Sentinel Plug-Ins 2020.1r1

Connector for Agent Manager

February 2020

Legal Notice

Contents

About This Guide

The *Connector for Agent Manager Guide* helps you to install and configure the Connector for Agent Manager, which helps to integrate data collection from various Sentinel Agent Manager and UNIX Agent Manager agents.

Audience

This guide is intended to introduce a Sentinel administrator to the process of integrating the associated data source with Sentinel to facilitate data collection from that source.

Additional Documentation

For complete documentation on the Sentinel products, see the NetIQ Documentation Web site.

For information on building your own plug-ins, go to the Sentinel SDK Web page.

Contacting Novell and NetIQ

Sentinel is now a NetIQ product, but Novell still handles many support functions.

- NetIQ Web site
- Technical Support
- Self Support
- Patch download site
- Sentinel Community Support Forums
- Sentinel TIDs
- Sentinel Plug-in Web site
- **Notification Email List:** Sign up through the Sentinel Plug-in Web site

Contacting Sales Support

For questions about products, pricing, and capabilities, please contact your local partner. If you cannot contact your partner, please contact our Sales Support team.

Worldwide: NetIQ Office Locations

United States and Canada: 888-323-6768

Email: info@netiq.com

Web site: www.netiq.com

1 Introduction

The Connector for Agent Manager allows a Sentinel system to receive events from NetIQ agents including agents for Windows, UNIX, and iSeries. These agents might send events directly to the Connector (NetIQ Security Agent for UNIX) or indirectly by using Sentinel Agent Manager. The Connector forwards events in a format similar to the FILE Connector, SYSLOG Connector, or Windows Event (WMI) Connector so they can be parsed by a wide variety of Collectors. This document describes the features, installation, configuration and usage of the Connector for Agent Manager.

Supported Platforms, Agents, and Collectors

This section provides information about Sentinel platforms, Agents, and Collectors that the Connector for Agent Manager supports.

Sentinel Platforms

The supported Sentinel platforms vary depending on whether the Connector is deployed on the local Collector Manager or a remote Collector Manager.

In Remote Collector Manager: The Connector supports Sentinel 7.4 and later.

In Local Collector Manager: Connector versions 2017.1r1 and later do not support the following Sentinel platforms:

- Sentinel 8.0.1.1 and prior
- Sentinel 7.4.4.1 and prior

Agents

The Connector receives data from the following:

- Sentinel Agent Manager 7.3 and later (SAM Agent)
- Security Agent for UNIX 7.4 and later (UNIX Agent)

Collectors

The Connector sends events to Collectors that support SYSLOG, WMS, and FILE connection methods.

Functionality

The Connector does the following:

- Listens on HTTP or HTTPS ports for events originating from NetIQ agents.
- Auto-instantiates event source, Connector, and Collector, as needed.

- Determines the best Collector to parse the events based on AppID (application ID) or regular expression matching rules.
- Forwards events to Collectors in a format similar to events coming from either a FILE, SYSLOG, or WMS connector.

Event Source Creation and Sharing

Each agent installed on a computer might send one or more types of data. For example, a UNIX agent might simultaneously send events originating from the Red Hat operating system, an Oracle database, and an Apache Web server. A Windows agent might simultaneously send events originating from the Windows operating system, an Active Directory domain controller, and an instance of Symantec Antivirus. By default, when the Connector receives this data, it instantiates a new event source, Connector instance, and a Collector instance to manage this data. The administrator can maintain granular control on each of these instances to start, stop, filter, or view event rates.

These event sources are identified by the Agent (or computer) and the connection method (FILE, SYSLOG, or WMS). The event source is named <*Agent*>/<*Connection method*>. The following table shows an example for several Windows Agents using the default event source creation mode.

Collectors	Agent Computers	Event Sources
Microsoft Active Directory and Windows	Computer 1	Computer1/WMS
	Computer 2	Computer2/WMS
	Computer 3	Computer3/WMS
	Computer 4	Computer4/WMS
Symantec Antivirus	Computer 1	Computer1/FILE
	Computer 2	Computer2/FILE
	Computer 3	Computer3/FILE
	Computer 4	Computer4/FILE

If Event Source Sharing is enabled, the Connector consolidates similar event sources across many different agents. Whenever possible, event sources are only identified by log name and connection method. The event source is named <*log name*>/<*Connection method*> (Shared). The Agent is not part of the unique identifier for an event source. Event Source Sharing can drastically reduce the number of event sources, but it also reduces the granularity of control the administrator has for event source management. You should enable Event Source Sharing in environments with a large number of agents (5000 or higher) or with a large number of distinct connection methods per agent. The following table shows an example for several Windows Agents using Event Source Sharing mode.

