NetWitness® Platform XDR

Version 12.1.0.0

Log Collection Configuration Guide

NETWITNESS

Platform XDR

Contact Information

NetWitness Community at https://community.netwitness.com contains a knowledge base that answers common questions and provides solutions to known problems, product documentation, community discussions, and case management.

Trademarks

License Agreement

Third-Party Licenses

Note on Encryption Technologies

Distribution

October, 2022

Contents

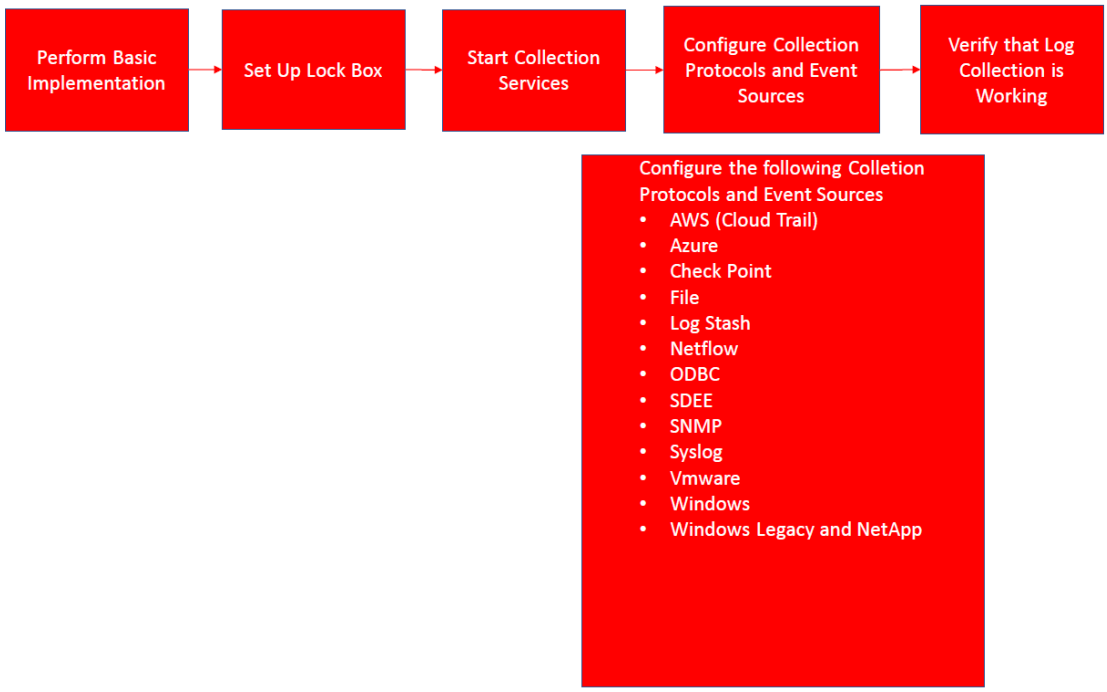
About Log Collection

This guide describes the high-level steps and subtasks for setting up and configuring log collection for event sources that include:

- What Log Collection does, how it works from a high level, and provides high-level deployment diagrams.
- How to start collecting events.
- Where to find instructions to set up more complex deployments.
- How to start any collection protocol.
- Which tools to use to troubleshoot Log Collection issues and lists global troubleshooting instructions.
- How to fine tune and customize Log Collection in your environment.
- How to configure individual collection protocols. Instructions are in the individual Log Collection sections.

Workflow

This workflow depicts the basic tasks needed to start collecting events through Log Collectors.



High-Level Procedures

At a high level, these are the procedures you must follow for log collection:

I. Add local and remote collectors to NetWitness.

   Set up a Log Collector locally on a Log Decoder (that is a Local Collector). You can also set up Log Collectors in as many remote locations (that is Remote Collectors) as you need for your enterprise. For details, see Basic Implementation.

II. Download the latest content from NetWitness Live. You must perform this task periodically, as the content provided on NetWitness Live is updated regularly.

   Log Collection content is marked as one of the following resource types:

   - **NetWitness Log Collector** - content enabling the collection of event source types.
   - **NetWitness Log Device** - the latest supported event source parsers.

   You can also subscribe to content on Live. For details, see the *Live Services Management Guide*.

III. Configure Settings: set up the lockbox and Certificates.

   For details, see Set Up a Lockbox and Configure Certificates.

IV. Configure Event Sources.

   You configure all the event sources on your network to send their log information to NetWitness. Whenever you add new event sources, you need to perform this procedure as well. All event source configuration guides are found in the NetWitness Supported Event Sources space in NetWitness Link.

V. Start and stop services for configured protocols. Occasionally, you may be required to stop and restart services, based on new event sources that you add to NetWitness.

VI. Verify that Log Collection is working.

   Whenever you set up a new event source or add a new collection protocol, you should verify that the correct logs are being sent to NetWitness.

Log Collection Architecture

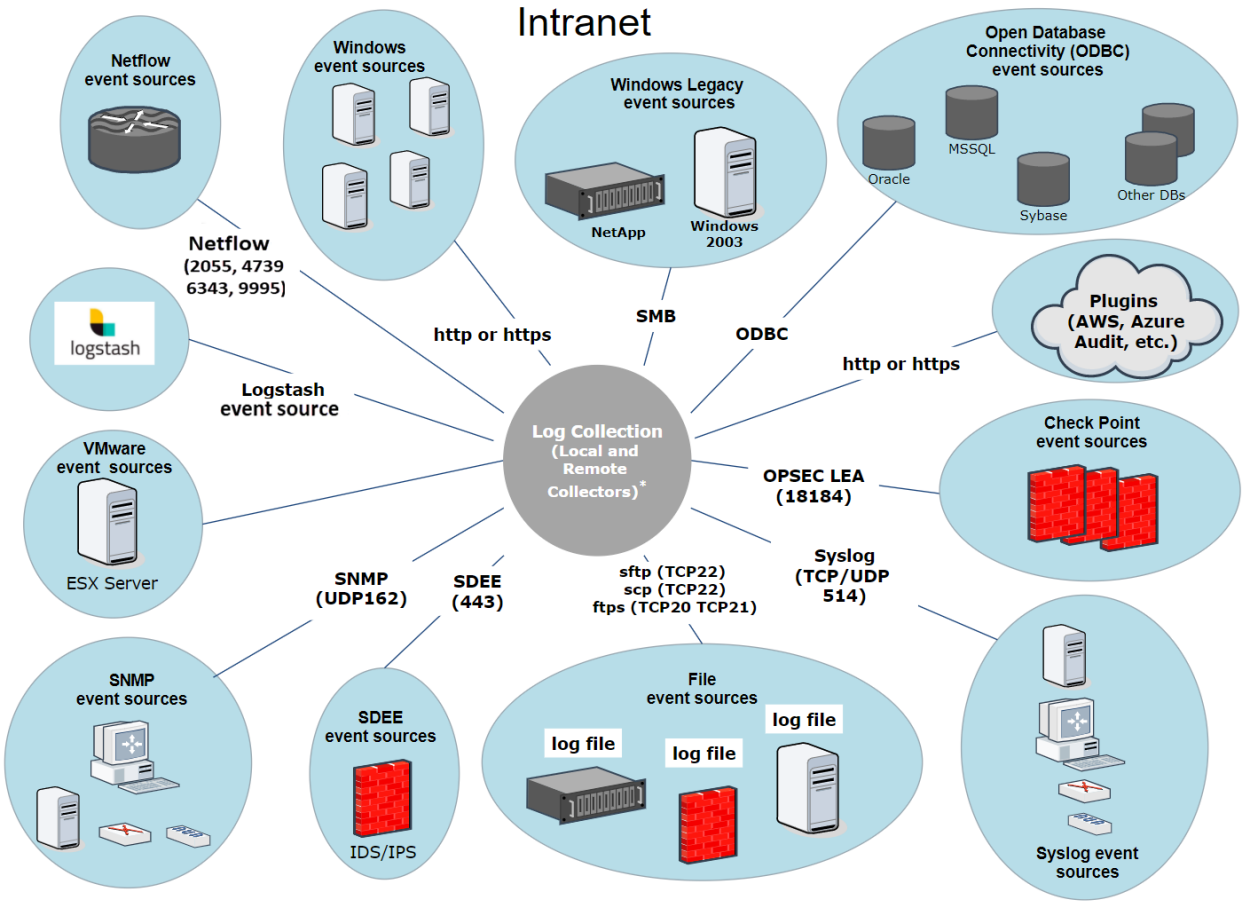
This topic describes how NetWitness performs log collection.

How to Deploy Log Collection

You can deploy Log Collection according to needs and preferences of your enterprise. This includes deploying Log Collection across multiple locations and collecting data from varying sets of event sources. You do this by setting up a Local Collector with one or many Remote Collectors.

Components of Log Collection

The following figure shows all the components involved in event collection through the NetWitness Log Collector.



*In Log Collection, Remote Collectors send events to the Local Collector and the Local Collector sends events to the Log Decoder.

Local and Remote Collectors

In this scenario, log collection from various protocols like Windows, ODBC, and so on, is performed through both the Remote Collector and Log Collector service. If the log collection is done by the Local Collector, it is forwarded to the Log Decoder service, just like the local deployment scenario. If the log collection is done by a Remote Collector, there are two methods in which these are transferred to the Local Collector:

- **Pull Configuration** - From a Local Collector, you select the Remote Collectors from which you want to pull events.
- **Push Configuration** - From a Remote Collector, you select the Local Collector to which you want to push events.

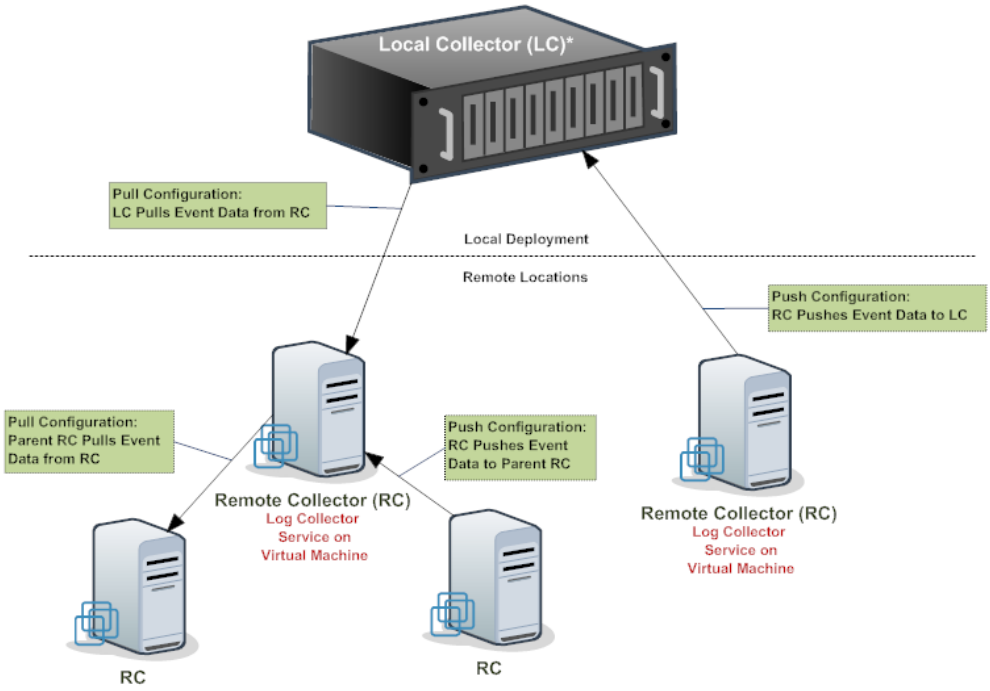
Note: Typically, the Push configuration is used. Pull is available if you have a DMZ in your environment. Less secure network segments are not allowed to make connections to more secure network segments. With Pull, the Log Collector (or Virtual Log Collector) in the secure network initiates the connection to the VLC in the less secure network, and the logs are then transferred without breaking the connection rules.

You can configure one or more Remote Collectors to push event data to a Local Collector, or you can configure a Local Collector to pull event data from one or more Remote Collectors.

Additionally, you can set up a chain of Remote Collectors for which you can configure:

- One or more Remote Collectors to push event data to a Remote Collector.
- A Remote Collector to pull event data from one or more Remote Collectors.

The following figure illustrates how the Local and Remote Collectors interact to collect events from all of your locations.



* The Local Collector (LC) is the Log Collector service on the Log Decoder appliance.

