to support the inspection process for multiple types of traffic. Examples include: `rate_filter`.
- Usage:
  - Inspect: Configure these inspectors within a network analysis policy (NAP). Examples include: `imap`, `ssh`.
  - Global, Context: Configure these inspectors once. Examples include: `port_scan, rate_filter`.
- Instance Type:
  - Singleton: Configure these inspectors for a single instance within a network access policy. For more details, see Singleton Inspectors, on page 5.
  - Multiton: Configure these inspectors for multiple instances within a network access policy (NAP). A NAP can contain multiple instances differentiated by network, port, or VLAN. Each instance is uniquely configured to process a specific traffic segment. For more details, see Multiton Inspectors, on page 6.
- Other inspectors required: Many inspectors depend upon other inspectors to fully process the data stream. When an inspector requires that you configure other inspectors, the documentation identifies those additional inspectors.

- Best practices for configuring the inspector: These are recommendations for optimal performance specific to each inspector.
- Configuration parameters for the inspector: You can set configuration parameters in the management center web interface under **Policies>Access Control>Network Analysis Policy>*Policy Name*>Snort 3 Version>*Inspector Name***.

Note Before modifying inspector parameters, we recommend that you understand the interaction between the inspector and enabled intrusion rules.

- Rules: The Snort 3 inspectors use rules to generate events. The built-in rules may contain classtype, references, and other metadata.
- Intrusion rule options: Customize intrusion rules by defining intrusion rule options for the data type handled by the inspector. See the Cisco Secure Firewall Management Center Snort 3 Configuration Guide for information on managing custom intrusion rules.

Note Writing custom intrusion rules is an advanced activity and must be undertaken with care. You may need to use inspectors and rule options not described in this documentation. Using some of the inspectors and intrusion rule options described in this document require specific settings in inspectors and rule options documented in the Snort open-source documentation. Some rule options have an impact on the Snort fast pattern matcher or placement of the detection cursor. For more information, see the Snort 3 open-source documentation, available at https://www.snort.org/snort3.

Singleton Inspectors

A network access policy (NAP) can use only one instance of a singleton inspector.

- A singleton inspector does not support multiple instances per NAP like multiton inspectors.
- A singleton inspector may not apply to some specific flows.

For example:

```
{
   "normalizer":{
      "enabled":true,
      "type":"singleton",
      "data":{
         "ip4":{
            "df":true
         }
      }
   }
}
```

Multiton Inspectors

A network access policy may use one or more instances of multiton inspectors, which you can configure as needed. A multiton inspector supports configuring settings based on specific conditions, including network, port, and VLAN. One set of supported settings comprises an instance. A multiton provides a default instance, and you can define additional instances based on specific conditions. If the traffic matches the conditions in an customized instance, the settings from that instance are applied. Otherwise, the settings from the default instance are applied. The name of the default instance is the same as the inspector's name.

For a multiton inspector, when you upload the overridden inspector configuration, you also need to define a matching `binder` configuration for each instance in the JSON file, otherwise, the upload results in an error. You can also create new instances, but make sure that you include a `binder` condition for every new instance that you create to avoid errors.

For example:

- Multiton inspector where the default instance is modified:

```
{
   "http_inspect":{
      "instances":[
         {
            "name":"http_inspect",
            "data":{
               "response_depth":5000
            }
         }
      ]
   }
}
```

- Multiton inspector where the default instance and default `binder` is modified:

```
{
   "http_inspect":{
      "instances":[
         {
            "name":"http_inspect",
            "data":{
               "response_depth":5000
            }
         }
      ]
   },
   "binder":{
      "rules":[
         {
            "use":{
               "type":"http_inspect"
            },
            "when":{
               "role":"any",
               "ports":"8080",
               "proto":"tcp",
               "service":"http"
            }
         }
      ]
   }
}
```

- Multiton inspector where a custom instance and a custom `binder` is added:

```
{
   "http_inspect":{
      "instances":[
         {
            "name":"http_inspect1",
            "data":{
               "response_depth":5000
            }
         }
      ]
   },
   "binder":{
      "rules":[
         {
            "use":{
               "type":"http_inspect",
               "name":"http_inspect1"
            },
            "when":{
               "role":"any",
               "ports":"8080",
               "proto":"tcp",
               "service":"http"
            }
         }
      ]
   }
}
```

Protocol and Service Identification in Snort 3

The `binder` inspector performs a unique function that affects all Snort service inspectors. Along with the Snort `wizard` module, the `binder` determines which stream or service inspector can inspect the network traffic. The configurations in the `binder` inspector include the ports, hosts, CIDRs, and services that define when another inspector in the same network analysis policy needs to inspect traffic.