Collectors	Log Sources	Event Sources
Microsoft Active Directory and Windows	Security	Security/WMS (Shared)
	System	System/WMS (Shared)
Symantec Antivirus	*File provider*	*File provider*/FILE (Shared)

There might be scenarios in which the agent, computer name, or IP address is important to proper event routing or data parsing. For example:

- The Connector might receive multi-record events separately and the Connector must combine these events using the information about which agent generated the data.
- Unique matching rules (based on regular expressions) might be used to associate a specific agent IP address with a specific Collector.

In these scenarios, Event Source Sharing is overridden. The following mechanisms override the Event Source Sharing and revert it to the default mode for event source identification (<*agent*>/<*Connection method*>).

- Set the `Stateful` property in the Collector plug-in properties to `true`. For NetIQ Collectors, this property is set to `true` by default.
- Set the `Stateful` property for specific Collectors in the Connector configuration.

For information about enabling Event Source Sharing and setting Stateful Collector Modes, see Step 15b and Step 15c in "Adding an Agent Manager Event Source Server" on page 13.

Collector Forwarding

The Connector evaluates incoming events to determine which Collector should parse the data. The Connector applies these matching rules in order until the Connector determines the Collector to send the event.

1. Send events with a specific App ID (application ID) to the Collector whose App ID matches with the event App ID.
2. Assess events to see if the agent IP address is in the cache and is associated with a known Collector. If the IP address is in the cache, the associated Collector parses the events.
3. Compare events to regular expression-based Unique Matching Rules in the Collectors. If a match is found, add the IP address for the agent to the cache and associate the IP address with the matching Collector. This Collector parses all future events (unless they have a specific AppID).
4. Send events that do not match the AppID, an agent IP address in the cache, or one of the Unique Matching Rules to the Universal Event Collector for basic parsing.

2 Installing the Connector

If the Connector for Agent Manager is not pre-installed with Sentinel, you should download and install the Connector as follows:

1. Download the appropriate `.zip` file from the Sentinel Plug-ins Web site (http://support.novell.com/products/sentinel/secure/sentinelplugins.html) and save the file to the local machine where you want to run Event Source Management.
2. Launch the Event Source Management (Live View) window.

   Log in to the Sentinel Control Center, go to the Event Source Management menu, and select the **Live View** option. For more information, see "*Event Source Management*" in the *Sentinel 7.1 Administration Guide*.
3. Select **Tools > Import plug-in** to display the Plug-in Import Type window.
4. In the Plug-in Import Type window, select the **Import Collector or Connector plug-in package file (.zip, .clz, .cnz)** option.
5. Click **Next.**
6. In the Choose Plug-in Package File window, browse to and select the Connector file you just downloaded.
7. Follow the Import Plug-In Wizard instructions to import the Connector into the plug-in repository.

   If another version of this Connector is already in use, you can click **View Deployed Plug-ins** to see which event source objects use the deployed Connectors.
8. (Optional) To update the deployed Connectors, select **Update Deployed Plug-ins**.
9. Click **Finish**.

3 Setting Up Connection

This section describes how to set up the Connector for Agent Manager to receive messages from agents and route the events they contain to a Collector for processing. You need to configure one or more Event Source Servers. All Collectors, Connectors, and Event Sources are auto-created when messages are received from the agents.

Adding an Agent Manager Event Source Server

Sentinel 7.1 or later comes with an Agent Manager Event Source Server pre-installed on the Collector Manager that runs on the Sentinel server. The pre-installed Event Source Server listens for HTTPS connections on port 1590. For remote Collector Managers, you need to manually create Event Source Servers. You can also run multiple Agent Manager Event Source Servers in a Collector Manager if needed.