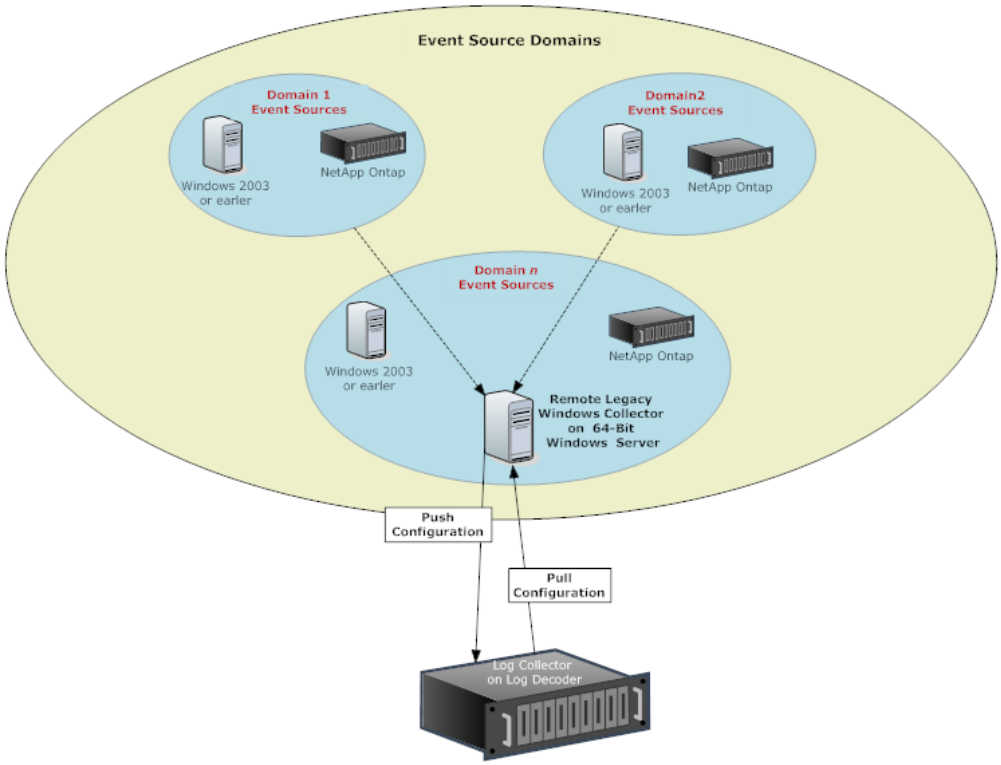
Windows Legacy Remote Collector

The NetWitness Windows Legacy Collector is a Microsoft Windows based remote log collector (RC) which can be installed on a Windows domain.

It supports collection from:

- Windows 2003 and earlier event sources
- NetApp ONTAP host evt files

The following figure illustrates the deployment required to collect events from Windows Legacy event sources.


Event Source Domains
Domain 1
Event Sources
Windows 2003
or earlier
NetApp Ontap
Domain2
Event Sources
Windows 2003
or earlier
NetApp Ontap
Domain n
Event Sources
Windows 2003
or earlier
NetApp Ontap
Remote Legacy
Windows Collector
on 64-Bit
Windows Server
Push
Configuration
Pull
Configuration
Log Collector
on Log Decoder

Basic Implementation

This topic tells how to perform the initial setup of Local Collectors and Remote Collectors.

Prerequisites

Verify that the Log Decoder is set up and:

- is capturing data.
- has the current content loaded.
- is properly licensed.

Roles of Local and Remote Collectors

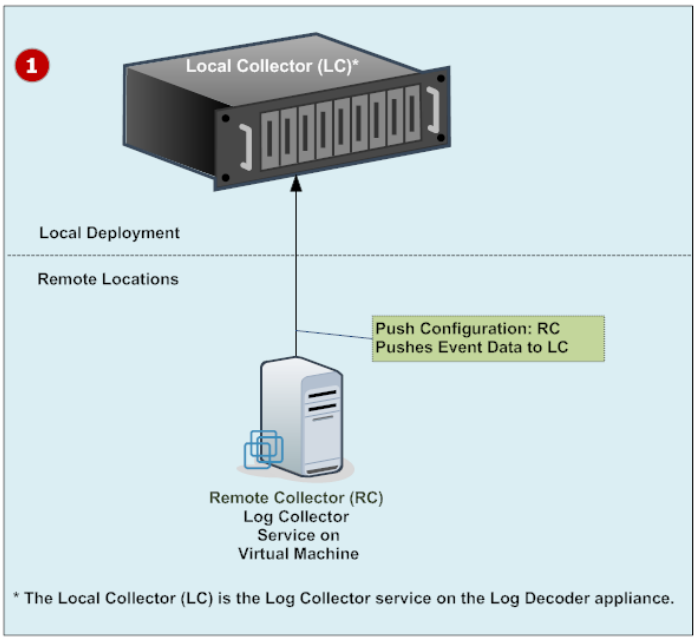
A Local Collector (LC) is a Log Collector service running on a Log Decoder host. In a local deployment scenario, the Log Collector service is deployed on a Log Decoder host, with the Log Decoder service. Log collection from various protocols like Windows, ODBC, and so on, is performed through the Log Collector service, and events are forwarded to the Log Decoder service. The Local Collector sends all collected event data to the Log Decoder service.

You must have at least one Local Collector to collect non-Syslog events.

A Remote Collector (RC), also referred to as a Virtual Log Collector (VLC), is a Log Collector service running on a stand-alone Virtual Machine. Remote Collectors are optional and they must send the events they collect to a Local Collector. Remote Collector deployment is ideal when you have to collect logs from remote locations. Remote Collectors compress and encrypt the logs before sending them to a Local Collector.

Deploying and Configuring Log Collection

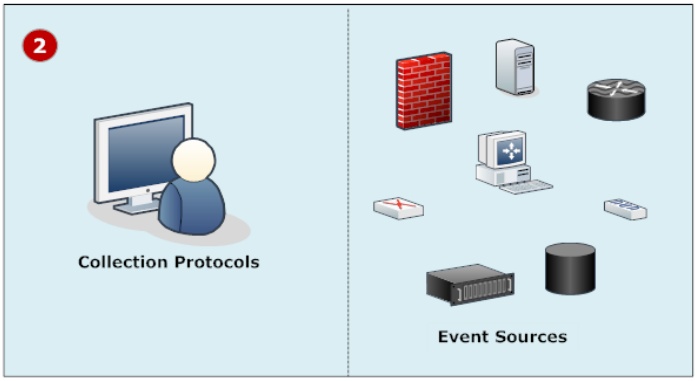
The following diagram illustrates the basic tasks you must complete to deploy and configure Log Collection. To deploy Log Collection, you need to set up a Local Collector. You can also deploy one or more Remote Collectors. After you deploy Log Collection, you need to configure the events sources in NetWitness and on the events sources themselves. The following diagram shows the Local Collector with one Remote Collector that pushes events to the Local Collector.



1 Set up Local and Remote Collectors.

The Local Collector is the Log Collector service running on the Log Decoder host.

A Remote Collector is the Log Collector service running on a virtual machine or Windows server in a remote location.



2 Configure event sources:

- Configure collection protocols
- Configure each event source to communicate with the NetWitness Log Collector.

For details on these procedures, see Configure Collection Protocols and Event Sources.

Adding Local Collector and Remote Collector to NetWitness

To add a Local Collector and Remote Collector to NetWitness:

1. Go to **(Admin) > Services**.
2. Click and select **Log Collector** from the menu.
   The **Add Service** dialog box is displayed.
3. Define the details of the **Log Collection** service.
4. Select **Test Connection** to ensure that your Local or Remote Collector is added.

Configuring Log Collection

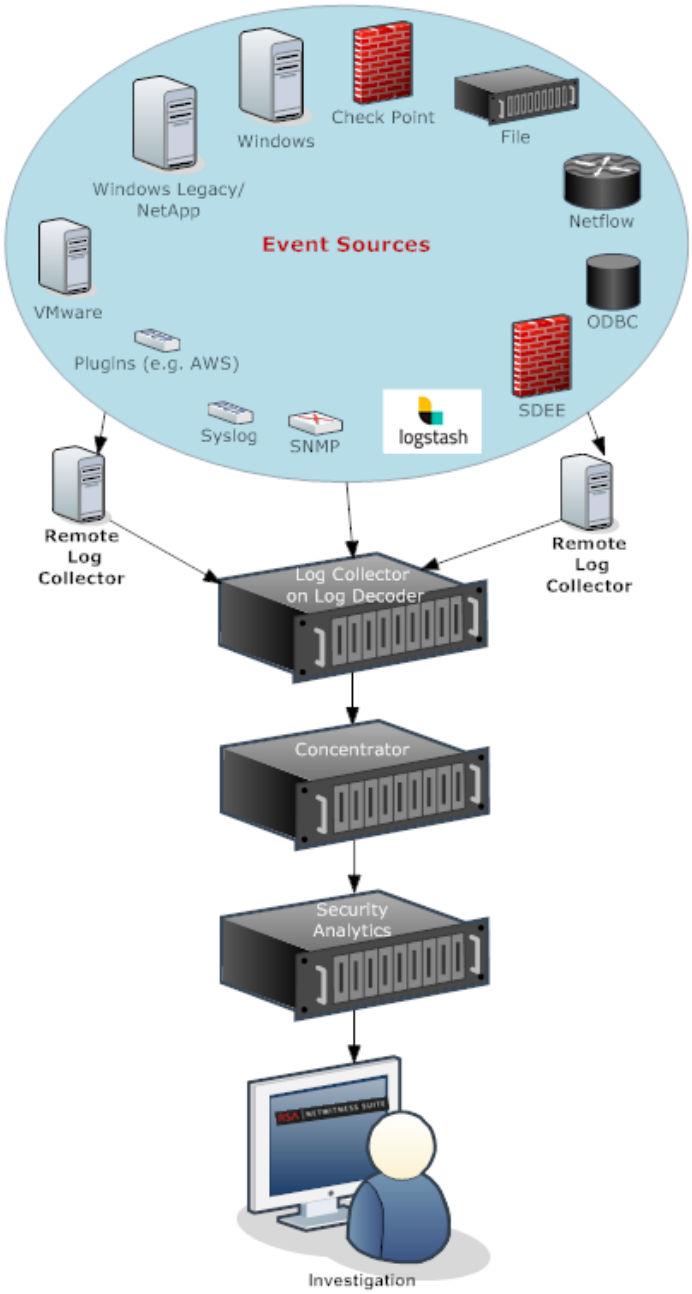
You choose the Log Collector—that is a Local Collector (LC) or Remote Collector (RC)—for which you want to define parameters in the Services view. The following figure shows how to navigate to the Services view, select a Log Collector service, and display the configuration parameter interface for that service.

To configure log collection:

1. Go to **(Admin) > Services**.
2. Select a Log Collection service.
3. Click **View > Config** to display the Log Collection configuration parameter tabs.
4. Define global Log Collection parameters in the **General** tab.
5. The UI presents tabs, depending on whether the current service is Local or Remote.
   - For a Local Collector, NetWitness displays the **Remote Collectors** tab. Select the Remote Collectors from which the Local Collector pulls events in this tab.
   - For a Remote Collector, NetWitness displays the **Local Collectors**. Select the Local Collectors to which the Remote Collector pushes events in this tab.
6. Edit configuration files as text files in the **Files** tab.
7. Define collection protocol parameters in the **Event Sources** tab.
8. Define the lockbox, encryption keys, and certificates in the **Settings** tab.
9. Define Appliance Service parameters in the **Appliance Service Configuration** tab.

Data Flow Diagram

You use the log data collected by the Log Collector service to monitor the health of your enterprise and to conduct investigations. The following figure shows you how data flows through NetWitness Log Collection to Investigation.



