

SCADA Preprocessors

The following topics explain preprocessors for Supervisory Control and Data Acquisition (SCADA) protocols, and how to configure them:

- Introduction to SCADA Preprocessors, on page 1
- License Requirements for SCADA Preprocessors, on page 1
- Requirements and Prerequisites for SCADA Preprocessors, on page 2
- The Modbus Preprocessor, on page 2
- The DNP3 Preprocessor, on page 4
- The CIP Preprocessor, on page 6

Introduction to SCADA Preprocessors

Supervisory Control and Data Acquisition (SCADA) protocols monitor, control, and acquire data from industrial, infrastructure, and facility processes such as manufacturing, production, water treatment, electric power distribution, airport and shipping systems, and so on. The Firepower System provides preprocessors for the Modbus, Distributed Network Protocol (DNP3), and Common Industrial Protocol (CIP) SCADA protocols that you can configure as part of your network analysis policy.

If the Modbus, DNP3, or CIP preprocessor is disabled, and you enable and deploy an intrusion rule that requires one of these preprocessors, the system automatically uses the required preprocessor, with its current settings, although the preprocessor remains disabled in the web interface for the corresponding network analysis policy.

License Requirements for SCADA Preprocessors

FTD License

Threat

Classic License

Protection

Requirements and Prerequisites for SCADA Preprocessors

Model support

Any.

Supported domains

Any

User roles

- Admin
- Intrusion Admin

The Modbus Preprocessor

The Modbus protocol, which was first published in 1979 by Modicon, is a widely used SCADA protocol. The Modbus preprocessor detects anomalies in Modbus traffic and decodes the Modbus protocol for processing by the rules engine, which uses Modbus keywords to access certain protocol fields.

A single configuration option allows you to modify the default setting for the port that the preprocessor inspects for Modbus traffic.

Related Topics

SCADA Keywords

Modbus Preprocessor Ports Option

Ports

Specifies the ports that the preprocessor inspects for Modbus traffic. Separate multiple ports with commas.

Configuring the Modbus Preprocessor

You should not enable this preprocessor in a network analysis policy that you apply to traffic if your network does not contain any Modbus-enabled devices.

In a multidomain deployment, the system displays policies created in the current domain, which you can edit. It also displays policies created in ancestor domains, which you cannot edit. To view and edit policies created in a lower domain, switch to that domain.

Procedure

Step 1 Choose **Policies** > **Access Control**, then click **Network Analysis Policies** or **Policies** > **Access Control** > **Intrusion**, then click **Network Analysis Policies**.

Note
If your custom user role limits access to the first path listed here, use the second path to access the policy.

Step 2 Click **Edit** ( ) next to the policy you want to edit.

If **View** ( ) appears instead, the configuration belongs to an ancestor domain, or you do not have permission to modify the configuration.

Step 3 Click **Settings** in the navigation panel.

Step 4 If **Modbus Configuration** under **SCADA Preprocessors** is disabled, click **Enabled**.

Step 5 Click **Edit** ( ) next to **Modbus Configuration**.

Step 6 Enter a value in the **Ports** field.

Separate multiple values with commas.

Step 7 To save changes you made in this policy since the last policy commit, click **Policy Information**, then click **Commit Changes**.

If you leave the policy without committing changes, cached changes since the last commit are discarded if you edit a different policy.

What to do next

- If you want to generate events and, in an inline deployment, drop offending packets, enable Modbus preprocessor rules (GID 144). For more information, see Setting Intrusion Rule States and Modbus Preprocessor Rules, on page 3.
- Deploy configuration changes; see Deploy Configuration Changes.

Related Topics

Managing Layers
Conflicts and Changes: Network Analysis and Intrusion Policies

Modbus Preprocessor Rules

You must enable the Modbus preprocessor rules in the following table if you want these rules to generate events and, in an inline deployment, drop offending packets.

Table 1: Modbus Preprocessor Rules

Preprocessor Rule GID:SID	Description
144:1	Generates an event when the length in the Modbus header does not match the length required by the Modbus function code. Each Modbus function has an expected format for requests and responses. If the length of the message does not match the expected format, this event is generated.

Preprocessor Rule GID:SID	Description
144:2	Generates an event when the Modbus protocol ID is non-zero. The protocol ID field is used for multiplexing other protocols with Modbus. Because the preprocessor does not process these other protocols, this event is generated instead.
144:3	Generates an event when the preprocessor detects a reserved Modbus function code.

The DNP3 Preprocessor

The Distributed Network Protocol (DNP3) is a SCADA protocol that was originally developed to provide consistent communication between electrical stations. DNP3 has also become widely used in the water, waste, transportation, and many other industries.

The DNP3 preprocessor detects anomalies in DNP3 traffic and decodes the DNP3 protocol for processing by the rules engine, which uses DNP3 keywords to access certain protocol fields.