To add an Agent Manager Event Source Server:

1. Ensure that the Collector Manager is able to resolve the hostname to a valid IP address. To do this, add the hostname to the `/etc/hosts` file to the line containing the IP address and then enter `hostname -f` to ensure that the hostname is displayed properly.
2. Restart Sentinel service:

   ```
   rcsentinel restart
   ```

3. Right-click the Collector Manager where you want the Event Source Server to run and select **Add Event Source Server**.
4. Choose the Agent Manager Connector.
5. Click **Next**.
6. In the Networking window, select a network interface setting:
   - **All network interfaces:** Binds the port on all the IP addresses of the machine (including the local loopback).
   - **Internal loopback interface:** Binds the port to only the local loopback address.
   - **Network interface with this IP:** Allows the port to be bound to one of the IP address on a machine with multiple IP addresses.
7. Choose the HTTPS or HTTP protocol and specify a port on which the Agent Manager Event Source Server listens for messages. The default port is 1590.

   NOTE: If the Agent Manager Event Source Server is on a Linux or UNIX computer, binding to ports less than 1024 requires root privileges. We recommend that you run the server on a port greater than 1024 and change the source devices to send to this new port or use port forwarding. For more information, see Appendix A, “Port Forwarding,” on page 21.
8. Click **Next**.

9 In the Security window, configure the SSL Cryptographic settings:

9a Select one of the options from the **Client Authentication Type** field. The client authentication type determines how strictly the SSL Agent Manager Event Source Server verifies the identity of Agent Manager Event Sources that are attempting to send their data. You must use a strict client authentication policy that is applicable in your environment to prevent rogue Agent Manager Event Sources from sending unwanted data.

- **Open:** Allows all SSL connections from the clients (event sources) and does not perform any client certificate validation or authentication.
- **Strict:** Validates whether the certificate is a valid X.509 certificate and also checks whether the client certificate is trusted by the Event Source Server.

  For the **Strict** option, click **Import** to import a truststore. The truststore must include the certificate of the certificate authority (CA) that signed the client certificate and the client certificate (optional). You can request for the client certificate from the organizational CA or a well-known CA.

  NOTE: The **Import** option is disabled when you configure Sentinel in FIPS mode. When you use custom certificates that are digitally signed by a CA, you must import the appropriate custom certificate file.

  For instructions on creating a truststore, see Appendix B, “Keystore and Truststore Creation,” on page 23. Click **Details** to view the list of certificates imported from the truststore.

9b Select one of the options from the **Server Key Pair Settings** field. The server key pair settings determine which certificate is used to authenticate the Agent Manager server to the event sources.

- **Internal (default):** Directs the Connector to generate a server key pair.
- **Custom:** Overwrites the internal certificate that comes with the Connector and uses the custom certificate. Click **Import** to import a keystore. The keystore must contain at least one private and public key pair. If the keystore has more than one key pair, a popup screen allows you to select one of the key pairs. For instructions on creating a keystore, see Appendix B, “Keystore and Truststore Creation,” on page 23. The **Details** to view the certificate imported from the keystore.

10 Click **Next**.

11 In **Event Source Auto-Configuration** tab, specify the default general properties. These properties will be set on any new Event Sources, which are auto-created by the Event Source Server.

- **Alert if no data received in specified time period:** (Optional) Select this option to send a No Data Alert event to Sentinel if no data is received by the Connector in the specified time period. You can use the **Send repeated alerts every time period** option to resend the alert if multiple time periods pass consecutively without receiving data from the Connector.
- **Limit Data Rate:** (Optional) Use this option to set a maximum limit on the rate of data the Connector can send to Sentinel. If the data rate limit is reached, Sentinel throttles back on the source to limit the flow.
- **Time Zone:** Specify the time zone to be set for the event source when it is created.
- **Trust Event Source Time:** (Optional) Select this option to set the event time to the time the event occurred rather than the time Sentinel received the data.

12 Click **Next**.

13 (Optional) In the **Queue Management** tab, specify the time interval to update the Queue statistics in the **Queue Statistics Refresh Period (in minutes)** field. For more information on the **Queue Management** tab, see “Managing the Queue” on page 17.

14 Click **OK**, then configure the following general settings in the General window:

- **Name:** Specify the name by which you want to identify this Connector.
- **Details:** Click **Details** to view additional information about the plug-in.
- **Run:** (Optional) Select this option to specify that this Connector must be started when the configuration is complete.

15 (Optional) Click **Advanced Configuration** if you need to configure custom applications for the WMS and FILE connection methods.

NOTE: This is not something you will normally need to do. Custom applications are only needed if the application is not already defined by a Collector plug-in. For more information about Collector-defined applications, see Appendix C, “Collector Developer Notes,” on page 25.

15a In the Advanced Configuration window, set the custom applications for WMS and FILE methods.