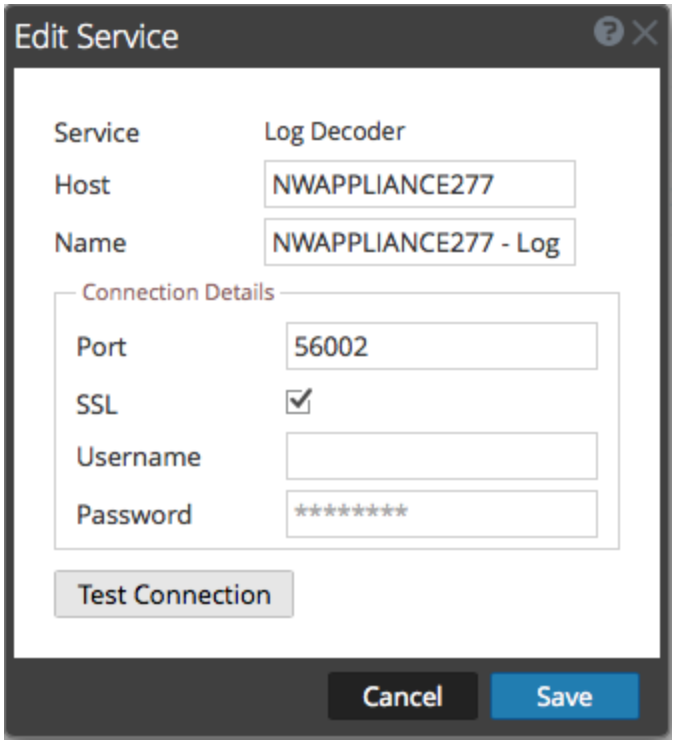
Provision Local Collectors and Remote Collectors

The NetWitness server verifies if an appliance has a Log Decoder service. If there is a Log Decoder service, it becomes a Local Collector. If a Log Decoder service is missing, it becomes a Remote Collector. A local Log Collector has an Event Destination and by default goes to the Local Log Decoder service. A Remote Collector does not have an Event Destination. The NetWitness Server identifies a Legacy Windows Collector as a Remote Collector.

To edit a Local Collector or Remote Collector:

1. Go to **(Admin) > Services**.
2. Select a Log Collector service.
3. In the **Services** view, select in the toolbar.

   The **Edit Service** dialog is displayed.



4. In the **Edit Service** dialog, provide the following information.

Field	Description
Service	Select Log Collector as the service type.
Host	Select a Log Decoder host.
Name	Type name you want to assign to the service.

Field	Description
Port	Default port is 50001 for clear text and 56001 for SSL encrypted.
SSL	Select **SSL** if you want NetWitness to communicate with the host using SSL. The security of data transmission is managed by encrypting information and providing authentication with SSL certificates.
(Optional) Username	Type the username of the Local Collector.
(Optional) Password	Type the password of the Local Collector.

5. Click **Test Connection** to determine if NetWitness connects to the service.
6. When the result is successful, click **Save**.

   If the test is unsuccessful, edit the service information and retry.

Configure Local and Remote Collectors

This topic describes how to configure Local and Remote Collectors.

When you deploy Log Collection, you must configure the Log Collectors to collect the log events from various event sources, and to deliver these events reliably and securely to the Log Decoder service, where the events are parsed and stored for subsequent analysis.

You can configure one or more Remote Collectors to push event data to a Local Collector, or you can configure a Local Collector to pull event data from one or more Remote Collectors.

This topic describes how to:

- **Configure Local Collector to Pull Events from Remote Collector**

  If you want a Local Collector to pull events from Remote Collector, you set this up in the Remote Collectors tab of the Local Collector's Configuration view.

- **Configure Remote Collector to Push Events to Local Collectors**

  If you want a Remote Collector to push events to a Local Collector, you set this up in the Local Collector tab of the Remote Collector's Configuration view. In the Push configuration, you can also:

  - **Configure Failover Local Collector for Remote Collector**

    You set up a destination made up of local collectors. When the primary Local Collector is unreachable, the Remote Collector attempts to connect to each Local Collector in this destination until it makes a successful connection.

  - **Configure Replication**

    You set up multiple destination groups so that NetWitness replicates the event data in each group. If the connection to one of the destination groups fails, you can recover the required data because it is replicated in the other destination group.

  - **Configure Log Routing for Specific Protocols**

    You set up multiple destinations in a destination group to direct event data to specific locations according to protocol type.

- **Configure Chain of Remote Collectors**

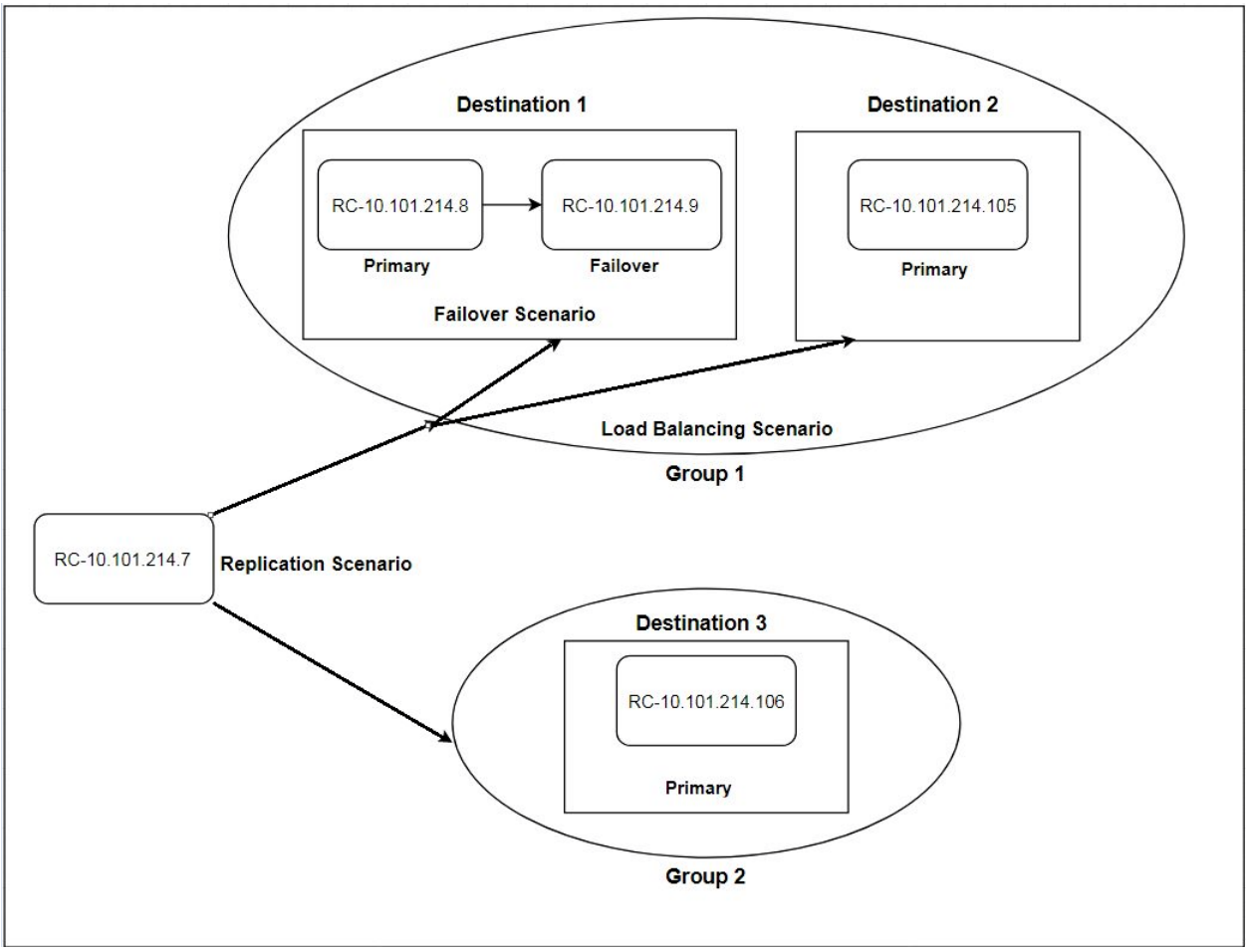
You can set up a chain of Remote Collectors to push event data to a Local Collector, or you can configure a Local Collector to pull event data from a chain of Remote Collectors.

- You can configure one or more Remote Collectors to push event data to a Remote Collector.
- You can configure a Remote Collector to pull event data from one or more Remote Collectors.

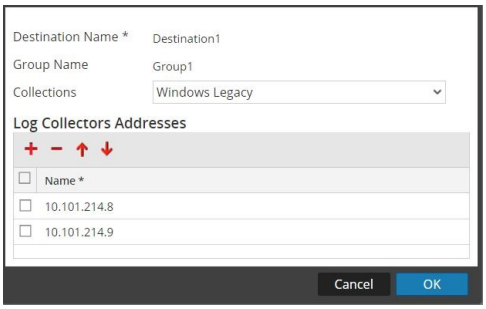
Failover, Replication and Load Balancing

This section describes failover, replication, and load balancing work in how NetWitness.

The following figure illustrates a Remote Collector configured for load balancing, failover and replication.

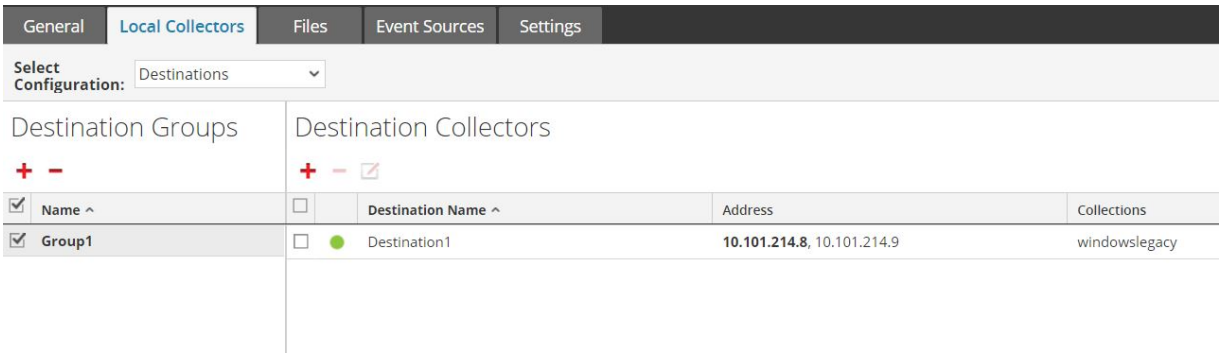


- **Failover** is achieved by setting up multiple collectors in the same Destination. Destination 1 has a primary Collector, and second, failover Collector. This is done in NetWitness by adding multiple Log Collectors to the same Destination.

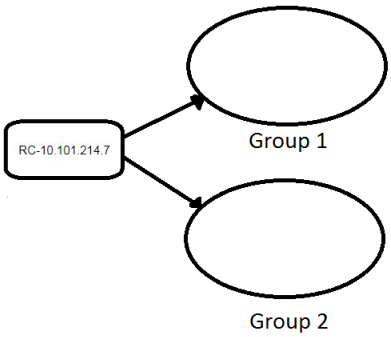


Since 10.101.214.8 is listed first, that becomes the primary collector, and 10.101.214.9 becomes the failover. To make 10.101.214.9 the primary, use the up arrow to change the order.

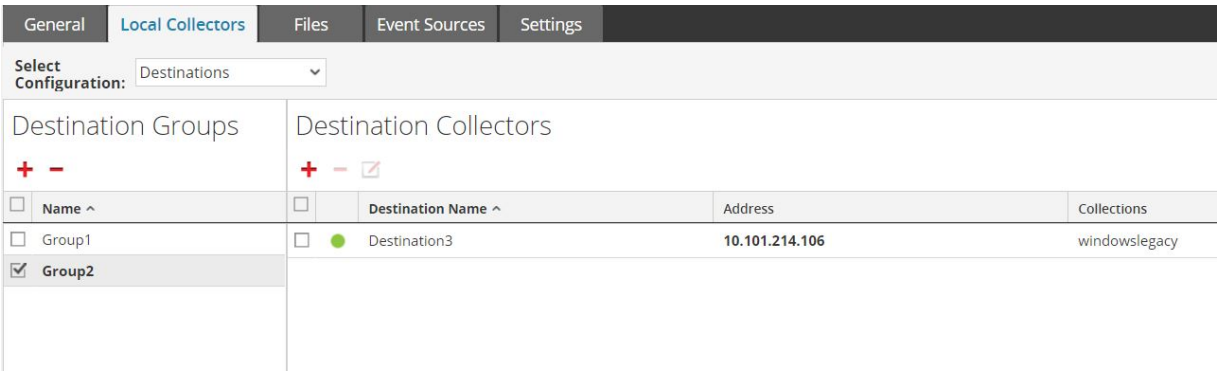
Below, you can see the two collectors both listed for Destination 1. The primary (10.101.214.8) is in bold.