Related Topics

DNP3 Keywords

DNP3 Preprocessor Options

Ports

Enables inspection of DNP3 traffic on each specified port. You can specify a single port or a comma-separated list of ports.

Log bad CRCs

Validates the checksums contained in DNP3 link layer frames. Frames with invalid checksums are ignored.

You can enable rule 145:1 to generate events and, in an inline deployment, drop offending packets when invalid checksums are detected.

Configuring the DNP3 Preprocessor

You should not enable this preprocessor in a network analysis policy that you apply to traffic if your network does not contain any DNP3-enabled devices.

In a multidomain deployment, the system displays policies created in the current domain, which you can edit. It also displays policies created in ancestor domains, which you cannot edit. To view and edit policies created in a lower domain, switch to that domain.

Procedure

Step 1 Choose **Policies** > **Access Control**, then click **Network Analysis Policies** or **Policies** > **Access Control** > **Intrusion**, then click **Network Analysis Policies**.

Note
If your custom user role limits access to the first path listed here, use the second path to access the policy.

Step 2 Click **Edit** ( ) next to the policy you want to edit.

If **View** ( ) appears instead, the configuration belongs to an ancestor domain, or you do not have permission to modify the configuration.

Step 3 Click **Settings** in the navigation panel.

Step 4 If **DNP3 Configuration** under **SCADA Preprocessors** is disabled, click **Enabled**.

Step 5 Click **Edit** ( ) next to **DNP3 Configuration**.

Step 6 Enter a value for **Ports**.

Separate multiple values with commas.

Step 7 Check or clear the **Log bad CRCs** check box.

Step 8 To save changes you made in this policy since the last policy commit, click **Policy Information**, then click **Commit Changes**.

If you leave the policy without committing changes, cached changes since the last commit are discarded if you edit a different policy.

What to do next

- If you want to generate events and, in an inline deployment, drop offending packets, enable DNP3 preprocessor rules (GID 145). For more information, see Setting Intrusion Rule States, DNP3 Preprocessor Options, on page 4, and DNP3 Preprocessor Rules, on page 5.
- Deploy configuration changes; see Deploy Configuration Changes.

Related Topics

Managing Layers
Conflicts and Changes: Network Analysis and Intrusion Policies

DNP3 Preprocessor Rules

You must enable the DNP3 preprocessor rules in the following table if you want these rules to generate events and, in an inline deployment, drop offending packets.

Table 2: DNP3 Preprocessor Rules

Preprocessor Rule GID:SID	Description
145:1	When **Log bad CRC** is enabled, generates an event when the preprocessor detects a link layer frame with an invalid checksum.
145:2	Generates an event and blocks the packet when the preprocessor detects a DNP3 link layer frame with an invalid length.

Preprocessor Rule GID:SID	Description
145:3	Generates an event and blocks the packet during reassembly when the preprocessor detects a transport layer segment with an invalid sequence number.
145:4	Generates an event when the DNP3 reassembly buffer is cleared before a complete fragment can be reassembled. This happens when a segment carrying the FIR flag appears after other segments have been queued.
145:5	Generates an event when the preprocessor detects a DNP3 link layer frame that uses a reserved address.
145:6	Generates an event when the preprocessor detects a DNP3 request or response that uses a reserved function code.

The CIP Preprocessor

The Common Industrial Protocol (CIP) is a widely used application protocol that supports industrial automation applications. EtherNet/IP (ENIP) is an implementation of CIP that is used on Ethernet-based networks.

The CIP preprocessor detects CIP and ENIP traffic running on TCP or UDP and sends it to the intrusion rules engine. You can use CIP and ENIP keywords in custom intrusion rules to detect attacks in CIP and ENIP traffic. See CIP and ENIP Keywords. Additionally, you can control traffic by specifying CIP and ENIP application conditions in access control rules. See Configuring Application Conditions and Filters.

CIP Preprocessor Options

Ports

Specifies the ports to inspect for CIP and ENIP traffic. You can specify an integer from 0 to 65535. Separate multiple port numbers with commas.

Note You must add the default CIP detection port 44818 and any other ports you list to the TCP stream **Perform Stream Reassembly on Both Ports** list. See TCP Stream Preprocessing Options and Creating a Custom Network Analysis Policy.

Default Unconnected Timeout (seconds)

When a CIP request message does not contain a protocol-specific timeout value and **Maximum number of concurrent unconnected requests per TCP connection** is reached, the system times the message for the number of seconds specified by this option. When the timer expires, the message is removed to make room for future requests. You can specify an integer from 0 to 360. When you specify 0, all traffic that does not have a protocol-specific timeout times out first.

Maximum number of concurrent unconnected requests per TCP connection

The number of concurrent requests that can go unanswered before the system closes the connection. You can specify an integer from 1 to 10000.

Maximum number of CIP connections per TCP connection

The maximum number of simultaneous CIP connections allowed by the system per TCP connection. You can specify an integer from 1 to 10000.