NOTE: To set the custom applications for WMS method, click the **WMS Applications** tab. To set the custom applications for FILE method, click the **FILE Applications** tab. These configuration options are same for both WMS and FILE methods.

- **Add:** Click **Add** to create a new custom application for the Connection method.
- **Applications:** Specify the application name. This is the application name that is sent by agents in events. It identifies the application that the event originated from. The Connector uses the application name to determine what Collector and Collector connection mode the event should be routed to.
- **Collector:** Specify the Collector to which events containing the application name will be routed.
- **Connector Mode:** Specify the Connector mode in the Collector's WMS or FILE connection method that will handle events containing the application name.
- **Default Collector:** Specify the default Collector to route FILE or WMS events to if the application name in the event is not defined.

  If the application name in an event is not defined as a custom application or in a Collector plug-in *Applications* property, the event will be routed to this Collector. For more information about Collector plug-ins *Applications* property, see Appendix C, “Collector Developer Notes,” on page 25.
- **Default Mode:** Specify the Connector mode within the default Collector, which is to be used for FILE or WMS events, if the application name in the event is not defined.
- **Show WMS Collector Applications:** Click **Show WMS Collector Applications** to see the list of applications defined for the WMS method in the Collector plug-in *Applications* properties.

  If you are on the FILE Applications tab, click **Show FILE Collector Applications** to see the list of applications defined for the FILE method in the Collector plug-in *Applications* properties.

  For more information about Collector plug-ins *Applications* property, see Appendix C, “Collector Developer Notes,” on page 25.

NOTE: Custom applications are normally used to define applications that are not specified in the Collector plug-in Applications properties. However, they might also be used to override applications specified by the Collector plug-in Applications properties. When determining what Collector and Collector connection mode to route an event to, the Connector does the following:

1. Checks if there is a Collector or mode specified in the Custom Applications table. If so, routes the event there.
2. If there is no custom application defined, checks if the application is specified in some Collector plug-in *Applications* property. If so, routes the event there.
3. If there is no custom application, and the application is not defined in a Collector plug-in *Applications* property, route the event to the default Collector or default mode that has been specified.

15b Click **Advanced Options** > **Share Event Sources** to automatically create event sources for each unique log source or the application (if the log source is not available).

The event sources are shared across multiple computers. For more information about sharing event sources, see "Event Source Creation and Sharing" on page 8.

NOTE

- If you enable **Share Event Sources** on an existing Connector, you should delete the event sources that were created using the old, default method. Otherwise, you will have a large number of unused and unnecessary event sources.
- Stateful Collector modes should not share event sources across computers. If you select **Shared Event Sources**, click the **Stateful Collector Modes** tab to specify stateful Collector modes.

15c Click the **Stateful Collector Modes** tab to display the list of Collector modes that are defined to be stateful. Stateful Collector modes require event sources to be identified by computer and log source, because they keep state per event source.

NOTE: If you have selected the **Share Event Sources** option in the **Advanced Options** tab, you must ensure that Collector modes which are stateful are all specified here, because stateful Collector modes should not share event sources across computers. You should add the Collector modes that are not already specified.

Some Collector modes are defined to be stateful in the Collector plug-ins. The *Stateful* property is used for this purpose. For more information, see "Stateful Collector Modes" on page 25. These Collector modes have a "`Yes`" in the Plugin Defined column. They cannot be modified or deleted.

To create additional stateful Collector modes, click **Add** and specify the Collector, Connector Method, and Connector mode. These Collector modes have a "`No`" in the Plugin Defined column.

16 Click **Finish**.

Managing the Queue

The Connector has queuing enabled to help handle high event rates, particularly in a situation where there is a temporary spike in events. However, if the sustained rate is higher than the Connector and its Collector Manager can handle, the Connector eventually starts dropping messages, leading to data loss.

To avoid this situation, use the settings and statistics on the **Queue Management** tab to configure and monitor queue management. You can use the **Queue Management** tab to monitor and manage the queues for each Collector associated with the Agent Manager Event Source Server.

To manage the queues:

1. Right-click the **Agent Manager Event Source Server** in the Event Source Management view, and click **Edit.**
2. In the **Edit Event Source Server** window, click the **Queue Management** tab to display the following information:
   - **Clear:** Select one or more queues whose associated files you want to clear.

     If the Event Source Server is running, clicking **OK** immediately removes all the files stored on the disk for the selected queues. However, if the Event Source Server is not running, clicking **OK** removes all the files stored on the disk for the selected queues when the Event Source Server starts.