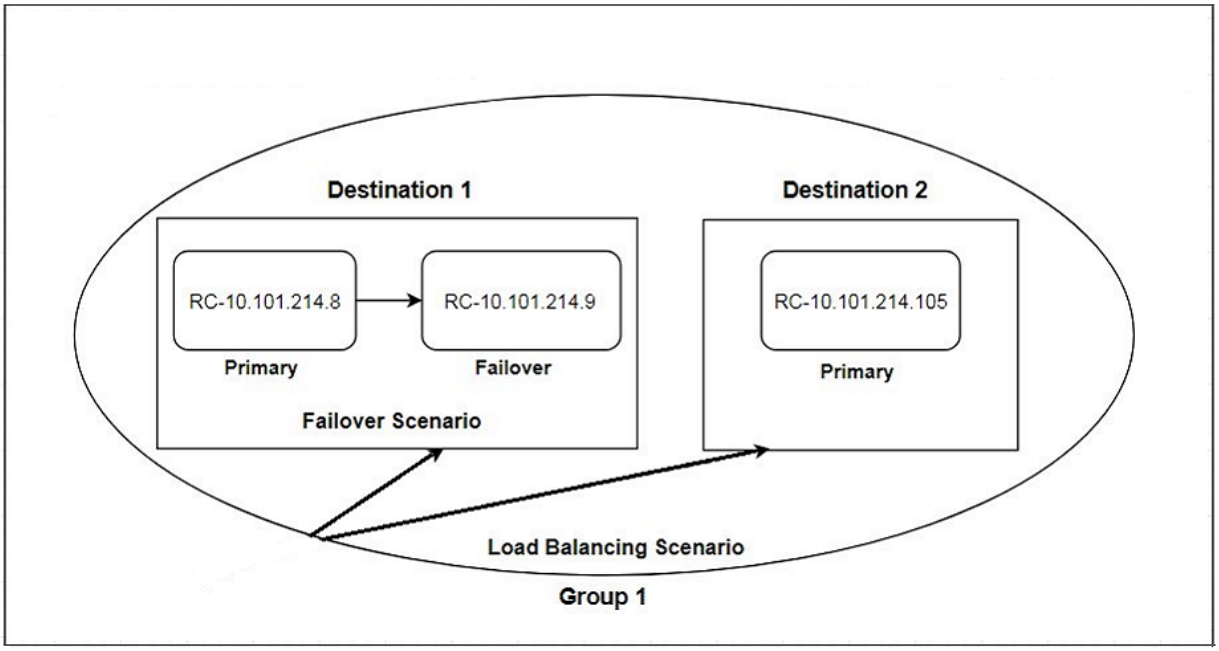
- **Replication** is accomplished by having multiple Destination Groups: each group receive the entire set of message data.



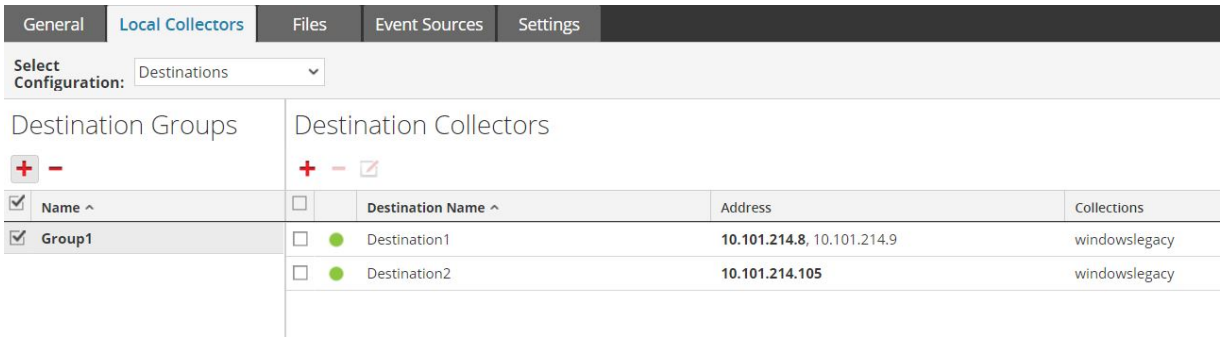
In the following screen, you can see that message data is sent to the collectors in Group 1 *and* Group 2.



- **Load balancing** is achieved by setting up multiple Destinations within a Group.



In the following screen, you can see that Group 1 has two destinations, Destination 1 and Destination 2. The message data is divided up equally among the Destinations in the group.



With two Destinations, each destination receives half the message data. With three Destinations, each would receive 1/3 of the total message data. Keep adding Destinations to further reduce the load on the collectors in each destination.

Note: You can also set up log routing so that event data for specific protocols is sent to specific destinations.

Configure a Local Collector or Remote Collector

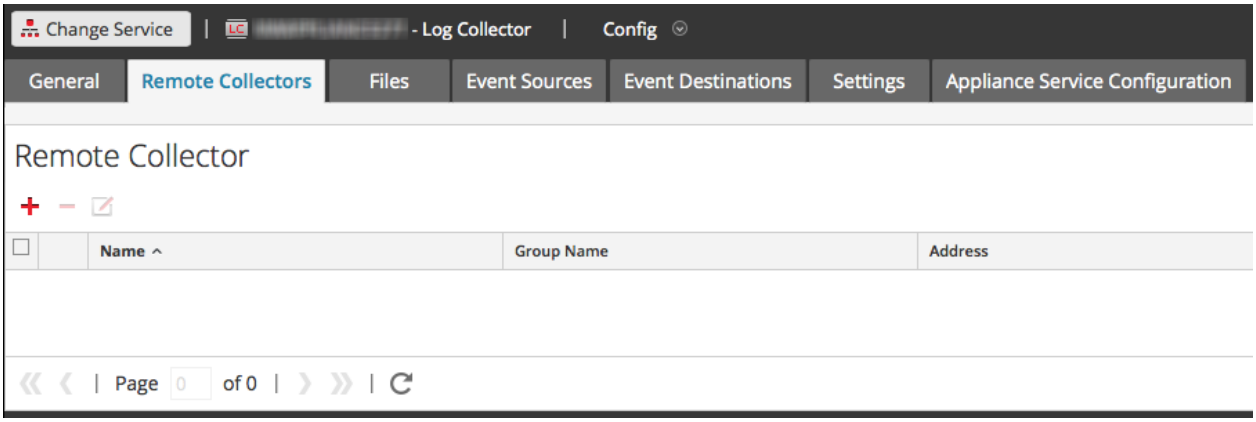
You choose the Log Collector, that is a Local Collector (LC) or Remote Collector (RC), for which you want to define deployment parameters in the Services view. The following procedure shows you how to navigate to the Services view, select a Local or Remote Collector, and display the deployment parameter interface for that service.

To configure a Local Collector or Remote Collector:

1. Go to **(Admin) > Services**.
2. Select a Local or Remote Log Collection service.
3. Under Actions, select **> View > Config** to display the Log Collection configuration parameter tabs.
4. Depending on your selection in step 2:
   - If you selected a Local Collector, the **Remote Collectors** tab is displayed. Select the Remote Collectors from which the Local Collector pulls events in this tab.
   - If you selected a Remote Collector, the **Local Collectors** are displayed. Select the Local Collectors to which the Remote Collector pushes events in this tab.

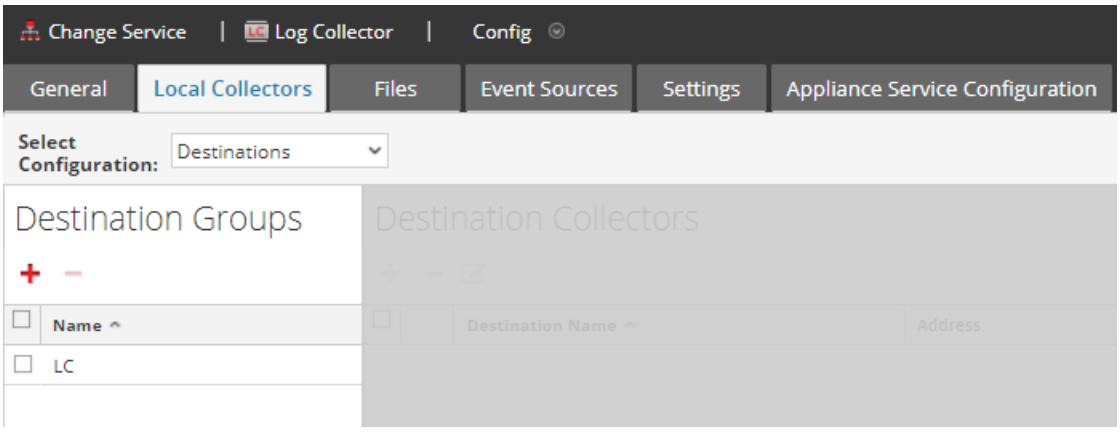
Remote Collectors Tab

The following figure depicts the **Remote Collectors** tab for a Local Collector that is configured to pull events from a Remote Collector. NetWitness displays this tab when you have selected a Local Collector in (Admin) > Services.

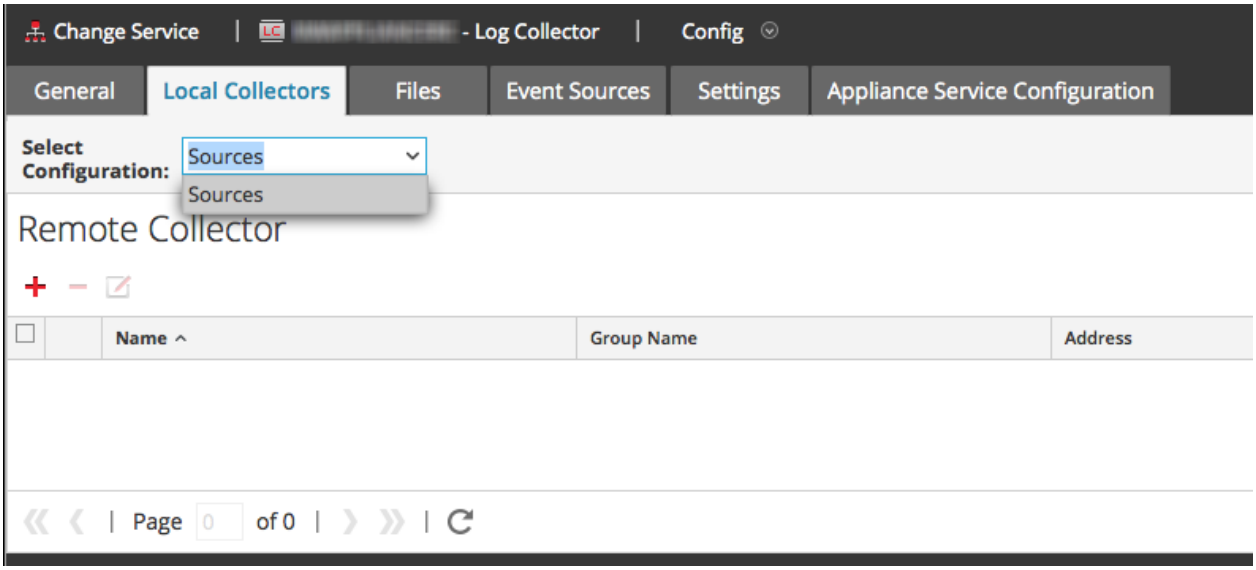


Local Collectors Tab for a Remote Collector

The following figure depicts a **Local Collectors** tab for a Remote Collector that is configured to push events to a Local Collector or another Remote Collector.



The following figure depicts the Local Collectors tab for a Remote Collector that is configured to pull events from a Remote Collector. NetWitness displays this tab when you have selected a Remote Collector in (Admin) > Services.



Parameters

Remote/Local Collectors Configuration Parameters

Configure Failover Local Collector

This topic tells you how to set up a Failover Local or Remote Collector.

Set up a Failover Local Collector

You can set up a Failover Local Collector that NetWitness will fail over to if your primary Local Collector stops operating for any reason.

1. Go to  **(Admin) > Services**.
2. In **Services**, select a Remote Collector service.
3. Click  **View > Config**.

   The Service Config view is displayed with the **Log Collector General** tab open.
4. Select the **Local Collectors** tab.
5. In the **Destination Groups** panel section, select  .

   The Add Remote Destination dialog displays.
6. Set up a Destination Group and select a primary Local Collector (for example, **LC-PRIMARY**).
7. Select the Group (for example, **Primary_Standby_LCs**) in the Destination Groups panel and click  .

   The Group you selected is displayed in the Local Collectors panel.
8. Add the Failover Local Collector (for example, **LC-STANDBY**).

   The following examples show the newly added primary and failover Local Collectors showing the primary Local Collector as **Active** and the Failover Local Collector as **Standby**. The active Local Collector is highlighted (for example, **LC-PRIMARY**).
9. (Optional) Add, delete, and change the order of Local Collectors to each Remote Destination.

   a. Click  to add a Log Collector as a failover Remote Destination.

   b. When connecting to a Remote Destination, the Remote Collector will attempt to connect to each Local Collector in this list in order, until it makes a successful connection.

   c. Select a Local Collector and use the up () and down () arrow buttons to change the order of connection.

   d. Select one or more Local Collectors and click to remove them from the list.