CIP Events

By design, application detectors detect and event viewers display the same application one time per session. A CIP session can include multiple applications in different packets, and a single CIP packet can contain multiple applications. The CIP preprocessor handles all CIP and ENIP traffic according to the corresponding intrusion rule.

The following table shows the CIP values displayed in event views.

Table 3: CIP Event Field Values

Event Field	Displayed Value
Application Protocol	CIP or ENIP
Client	CIP Client or ENIP Client
Web Application	The specific application detected, which is: • For access control rules that allow or monitor traffic, the last application protocol Access control rules that you configure to log connections might not generate e applications, and access control rules that you do not configure to log connecti events for CIP applications. • For access control rules that block traffic, the application protocol that triggere When an access control rule blocks a list of CIP applications, event viewers disp that is detected.

CIP Preprocessor Rules

If you want the CIP preprocessor rules listed in the following table to generate events, you must enable them. See Setting Intrusion Rule States for information on enabling rules.

Table 4: CIP Preprocessor Rules

GID:SID	Rule Message
148:1	CIP_MALFORMED
148:2	CIP_NON_CONFORMING
148:3	CIP_CONNECTION_LIMIT

GID:SID	Rule Message
148:4	CIP_REQUEST_LIMIT

Guidelines for Configuring the CIP Preprocessor

Note the following when configuring the CIP preprocessor:

- You must add the default CIP detection port 44818 and any other CIP **Ports** you list to the TCP stream **Perform Stream Reassembly on Both Ports** list. See CIP Preprocessor Options, on page 6, Creating a Custom Network Analysis Policy, and TCP Stream Preprocessing Options.
- Event viewers give special handling to CIP applications. See CIP Events, on page 7.
- We recommend that you use an intrusion prevention action as the default action of your access control policy.
- The CIP preprocessor does not support an access control policy default action of **Access Control: Trust All Traffic**, which may produce undesirable behavior, including not dropping traffic triggered by CIP applications specified in intrusion rules and access control rules.
- The CIP preprocessor does not support an access control policy default action of **Access Control: Block All Traffic**, which may produce undesirable behavior, including blocking CIP applications that you do not expect to be blocked.
- The CIP preprocessor does not support application visibility for CIP applications, including network discovery.
- To detect CIP and ENIP applications and use them in access control rules, intrusion rules and so on, you must manually enable the CIP preprocessor in the corresponding custom network analysis policy. See Creating a Custom Network Analysis Policy, Setting the Default Network Analysis Policy, and Configuring Network Analysis Rules.
- To drop traffic that triggers CIP preprocessor rules and CIP intrusion rules, ensure that **Drop when Inline** is enabled in the corresponding intrusion policy. See Setting Drop Behavior in an Inline Deployment.
- To block CIP or ENIP application traffic using access control rules, ensure that the inline normalization preprocessor and its **Inline Mode** option are enabled (the default setting) in the corresponding network analysis policy. See Creating a Custom Network Analysis Policy, Setting the Default Network Analysis Policy, and Preprocessor Traffic Modification in Inline Deployments.

Configuring the CIP Preprocessor

Before you begin

- You must add the default CIP detection port 44818 and any other ports you list as CIP **Ports** to the TCP stream **Perform Stream Reassembly on Both Ports** list. See CIP Preprocessor Options, on page 6, Creating a Custom Network Analysis Policy, and TCP Stream Preprocessing Options.
- Familiarize yourself with Guidelines for Configuring the CIP Preprocessor, on page 8.
- The CIP preprocessor is not supported for Firepower Threat Defense devices.

Procedure

Step 1 Choose **Policies** > **Access Control**, then click **Network Analysis Policies** or **Policies** > **Access Control** > **Intrusion**, then click **Network Analysis Policies**.

Note
If your custom user role limits access to the first path listed here, use the second path to access the policy.

Step 2 Click **Edit** ( ) next to the policy you want to edit.

If **View** ( ) appears instead, the configuration belongs to an ancestor domain, or you do not have permission to modify the configuration.

Step 3 Click **Settings** in the navigation panel.

Step 4 If **CIP Configuration** under **SCADA Preprocessors** is disabled, click **Enabled**.

Step 5 You can modify any of the options described in CIP Preprocessor Options, on page 6.

Step 6 To save changes you made in this policy since the last policy commit, click **Policy Information**, then click **Commit Changes**.

If you leave the policy without committing changes, cached changes since the last commit are discarded if you edit a different policy.

What to do next

- If you want to generate events and, in an inline deployment, drop offending packets, enable CIP intrusion rules and, optioally, CIP preprocessor rules (GID 148). For more information, see Setting Intrusion Rule States, CIP Preprocessor Rules, on page 7, and CIP Events, on page 7.
- Deploy configuration changes; see Deploy Configuration Changes.