     WARNING: If you clear the queue, all events in the associated files are permanently deleted.

   - **Queue Name:** Displays the name of the Collector associated with the queue.
   - **File Usage:** Displays the number of files written for the associated queue.

     For example, 230/1000 denotes that messages are written to 230 files out of the specified limit of 1000 files.
   - **In/Out EPS:** Displays the rate at which messages are received and removed for the associated queue.

     For example, 300/270 denotes that the Event Source Server receives 300 messages per second and sends 270 messages per second.

     NOTE: If there is a large or a growing discrepancy between these two numbers, you should consider setting up a new Collector Manager to distribute the load.

   - **Time To Overflow:** Displays the estimated time after which the number of files reaches the maximum limit and the messages are dropped.
   - **Queue Statistics Refresh Period (in minutes):** Specify the time interval in minutes after which the Event Source Server updates the queue statistics.

NOTE: There is a delay of up to 1 minute in updating the queue statistics in the **Queue Management** tab.

---

4 Performance Test Results

The performance of the Connector can vary depending on your environment, configuration, and hardware. You might need to change the configuration of your Connector if you want to improve its performance.

Test Environment

The following hardware was used for performance testing:

Component	Number of CPUs	Operating System	CPU Model	Number of cores per CPU	Memory (RAM)
Sentinel Server	2	SLES 11 SP2 (64-bit)	Intel(R) Xeon(R) CPU E5-2680	8	128 GB
SAM 1 (Central Computer+Agent Manager+Database)	2	Windows 2008 R2 SP1 (64-bit)	Intel(R) Xeon(R) CPU x5570	4	32 GB
SAM 2 (Central Computer+Agent Manager)	2	Windows 2008 R2 SP1 (64-bit)	Intel(R) Xeon(R) CPU x5570	4	8 GB
Remote Collector Manager	2	SLES 11 SP2 (64-bit)	Intel(R) Xeon(R) CPU x5570	4	8 GB

Test Results

These test results are based on the hardware configurations mentioned above. If your environment is different, your results may differ.

With the default configuration, performance results are as follows:

- **Sentinel Agent Manager (SAM) Central Computer:**
  - Maximum number of Agents per SAM Central Computer: 2500.
  - Maximum event rate per SAM Central Computer: 3000 events per second (EPS).

    NOTE: This testing was performed using 2500 Agents which generated 10,000 event sources.
- **Connector for Agent Manager:**

  The Maximum event rate per connector when all events are dropped using a raw data filter on the Collector node: 10,000 events per second (EPS).

NOTE: This testing was performed using four SAM Central Computers.

The following table shows the maximum event rate (EPS) per Connector in the default and shared modes:

Table 4-1 *Sharing Event Sources*

Agents	Event Sources	SAM (Agent Manager + Central Computers)	Event Source Mode	Maximum Event Rate Per Connector (EPS)
2500	10,000	1	Default	2000
2500	6	1	Shared	3000

NOTE: This testing was performed using 2500 Agents, one SAM central computer, and one SAM event source server.

Scaling the Connector

The following section provides information for scalability of the Connector based on the testing done at NetIQ. It is likely that larger, more powerful, hardware configurations might handle a greater load.

To achieve high sustained event rates:

- **Enable event source sharing:** Enable event source sharing in environments with a large number of agents (5000 or higher). Event source sharing helps to minimize the number of event sources and improve event rates. For information about enabling event source sharing, see Step 15b in “Adding an Agent Manager Event Source Server” on page 13.
- **Add additional Central Computers:** The maximum number of Agents recommended per SAM Central computer is 2500. If you want to scale the number of Agents more than 2500, deploy additional Central Computers.
- **Distribute event sources:** Distribute event sources across Central Computers, Event Source Servers, Collector-Connector pairs, and Remote Collector Managers for scalability.

A Port Forwarding

Linux

To receive Agent Manager messages (events) sent to a port less than 1024, run the Agent Manager Event Source Server on a port greater than 1024 and use port forwarding.

For more information on setting up port forwarding see, the NetIQ technical support knowledge base article 3493251.

B Keystore and Truststore Creation

This section describes how to create a keystore and truststore.

Creating a Keystore

You can create a keystore by using the Java's *keytool* utility. The *keytool* utility comes with any JRE installation and is located in the `<sentinel_install_directory>/jdk/jre/bin` directory.