   The selected Local Collectors are added to the Log Collector section. When the Remote Collector starts collecting data, it pushes data to these Log Collectors.

Set up a Failover Remote Collector

You can set up a Failover Remote Collector that NetWitness will fail over to if your primary Remote Collector stops operating for any reason.

To set up a failover remote collector:

1. Go to  **(Admin) > Services**.
2. In **Services**, select a Remote Collector service.
3. Click  **View > Config**.

   The Service Config view is displayed with the **Log Collector General** tab open.

4. Select the **Local Collectors** tab.
5. Select **Sources** in **Select Configuration** drop-down menu.
6. Click + to display in **Add Source** dialog.
7. Define the failover Remote Collector and click **OK**.

Parameters

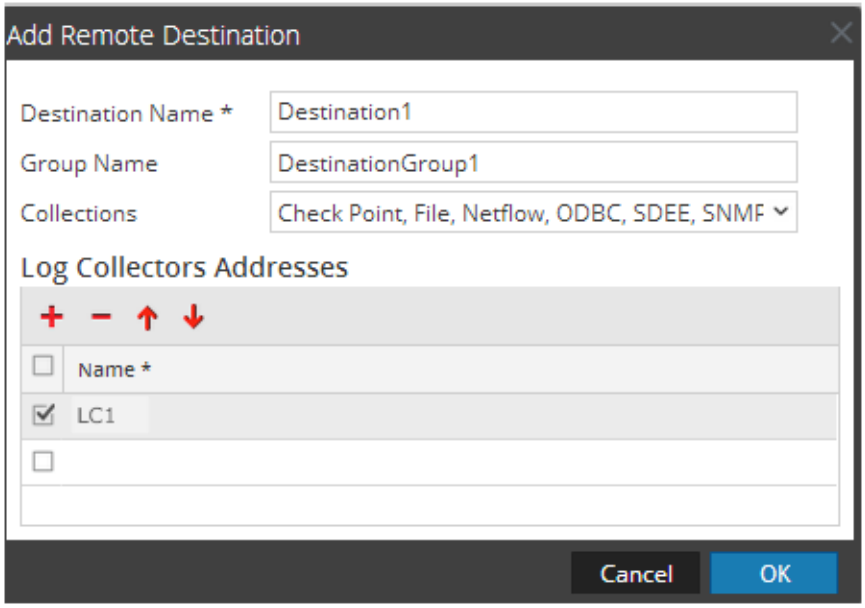
Remote/Local Collectors Configuration Parameters

Configure Replication

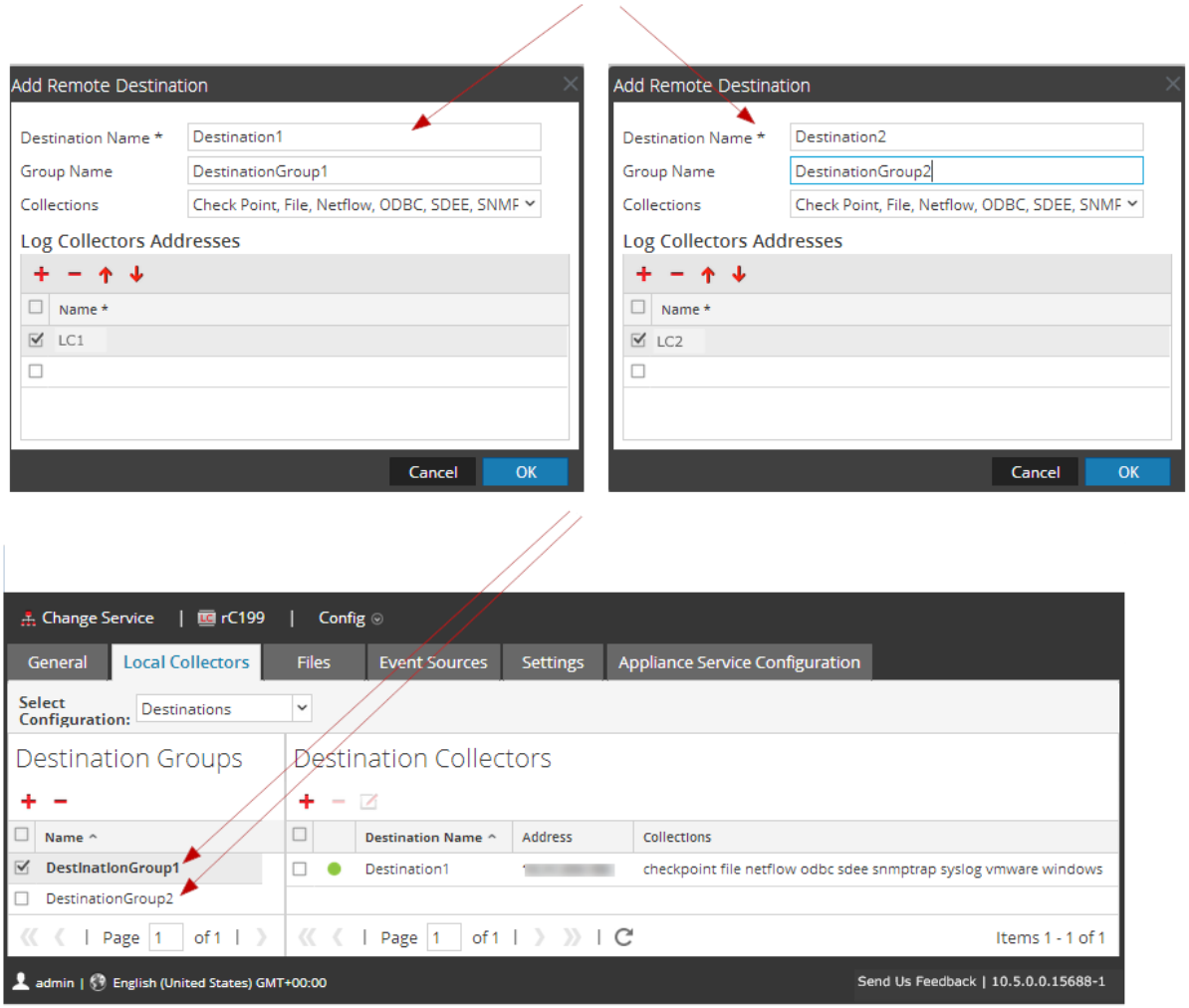
This topic tells you how to replicate event data sent by a Remote Collector.
You can specify multiple Destination Groups so that the event data is replicated to each group.

To replicate event data to multiple Local Collectors:

1. Go to **(Admin) > Services.**
2. Select a Remote Log Collection service.
3. Under Actions, select **> View > Config**.
   The Service Config view is displayed with the **Log Collector General** tab open.
4. Select the **Local Collectors** tab.
5. In the **Destination Groups** panel section, click +.
   The **Add Remote Destination** dialog is displayed.



6. Set up a separate Destination for each Local Collector and designate the protocols for which you want to push event messages to that Local Collector. The following examples shows the addition of two Destination Local Collectors (**Destination1** and **Destination2**) for the **Check Point**, **File**, **Netflow**, **ODBC**, **SDEE**, **SNMP**, **Syslog**, and **Windows** collection protocols:



   a. Type the **Destination Name**.

   b. Type the **Group Name**. If you do not type a Group Name, the Destination Name is taken as the Group Name.

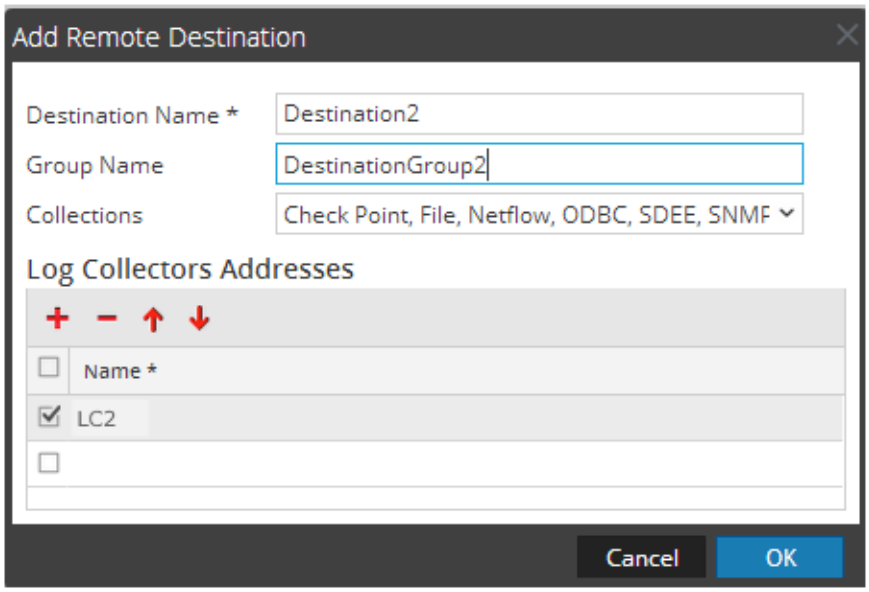
   c. Select the collection protocols in the drop-down list.

   d. Select a Local Collector (for example, **LC1**).

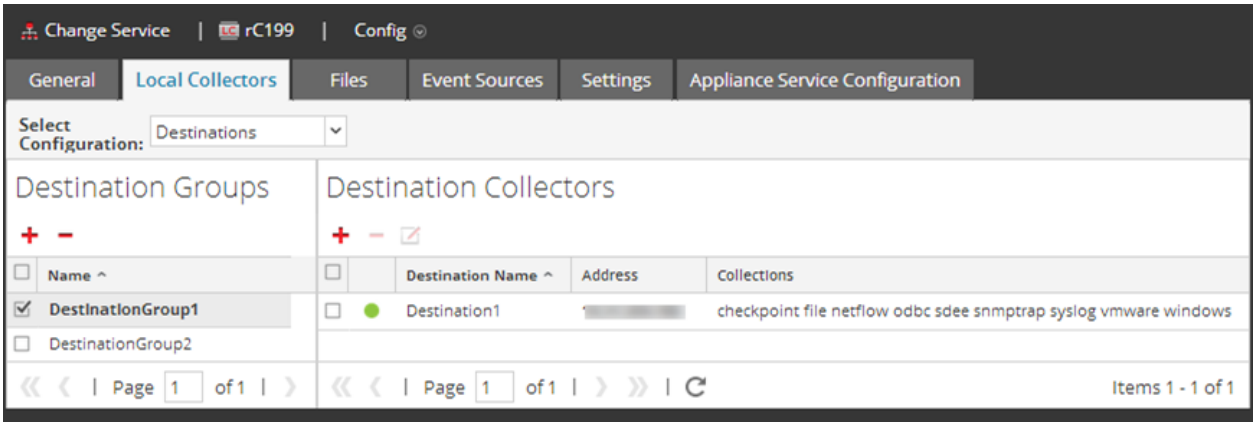
   e. Click **OK**.

   f. Select the new group (for example, **DestinationGroup2**) group in the **Destination Groups** panel and click + in the **Local Collector** panel.

g. In the **Local Collector** panel, click + and complete the **Add Remote Destination** dialog as illustrated in the following figure.



The **Check Point**, **File**, **Netflow**, **ODBC**, **SDEE**, **SNMP**, **Syslog**, and **Windows** collection protocols are sent to two Local Collectors (**LC1** and **LC2**). Both Local Collectors are active and collecting event data.



Configure Chain of Remote Collectors

This topic describes how to chain Remote Collectors (also referred to as VLCs).

You can set up a chain of Remote Collectors to push event data to a Remote Collector, or you can configure a Remote Collector to pull event data from a chain of Remote Collectors.

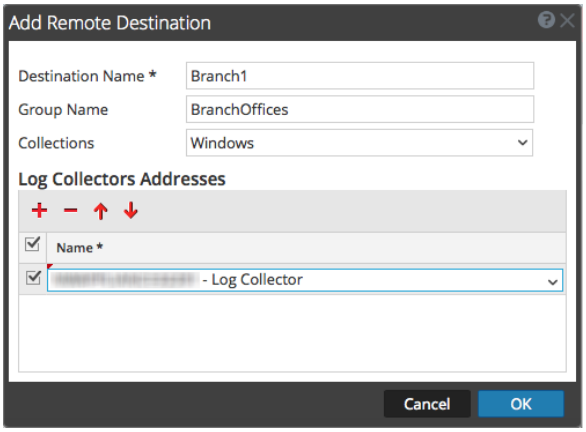
- **Remote Collectors to push data.** Push data from a Remote Collector to other Remote Collectors or Local Collectors.
- **Remote Collector to pull data.** Use a Remote Collector to pull data from one or more Remote Collectors.