The `wizard` supports port-independent configuration of services which allows for the detection of malware command and control channels.

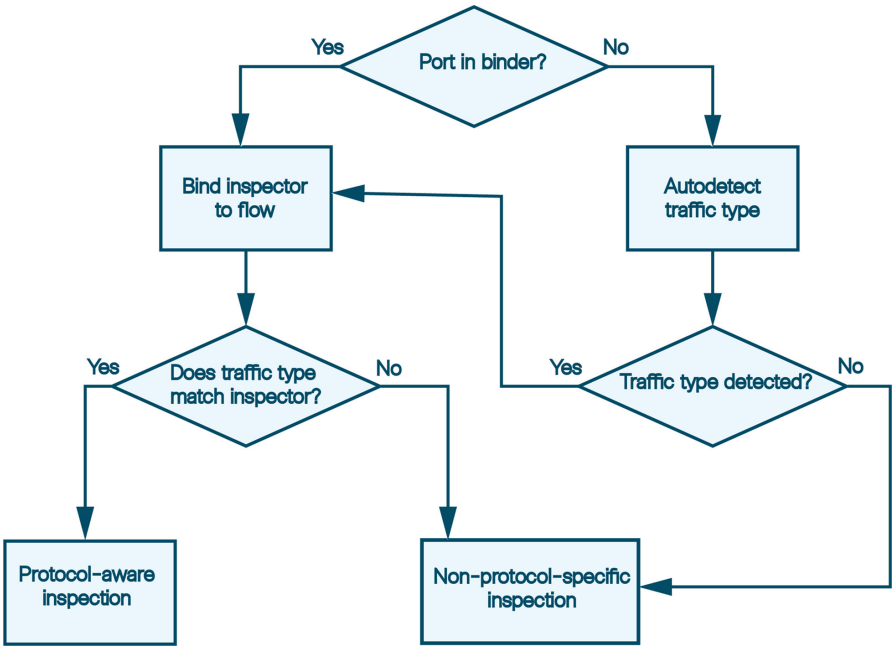
Note You cannot configure the `wizard` through the Secure Firewall Management Center web interface.

When traffic arrives at a firewall device, the `binder` inspector searches for intrusion policies and selects the appropriate network access policy (NAP) to apply. Within a NAP, the `binder` determines the appropriate stream and service inspectors to use for the data flow. Later, if the service associated with a flow changes, the NAP uses the `binder` to select a different service inspector.

The `binder` inspector configuration includes `when` parameters that describe traffic characteristics, and `use` parameters that specify which inspector to apply to traffic matching those characteristics. When determining which inspector to apply to a data flow, the `binder` inspector compares traffic against its `when` clauses in order, from the top down, and applies the `use` clause that corresponds to the first `when` clause that matches the traffic.

If no specific `binder` criteria match a data flow, the `wizard` analyzes the data flow to determine the service. The `wizard` invokes the `binder` to select the appropriate inspector for that service. If no service can be identified, the `binder` typically binds a stream inspector to the flow, and the system performs non-protocol-specific reassembly of the data packets without regard to payload content.

The following diagram illustrates how inspectors perform protocol-specific or non-protocol-specific inspection. Service inspection depends on how you configure `port`, `host`, `service`, and `CIDR` parameters in the `binder` inspector:



You can customize the inspector selection criteria by defining the `use` and `when` parameters in the `binder` inspector for a NAP in the management center web interface. For more information on the `binder` parameters, see Binder Inspector Overview. For information on navigating the management center web interface to configure inspectors, see the Cisco Secure Firewall Management Center Snort 3 Configuration Guide.

If you configure the `binder` incorrectly, it cannot detect the service for the flow or bind an inspector to it. If the rules engine and autodetection cannot understand and identify the traffic, configuring a `when` criteria such as the port in the `binder` inspector does not force inspection. For example, if you configure port 88 in the `binder` as an HTTP port, the `binder` binds the `http_inspect` inspector to any flow on that port. But if the flow is not HTTP, the rules engine does not inspect the data as HTTP, but instead performs port-based detection.

Autodetection and Enabled or Disabled 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 described above.

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.