The following is an example of how to create a keystore:

```
keytool -genkey -alias alias -keystore .keystore
    Enter keystore password: password (Agent Manager Connector prompts for this
password while importing the keystore)
    What is your first and last name?
      [Unknown]: Test
    What is the name of your organizational unit?
      [Unknown]: Engineering
    What is the name of your organization?
      [Unknown]: Novell
    What is the name of your City or Locality?
      [Unknown]: Vienna
    What is the name of your State or Province?
      [Unknown]: VA
    What is the two-letter country code for this unit?
      [Unknown]: US
    Is <CN=Test, OU=Novell, O=Novell, L=Vienna, ST=VA, C=US> correct?
      [no]: yes
    Enter key password for <alias>
      (RETURN if same as keystore password):<Press RETURN Key>
```

The above example creates a file "`.keystore`" with a private key accompanied by the certificate for the corresponding public key. For more information on using the keytool, use the `keytool --help` command or see the *Oracle documentation*.

Creating a Truststore

The Connector includes a utility that creates a truststore from DER- or PEM-encoded certificates.

To create a truststore:

1. Extract the appropriate Connector `.zip` file to a directory.
2. Open a command prompt and browse to the directory where you extracted Connector files.
3. Run `TruststoreCreator.bat` (in Windows) or `TruststoreCreator.sh` (in Linux and Solaris)
4. Specify a file name for the truststore and press Enter.
5. Specify a password for the truststore and press Enter.
6. Specify the certificates you want to import. If there is more than one certificate to import, separate the file names with commas. Press Enter.

The TruststoreCreator utility creates a keystore file with the specified name and password, which contains the specified certificates.

C Collector Developer Notes

Collector Properties

This Connector is designed to present data in the same format as one of the following Connectors:

- File Connector
- Syslog Connector
- Windows Event (WMI) Connector

For more information about how the events are formatted for any of these connection methods, see the specific Connector documentation at the Sentinel Plug-ins Web site.

Stateful Collector Modes

It is important to set the Collector that uses data from this Connector to the correct **Stateful** mode. The values are `true` or `false`.

The Stateful property should be set to `true` if agent identification is critical to handling events properly. Agent identification might be necessary in the following situations:

- Multi-record events arrive separately at the Connector and must be combined by the Connector using information about which agent generated the data.
- Unique matching rules (based on regular expressions) are used to associate a specific agent IP address with a specific Collector.

If Stateful is set to `true`, the agent will always be included in all auto-instantiated event sources (the default mode), even if Event Source Sharing is enabled. If Stateful is improperly set to false, events may be corrupted or incorrectly parsed, in this situation. However, if these situations don't apply and event construction isn't dependent on the agent ID, Stateful should be set to false. This will enable event sharing for the Collector. If this setting is incorrectly set to true, there is a risk of unnecessary event source propagation, which can have negative effects on performance.

For more information about event source sharing, see “Event Source Creation and Sharing” on page 8. For more information about Stateful Collector Modes, see Step 15c on page 16.

D Revision History

Revision: 2020.1r1

February 2020.1r1

This version of the Connector includes the following enhancement and software fix:

- You can now match the AppID of an event to a regular expression either in the Sentinel Agent Manager Connector or the respective Collector, when configuring data collection based on matching application IDs.
- When the upstream Sentinel Agent Manager server was down, the Connector was caching events till the cache was full. The older queued events were dropped from cache and this resulted in loss of events.

  There is no data loss now. The Sentinel Agent Manager server now has the ability to cache events. Therefore, it does not send events when the Connector queue is full. (`Bug 1052362`)

Revision: 2017.1r1

June 2017

This version of the Connector includes the following software fixes:

- Fix for compatibility issues with Sentinel 8.1 and later. (`BUG 1027051`)
- The Subject Alternative Name (SAN) extension field is now added to the Connector's internal (default) certificate so that the Security Agent for UNIX for Sentinel can communicate with the Connector using both hostname and IP address. (`BUG 988943`)
- The subject name, issued to, and issued by fields in the Connector's internal (default) certificate have been corrected so that the Security Agent for UNIX for Sentinel can communicate seamlessly. (`BUG 962707`)
- The Connector now sets the `CONNECTION_MODE` property for events properly when the Collector parsing the events supports multiple connection modes. (`BUG 880564`)

Revision: 2011.1r3

May 2014

What's New?