Configure Remote Collector to Push Event Data to Remote Collector

1. Go to **(Admin) > Services.**
2. In **Services**, select a **Remote Collector**.
3. Under Actions, select **> View > Config** to display the Log Collection configuration parameter tabs.

   The **Log Collector Service Config** view is displayed with the **Log Collector General** tab open.
4. Select the **Local Collectors** tab.
5. Select **Destinations** in the **Select Configurations** drop-down menu.
6. In the **Destination Groups** panel section, select .

   The **Add Remote Destination** dialog is displayed.
7. Set up a **Destination Group**:
   a. Enter a **Destination Name**.
   b. (Optional) **Enter a Group Name**. If you leave Group Name blank, NetWitness sets it to the value that you specified in Destination Name.
   c. Select one or more collection protocols in the **Collections** drop-down list.
   d. Under **Log Collectors Addresses**, click to select a Remote Collector.



Note: If you do not select a collection protocol, the Remote Collector pushes all collection protocols to the Remote Collectors.

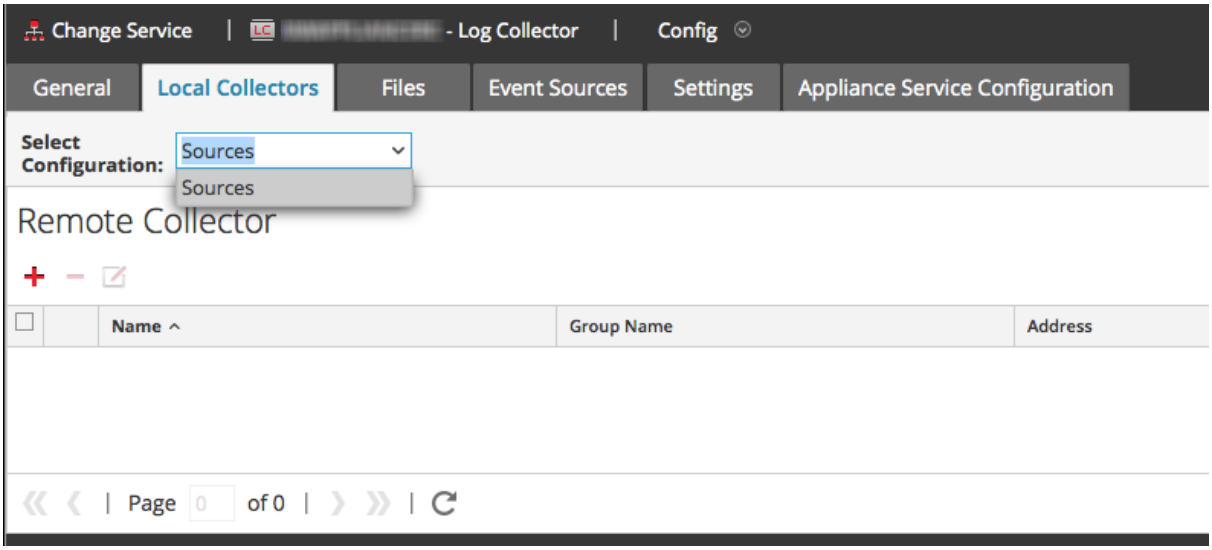
Configure Remote Collector to Pull Event Data from a Remote Collector

1. Go to **(Admin) > Services.**
2. In **Services**, select a **Remote Collector**.

3. Under Actions, select  > **View** > **Config** to display the Log Collection configuration parameter tabs.

   The **Service Config** view is displayed with the **Log Collector General** tab open.

4. Select the **Local Collectors** tab.
5. Select **Sources** in the **Select Configurations** drop-down menu.



6. In the **Remote Collectors** panel, select +.

   The **Add Source** dialog is displayed.

7. In the **Add Source** dialog:

   a. Select one or more collection protocols.

      If you do not select a collection protocol, the Remote Collector pulls all collection protocols from the Remote Collector.

   b. Click **OK**.

   The Remote Collector is added to the Remote Collector section. When the Log Collector starts collecting data, it pulls event data from this Remote Collector.

Throttle Remote Collector to Local Collector Bandwidth

To improve performance, you can throttle the bandwidth to control the rate that the Remote Collector sends event data to Local Collector or between Message Brokers. To do this, you configure the Linux kernel's filtering and IPTable functionality.

This works for both push and pull Remote Collector configurations. The **set-shovel-transfer-limit.sh** shell script located on the **/opt/netwitness/bin** automates the configuration of the kernel filter and iptables related to this port.

This topic describes how to throttle Remote Collector to Local Collector bandwidth using the **set-shovel-transfer-limit.sh** shell script. It contains the following sections:

- The **set-shovel-transfer-limit.sh** shell script command line help.

  Note: The filter value that you need to set depends on the rate at which remote log collector is sending events to the Local Collector.

- An example that sets the Filter to 4096 kilobits per second.

Command Line Help for Set Shovel Transfer Limit Script

Issue the `-h` command to display help for `set-shovel-transfer-limit.sh` shell script.

```
cd /opt/netwitness/bin
./set-shovel-transfer-limit.sh
```

Usage:

```
set-shovel-transfer-limit.sh -s|-c|-d|[-i interface] [-r rate]
```

where:

- -c = clear existing
- -d = display filter
- -s = set new values
- -i = interface is the name of the network interface. Default value is **eth0**
- -r = rate is the bandwidth rate. Default value is **256kbps**

Bandwidths and rates can be specified in:

- **nolimit**: disables throttling
- **kbit**: Kilobits per second
- **mbit**: Megabits per second
- **kbps**: Kilobytes per second
- **mbps**: Megabytes per second
- **bps**: Bytes per second

Set the Filter to 4096 Kilobits per Second

This example sets the Filter to 4096 kilobits per second.

```
[root@<hostname> bin]#./set-shovel-transfer-limit.sh -s -r 4096kbit

RATE=4096kbit
 PORTNUMBER=5671
 DEVICE_INTERACE=eth0

iptables: No chain/target/match by that name.
iptables: No chain/target/match by that name.
iptables: Saving firewall rules to /etc/sysconfig/iptables:[ OK ]

Current/new values...

iptables -t mangle -n -v -L
 Chain PREROUTING (policy ACCEPT 2 packets, 161 bytes)
 pkts bytes target  prot opt in  out  source         destination
```

```
Chain INPUT (policy ACCEPT 2 packets, 161 bytes)
 pkts bytes target prot opt in out   source          destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target prot opt in out   source          destination

Chain OUTPUT (policy ACCEPT 2 packets, 248 bytes)
 pkts bytes target prot opt in out   source          destination
     0    0 MARK   tcp -- *    eth0     0.0.0.0/0     0.0.0.0/0     multiport
dports 5671 MARK set 0xa
     0    0 MARK   tcp -- *    eth0     0.0.0.0/0    0.0.0.0/0    multiport
sports 5671 MARK set 0xa

Chain POSTROUTING (policy ACCEPT 2 packets, 248 bytes)
 pkts bytes target prot opt in out   source          destination

tc -s -d class show dev eth0
 class htb 1:1 root rate 10000Kbit ceil 10000Kbit burst 1600b/8 mpu 0b
overhead 0b cburst 1600b/8 mpu 0b overhead 0b level 7
 Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
 rate 0bit 0pps backlog 0b 0p requeues 0
 lended: 0 borrowed: 0 giants: 0
 tokens: 20000 ctokens: 20000

class htb 1:2 parent 1:1 prio 0 quantum 51200 rate 4096Kbit ceil 4096Kbit
burst 1599b/8 mpu 0b overhead 0b cburst 1599b/8 mpu 0b overhead 0b level 0
 Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
 rate 0bit 0pps backlog 0b 0p requeues 0
 lended: 0 borrowed: 0 giants: 0
 tokens: 48828 ctokens: 48828
```

Set Up a Lockbox

What Is a Lockbox

A lockbox is an encrypted file that you use to store confidential information about an application. The NetWitness Lockbox stores an encryption key for the Log Collector.

The encryption key is used to encrypt all event source passwords and the event broker password.

When you create the Lockbox, you need to define a password for the Lockbox.

The Log Collector operates the Lockbox in a mode during data collection that does not require you to specify the password (the Log Collector uses the host system fingerprint instead).

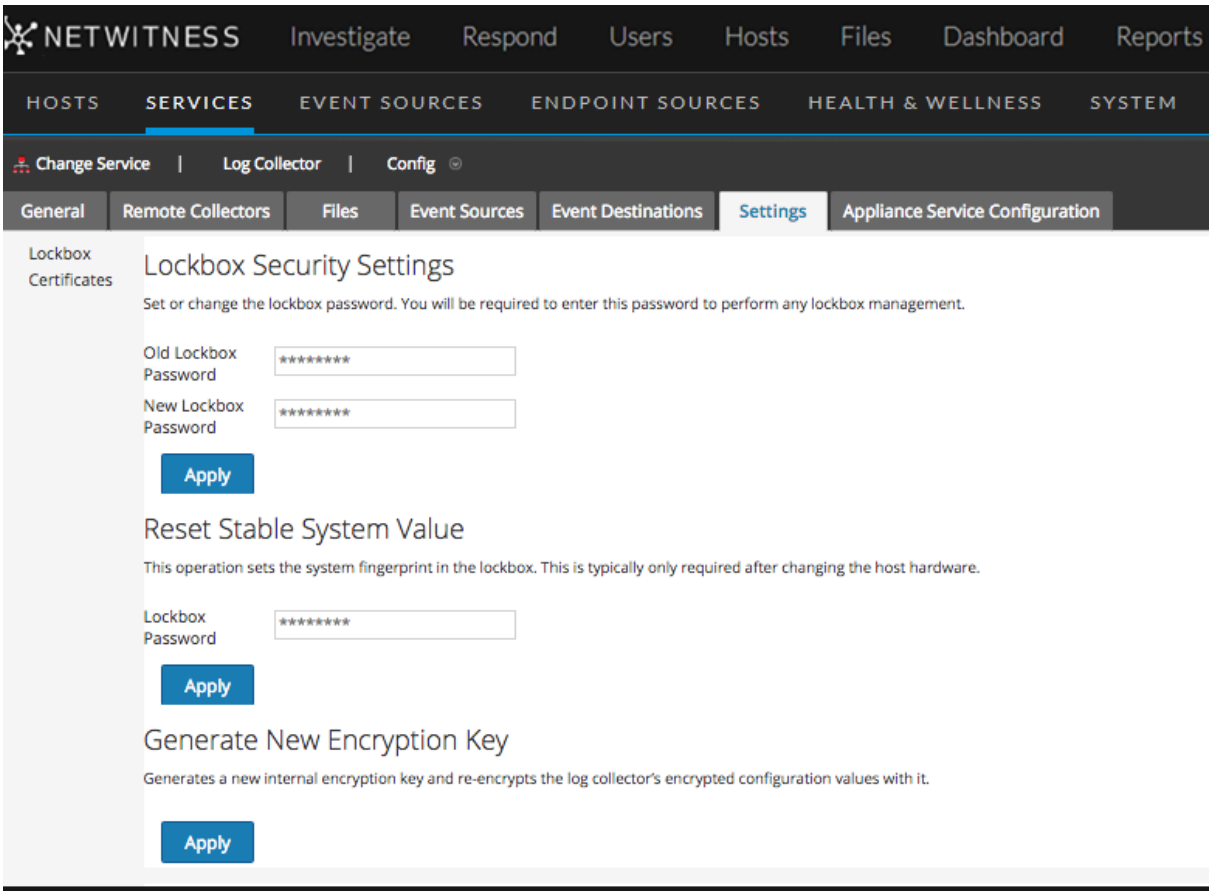
These are the lockbox security settings.

Feature	Description
Old Lockbox Password	When you set up a Lockbox for the first time, this field is blank. NetWitness populates this field after you enter a New Lockbox Password and click Apply.
New Lockbox Password	Initial or new lockbox password. To maximize lockbox security, specify a password that is eight or more characters in length with at least one numeric character, uppercase character, and non-alphanumeric character such as # or !
Apply	Click **Apply** to save the changes to the lockbox password.

Set Up a Lockbox

To set up a lockbox you need to set a password, as follows:

1. Go to **(Admin) > Services**.
2. Select a Log Collection service.
3. Under Actions, select **> View > Config** to display the Log Collection configuration parameter tabs.
4. Click the **Settings** tab.



5. In the options panel, select **Lockbox** to configure Lockbox settings.
6. Under **Lockbox Security Settings**, enter a password in the **New Lockbox Password** field and click **Apply**.

Start Collection Services

If a collection service stops, you may need to start it again. You can also enable the automatic start of collection services.

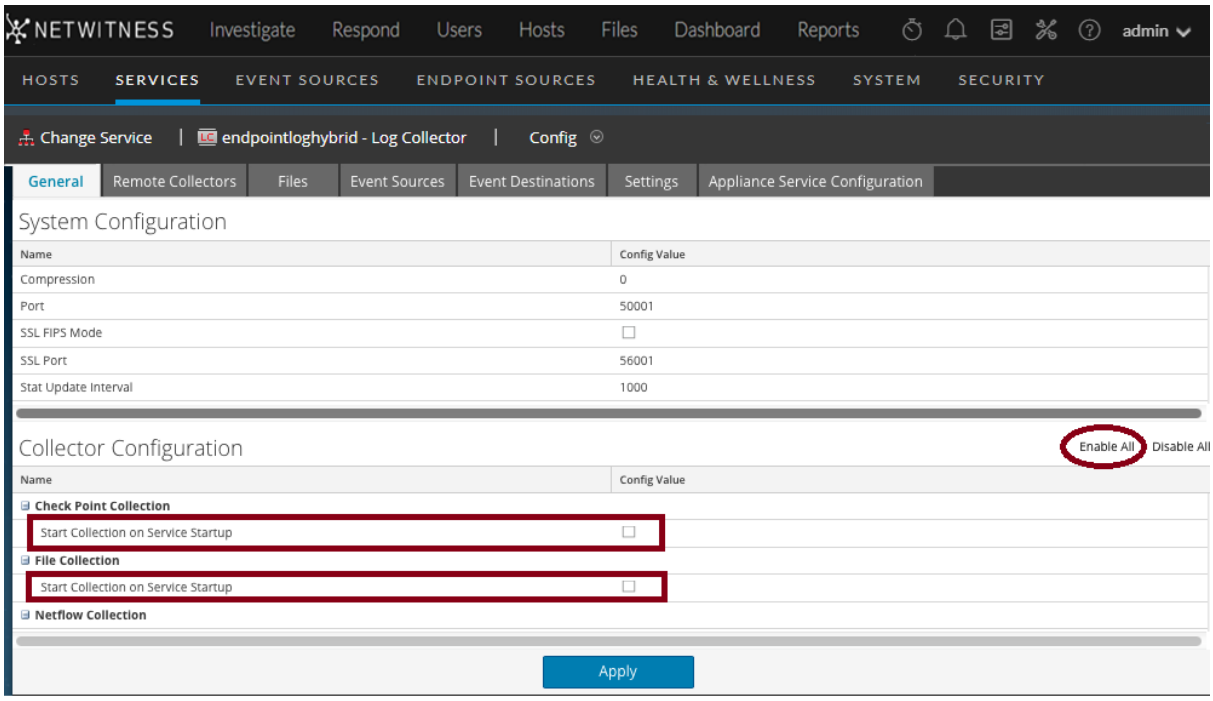
Start a Collection Service

1. Go to (Admin) > Services.
2. Select a Log Collector service and click under **Actions**.
3. Click **View** > **System**.
4. Click **Collection** > ***service*** (for example **File**) and click **Start**.

Enable Automatic Start of Collection Services

1. Go to (Admin) > Services.
2. Select a Log Collector service and click under **Actions**.
3. Click **View** > **Config**.

   The General tab is displayed.
4. In the Collector Configuration panel, select **Start Collection on Service Startup** for the individual collection services that you want to start automatically. Alternatively, select **Enable All** to automatically start all collection services.



5. Click **Apply** for your changes to take effect.

Verify That Log Collection Is Working

This topic tells you how to verify that you have set up Log Collection correctly.

The following methods verify that Log Collection is working.

- Verify that there is event activity the Event Source Monitoring tab of the **Administration > Health & Wellness** view.
- Verify that there are parsers in the **device.type** field in the **Details** column in the **Investigation > Events** view for the collection protocol you configured.

Please refer to the topics for each Collection Protocol for steps on how to verify that the protocol is set up correctly.

Configure Certificates

You manage certificates by creating trust stores on the Log Collector. The Log Collector refers to these trust stores to determine whether or not the event sources are trusted.

Note: You can configure custom certificates for one or more event sources: for details, see (Optional) Configure Custom Certificates on Log Collectors.

Add a Certificate

To add a certificate:

1. Go to  **(Admin) > Services**.
2. In the **Services** grid, select a **Log Collector** service.
3. Click  under **Actions** and select **View > Config**.
4. Click the **Settings** tab.
5. In the options panel, select **Certificates**.
6. Click  in the **Certificates** tool bar.
   The **Add Cert** dialog is displayed.
7. Click **Browse** and select a certificate (***.PEM**) from your network.
8. Specify a password (if required).
9. Click **Save**.

Certificates Panel

The following table describe the buttons and columns available in the Certificates panel.

Field	Description
	Opens the Add Cert dialog in which you can add a certificate and password.

Field	Description
	Deletes the selected certificates.
	Selects certificates.
Trust Store Name	Displays the name of the trust store.
Certificate Distinguished Name	For Check Point event source only, displays the distinguished name for the certificate.
Certificate Password Name	For Check Point event source only, displays the password name for the certificate.

Add Cert Dialog

The following table describes the parameters available in the **Add Cert** dialog.

Field	Description
Trust Store Name	Enter a trust store name.
File	Click Browse to select a certificate (*.PEM file) file from your network
Password	Specify the password for this certificate.
Close	Closes the dialog without adding a certificate.
Save	Adds the certificate.

(Optional) Configure Custom Certificates on Log Collectors

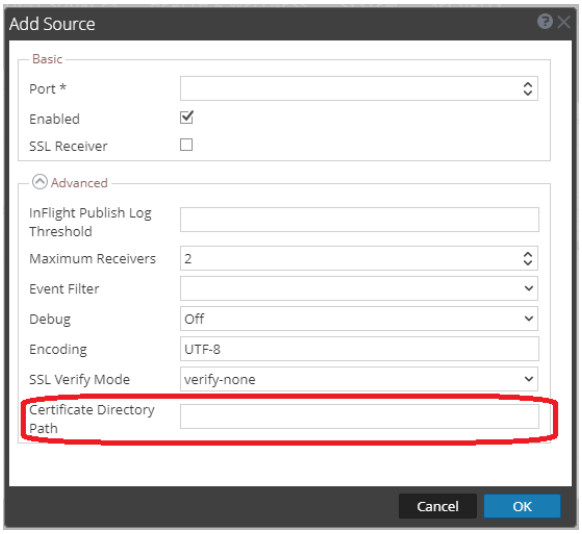
You can configure custom certificates for the syslog listener on Log Collectors. This enables you to put your own trusted certificate in place on the syslog listener for specific event sources, while all other functionality uses the pre-installed certificates.

To configure custom certificates for an event source:

1. Upload the custom certificate and key files onto the Log Collector or Virtual Log Collector, and save them into a folder. You need to add this path information in step 4 below.
2. Go to **(Admin) > Services**, select a Log Collector service and **> View > Config**.
3. Select the **Event Sources** tab, then choose **Syslog** from the drop-down menu.
4. Add (or select) a **syslog-tcp** event source type.
5. In the **Sources** pane, add (or edit) a source.

   The Add Source dialog box is displayed.
6. In the Add Source dialog box, click the Advanced section toggle.

   The Advanced parameters are displayed.
7. For the **Certificate Directory Path**, enter the pathname for the folder that contains the certificate files.



   The Log Collector SSL syslog connections will use the `logcollector_cert.pem` and `logcollector_key.pem` files in the folder specified.
8. Press **OK** to save all parameters that you added or changed.
9. Restart syslog collection for the changes to take effect.

Note: More than one event source can use the same certificates: for each event source that shares the certificate, specify the same **Certificate Directory Path**. However, the specified path cannot contain more than one certificate (that is, one cert and one key file). To use a different custom certificate, you must specify a different **Certificate Directory Path**.

Log Collection Basics

How Log Collection Works

The Log Collector service collects logs from event sources throughout the IT environment in an organization and forwards the logs to other NetWitness components. The logs and the descriptive content are stored as meta data for use in investigations and reports.

Event sources are the assets on the network, such as servers, switches, routers, storage arrays, operating systems, and firewalls. In most cases, your Information Technology (IT) team configures event sources to send their logs to the Log Collector service and the NetWitness administrator configures the Log Collector service to poll event sources and retrieve their logs. As a result, the Log Collector receives all logs in their original form.

Collection Protocols

NetWitness can collect logs from a wide variety of event sources. When you are configuring log collection for a specific event source, you need to know, first and foremost, the protocol that is used to collect the logs.

Collection Protocol	Description
Check Point	Collects events from Check Point event sources using OPSEC LEA. OPSEC LEA is the Check Point Operations Security Log Export API that facilitates the extraction of logs. For details, see Configure Check Point Event Sources in NetWitness.
File	Collects events from log files. Event sources generate log files that are transferred using a secure file transfer method to the Log Collector service. For details, see Configure File Event Sources in NetWitness. **Note:** In NetWitness 11.4 and later, you can perform File Log collection for many event sources using Endpoint Agents, thus simplifying the collection process. For details, see the *NetWitness Endpoint Configuration Guide*. For a list of which event sources are supported, see the section "Currently Supported File Log Event Source Types."
Netflow	Accepts events from Netflow v5 and Netflow v9. For details, see Configure Netflow Event Sources in NetWitness.
ODBC	Collects events from event sources that store audit data in a database using the Open Database Connectivity (ODBC) software interface. For details, see Configure ODBC Event Sources in NetWitness.

Collection Protocol	Description
Plugins	The Plugins collection is a generic collection framework for collecting events using external scripts written in other languages. NetWitness currently provides collection for Amazon Web Services (AWS) CloudTrail and Microsoft Azure. • **AWS**: Collects events from Amazon Web Services (AWS) CloudTrail. Specifically CloudTrail records AWS API calls for an account. For details, see Configure AWS (CloudTrail) Event Sources in NetWitness • **Azure**: Collects events from Microsoft Azure. For details, see Configure Azure Event Sources in NetWitness. Customers can use this framework to develop their own collection protocols.
SDEE	Collects Intrusion Detection System (IDS) and Intrusion Prevention Service (IPS) messages. For details, see Configure SDEE Event Sources in NetWitness.
SNMP Trap	Accepts SNMP traps. For details, see Configure SNMP Event Sources in NetWitness.
Syslog	Accepts messages from event sources that issue syslog messages. For details, see Configure Syslog Event Sources. **Note:** You do not configure Syslog Collection for Local Log Collectors. You only need to configure Syslog Collection for Remote Collectors.
VMware	Collects events from a VMware virtual infrastructure. For details, see Configure VMware Event Sources in NetWitness.
Windows	Collects events from Windows machines that support the Microsoft Windows model. Windows 6.0 is an event logging and tracing framework included in the operating system beginning with Microsoft Windows Vista and Windows Server 2008. For details, see Configure Windows Event Sources in NetWitness.
Logstash	Utilize Logstash as a collection method, leveraging the various number of plugins supported by Logstash, such as: • Custom - This send the events from Logstash to NetWitness Platform. • Filebeat- This collects events from file. • Auditbeat - This collects audit events from an operating system (for example CentOS). • Export connector - This exports events from Decoder or Log Decoder to third party system. By default, Logstash version 7.10.0 is installed with all standard Logstash plugins (Beats, Export connector and so on) when Log collector is installed. For more information see, Configure Logstash Event Sources in NetWitness.

Collection Protocol	Description
Windows Legacy	Collects events from: • Older Windows versions such as Windows 2000 and Window 2003 and collects from Windows event sources that are already configured for enVision collection without having to reconfigure them. • NetApp ONTAP appliance event source so that you can now collect and parse NetApp evt files. • For more information, see.Windows Legacy and NetApp Collection Configuration. **Note:** You install the NetWitness Windows Legacy Collector on a physical or virtual Windows 2008 R2 SP1 64-Bit server using the **SALegacyWindowsCollector-*version-number.exe***.

Log Collection Basic Procedure for all Protocols

The basic procedure is the same for all of the supported Collection Protocols.