The Connector fixes the following issue:

The Agent Manager Fails to Connect with Sentinel Running in FIPS Mode

Issue: Sentinel 7.1.2 includes Oracle Java 1.7 update 51, which has a known issue related to RSA client key exchange in FIPS mode (http://www.oracle.com/technetwork/java/javase/7u51-relnotes-2085002.html). This issue causes connection problems between Sentinel Agent Manager and Connector for Agent Manager. For more information about this issue, see the Sentinel 7.1.2 Release Notes.(BUG 876808)

Fix: The Connector now allows you to configure the exclusion list for key exchange algorithms, which are used to connect to Sentinel. The RSA client key exchange ciphers are removed from the exclusion list. Therefore, the Connector for Agent Manager now successfully connects to Sentinel, versions 7.1.2 and later, running in FIPS mode.

Revision: 2011.1r2

April 2014

What's New?

This version includes the following enhancements:

- The Connector now receives events directly from the NetIQ Security Agent for UNIX 7.4 and later. You no longer need to install Sentinel Agent Manager to receive events from this source.
- The Connector now allows you to minimize the number of event sources by consolidating them based on the connection method for each Collector by using the **Shared Event Sources** option.
- The Connector now supports the Unique Matching Rule which helps you to route the incoming events to the appropriate Collectors.
- Includes several enhancements that improve the overall performance and scalability of the Connector.

Software Fixes

The Connector provides software fixes for the following issues:

The Connector for Agent Manager Auto-Configures Event Sources For Each Event Provider and Application

Issue: The Connector for Agent Manager auto-configures event sources for each event provider and application type in each Agent. This results in a large number of event sources per Agent.(BUG 831031)

Fix: The Connector for Agent Manager now auto-configures only one event source per Agent for each Collector and therefore reduces the number of auto-configured event sources.

You can reduce the number of event sources further by using the **Shared Event Sources** option. For more information about **Shared Event Sources**, see "Event Source Creation and Sharing" on page 8.

Revision: 2011.1r1

April 2013

New Connector.

E Known Issues

NetIQ Corporation strives to ensure our products provide quality solutions for your enterprise software needs. The following issue is currently being researched. If you need further assistance with any issue, please contact Technical Support.

Internal Certificate of the Connector Has Expired in Non-FIPS Mode

Issue: When Sentinel is running in non-FIPS mode, the internal certificate of the Connector is expired and Sentinel is not able to receive events from the agents. (Bug 986791)

Workaround: Perform the following steps to regenerate the Connector's internal certificate:

1. Launch the ESM from the SCC.
2. Right-click and edit the Agent Manager Event Source Server.
3. In the **Security** tab, change the **Server Key Pair Settings** option from **Internal** to **Custom** and click **OK**.
4. Repeat Step 2.
5. In the **Security** tab, edit the same Agent Manager Connector again and change the **Server Key Pair Settings** option from **Custom** to **Internal** and click **OK.**

   This will generate the new internal certificate with one year validity.

Connector in FIPS Mode Does Not Work With Security Agent for UNIX

Issue: When Sentinel server and Collector Manager is in FIPS mode, the Connector will use the Sentinel web server's certificate. Since the Sentinel web server certificate does not have the SAN (Subject Alternative Name) extension field, Security Agent for UNIX fails to communicate with the Connector. The IP address or hostname used in Security Agent for UNIX and the subject name of the web server certificate differs. (Bug 997589)

Workaround: Perform the following steps on the Sentinel server and remote Collector Manager:

1. Log in to the Sentinel server as `root` user.
2. Stop the Sentinel service:

   ```
   rcsentinel stop
   ```
3. Generate a key pair with the alias **webserver** in a JKS format key store using the following command:

   For Sentinel server:

   ```
   /opt/novell/sentinel/jdk/jre/bin/keytool -genkey -dname
   cn=<distinguished_name> -alias webserver -validity <validity_period_in_days> -
   storetype JKS -keystore <keystore_name> -storepass <keystore_password> -keypass
   <key_password> -keyalg RSA -ext san=dns:<dns>,ip:<ip>
   ```

For Collector Manager:

```
/opt/novell/sentinel/jdk/jre/bin/keytool -genkey -dname
cn=<distinguished_name> -alias client -validity <validity_period_in_days> -
storetype JKS -keystore <keystore_name> -storepass <keystore_password> -keypass
<key_password> -keyalg RSA -ext san=dns:<dns>,ip:<ip>
```

Example: `/opt/novell/sentinel/jdk/jre/bin/keytool -genkey -dname cn=sentinel-server.acme.com -alias webserver -validity 365 -storetype JKS -keystore sam_connector.jks -storepass password -keypass password -keyalg RSA -ext san=dns:sentinel-server.acme.com,ip:1.2.3.4`

4. Convert the key pair from JKS format to PKCS12 format:

   For Sentinel server:

   ```
   /opt/novell/sentinel/jdk/jre/bin/keytool -noprompt -importkeystore -
   srcstorepass <source_keystore_password> -deststorepass
   <destination_keystore_password> -srckeystore sam_connector.jks -srcalias
   webserver -destkeystore <destination_keystore_name> -deststoretype PKCS12
   ```

   For Collector Manager:

   ```
   /opt/novell/sentinel/jdk/jre/bin/keytool -noprompt -importkeystore -
   srcstorepass <source_keystore_password> -deststorepass
   <destination_keystore_password> -srckeystore sam_connector.jks -srcalias
   client -destkeystore <destination_keystore_name> -deststoretype PKCS12
   ```

   NOTE: You must use the same password that you used in Step 3.

   Example: `/opt/novell/sentinel/jdk/jre/bin/keytool -noprompt -importkeystore -srcstorepass password -deststorepass password -srckeystore sam_connector.jks -srcalias webserver -destkeystore sam_connector.p12 -deststoretype PKCS12`

5. Delete the **webserver** key pair from the Sentinel FIPS keystore database. For Collector Managers, delete both the server and client certificates:

   For Sentinel server versions prior to 8.1:

   ```
   /usr/bin/certutil -D -n "webserver" -d <FIPS_keystore_database _path>
   ```

   For Sentinel server 8.1 and later:

   ```
   /usr/bin/certutil -D -n "webserver" -d sql:<FIPS_keystore_database _path>
   ```

   For Collector Manager versions prior to 8.1:

   ```
   /usr/bin/certutil -D -n "server" -d <FIPS_keystore_database _path>
   ```

   ```
   /usr/bin/certutil -D -n "client" -d <FIPS_keystore_database _path>
   ```

   For Collector Manager 8.1 and later:

   ```
   /usr/bin/certutil -D -n "server" -d sql:<FIPS_keystore_database _path>
   ```

   ```
   /usr/bin/certutil -D -n "client" -d sql:<FIPS_keystore_database _path>
   ```

   Example: `/usr/bin/certutil -D -n "webserver" -d sql:/etc/opt/novell/sentinel/3rdparty/nss/`

6. Insert the key pair from the PKCS12 keystore (created in Step 4) into the Sentinel FIPS keystore database:

   For Sentinel server:

   ```
   /usr/bin/pk12util -d <FIPS_keystore_database _path> -i <p12_keystore_name> -W
   <p12_keystore_password> -K <FIPS_keystore_database_password> -n webserver
   ```

   For Collector Manager:

```
/usr/bin/pk12util -d <FIPS_keystore_database _path> -i <p12_keystore_name> -W
<p12_keystore_password> -K <FIPS_keystore_database_password> -n client
```

NOTE: The argument to –W must be the same as the argument to –deststorepass in Step 4.

The argument to –K is the current password for the Sentinel FIPS keystore database that you chose when you set up FIPS mode.

Example: `/usr/bin/pk12util -d /etc/opt/novell/sentinel/3rdparty/nss/ -i sam_connector.p12 -W password -K Password@123 -n webserver`

7. Convert the key pair created in Step 3 from JKS format to .cer:

   For Sentinel server:

   ```
   /opt/novell/sentinel/jdk/jre/bin/keytool -export -keystore <keystore_name> -
   alias webserver -file <sentinel.cer>
   ```

   For Collector Manager:

   ```
   /opt/novell/sentinel/jdk/jre/bin/keytool -export -keystore <keystore_name> -
   alias client -file <sentinel.cer>
   ```

   Example: `/opt/novell/sentinel/jdk/jre/bin/keytool -export -keystore sam_connector.jks -alias webserver -file sentinel.cer`

8. Import the Collector Manager certificate generated in Step 7 to Sentinel server and Sentinel server certificate to Collector Manager.

   ```
   /opt/novell/sentinel/bin/convert_to_fips.sh -i <sentinel.cer>
   ```

   Follow the onscreen instructions and restart sentinel services.