To configure collection for an event source:

1. **Set up your Event Source for collection.** Each supported event source has a configuration document available in the NetWitness Supported Event Sources space on NetWitness Link
   a. Navigate to the NetWitness Supported Event Sources space on NetWitness Link.
   b. Find the Instructions for your Event Source.
      The Overview page lists all of the currently supported Event Sources, as well as information about the collection method, device class, and supported versions.
   c. Download the configuration instructions for your event source, and follow them.
2. **Configure collection onNetWitness.** The event source configuration guide contains these instructions. However, this guide also provides these instructions, based on the collection method used by your event source. See Collection Protocols for details.
3. **Start the Service for your Collection Method.** Normally, you only need to do this for the first event source that uses this collection method. For example, the first time you configure an event source that uses File Collection, you may need to start the File Service in NetWitness.
4. **Verify that Collection is working for your Event Source.**

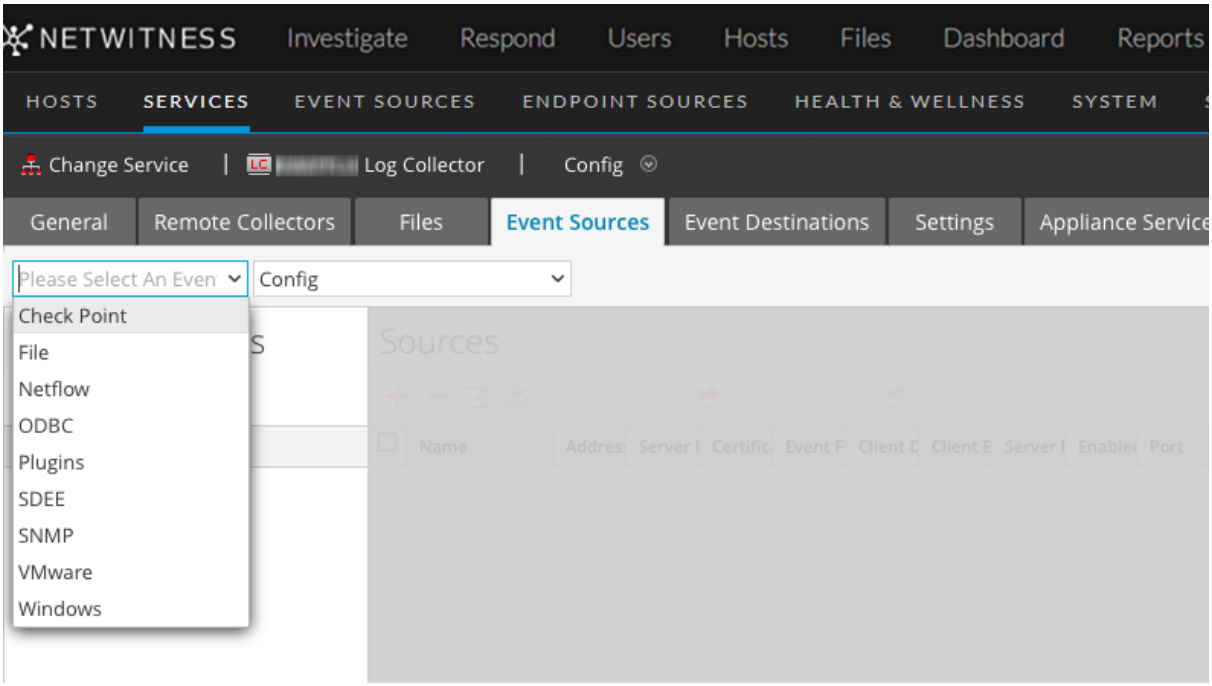
The remainder of this topic discusses steps 2, 3, and 4 in more detail.

Configure Collection in NetWitness

The process to configure event sources is dependent upon the collection method they use. Note, however, that they are very similar. The following procedure is generic: more details for individual collection methods are available in topics that cover the details for each specific collection method.

Basic procedure to configure an event source in NetWitness:

1. Go to **(Admin)** > **Services** from the NetWitness menu.
2. Select a Log Collection service.
3. Under Actions, select > **View** > **Config** to display the Log Collection configuration parameter tabs.
4. Click the **Event Sources** tab.



5. In the **Log Collector Event Sources** tab, select your collection method from the drop-down menu.
6. In the **Event Categories** panel toolbar, click .
   The Available Event Source Types dialog box is displayed.
7. Select an event source type and click **OK**.
   The newly added event source type is displayed in the Event Categories panel.
8. Select the new type in the **Event Categories** panel and click in the Sources toolbar.
   The **Add Source** dialog is displayed.
9. Enter values for the available parameters.

Refer to the Parameters section of the specific collection method that you are configuring.

10. Click **OK**.

Start the Service for your Collection Method

To start the service for your collection method:

1. Go to **(Admin) > Services**.
2. Select a **Log Collector** and select **> View > System**.
3. Click **Collection > *protocol* > Start**

   where ***protocol*** is the protocol that you wish to start, for example **Netflow**.

Verify that Collection is working for your Event Source

You can verify that a collection method is working from the **(Admin) > Health & Wellness > Event Source Monitoring** tab.

To verify that collection is working for an event source:

1. Go to **(Admin) > Health & Wellness**
2. Click the **Event Source Monitoring** tab.
3. In the grid, find the **Log Decoder**, **Event Source**, and **Event Source Type**.
4. Look for activity in the **Count** column for an event source to verify that collection is accepting events.

Search for Specific Event Sources

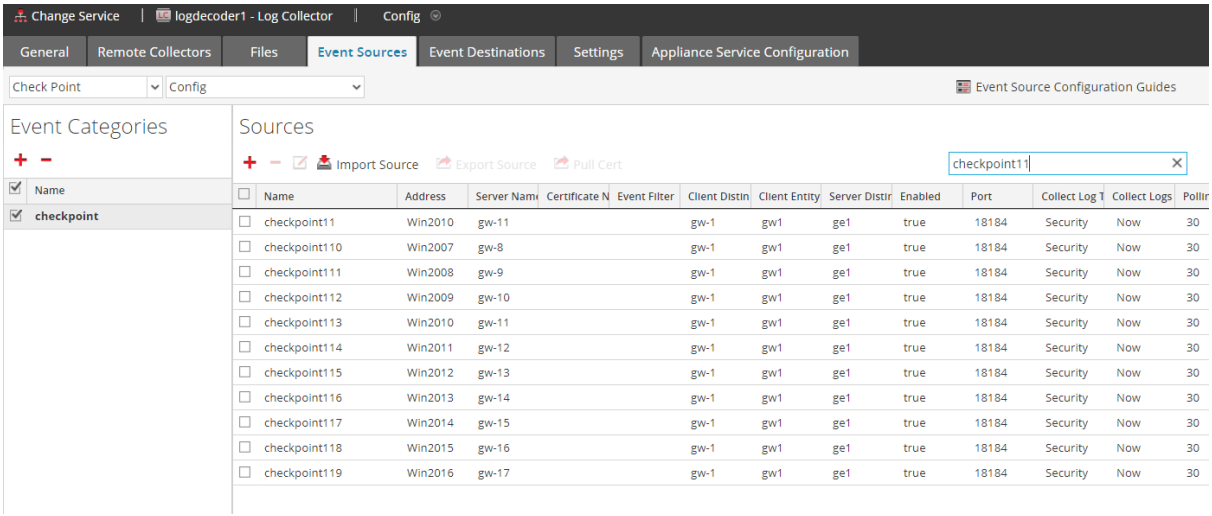
In some cases, your Log Collector may contain a lot of pre-configured event sources for a specific collection protocol (for example File). If so, you can quickly search for one or more event sources based on address (IP/hostname) or name.

To search for one or more specific event sources:

1. Go to **(Admin) > Services** from the NetWitness menu.
2. Select a Log Collection service.
3. Under Actions, select **> View > Config** to display the Log Collection configuration parameter tabs.
4. Click the **Event Sources** tab.
5. In the **Event Sources** tab, select any collection protocol/**Config** from the drop-down menu.

6. From the **Filter by Name / Address** text field, enter an IP address or hostname and click **Enter**.

   Event sources that match the information entered into the search box are returned. For example, the image below shows a list of Check Point event sources whose names match the string **checkpoint11**.



Configure Event Filters for a Collector

This topics tells you how to create and maintain Event filters across all collection protocols.

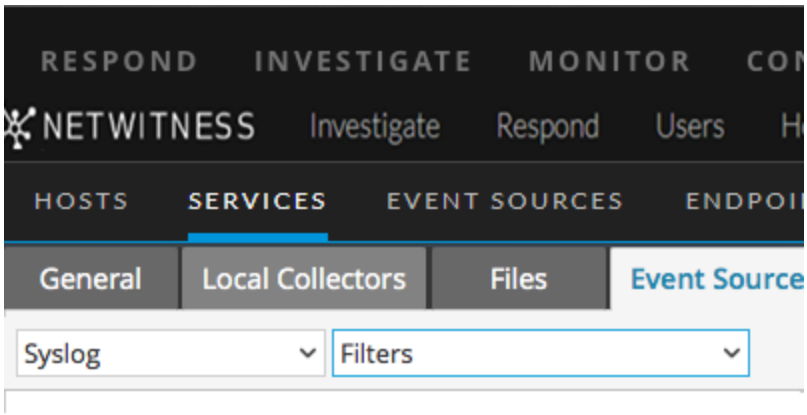
Note: Prior to 11.3, you could not configure Syslog Collection for Local Log Collectors. You ***can*** configure Syslog for local Log Collectors that are on version 11.3 or later.

Configure an Event Filter

To configure an event filter for an event source:

1. Go to **(Admin) > Services**.
2. Select a Log Collection service.
3. Under Actions, select **View > Config** to display the Log Collection configuration parameter tabs.
4. Click the **Event Sources** tab.
5. In the **Event Sources** tab, select any collection method / **Filter** from the drop-down menus.

   The following screen shows **Syslog** selected.

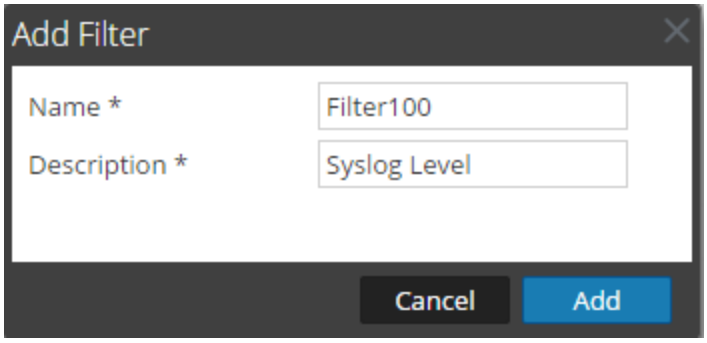


Note: Syslog configuration is only available on Remote Collectors prior to 11.3: if you are working with a Local Collector service, **Syslog** is not available from the drop-down menu for Log Collectors on version 11.0, 11.1, or 11.2.

The **Filters** view displays the filters that are configured for the selected collection method, if any.

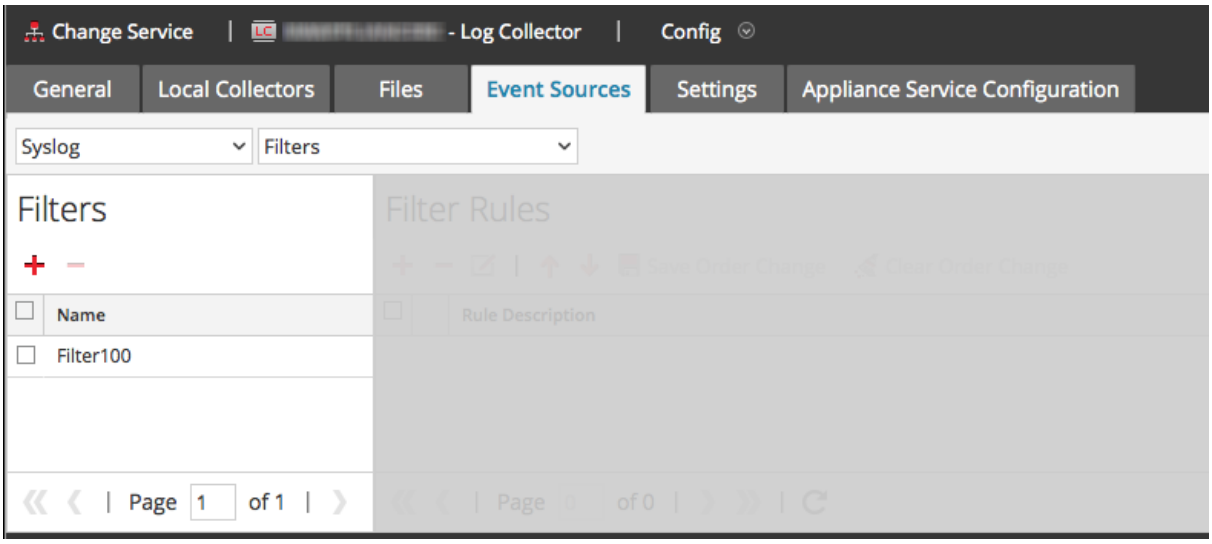
6. In the **Filters** panel toolbar, click  .

   The **Add Filter** dialog displays.

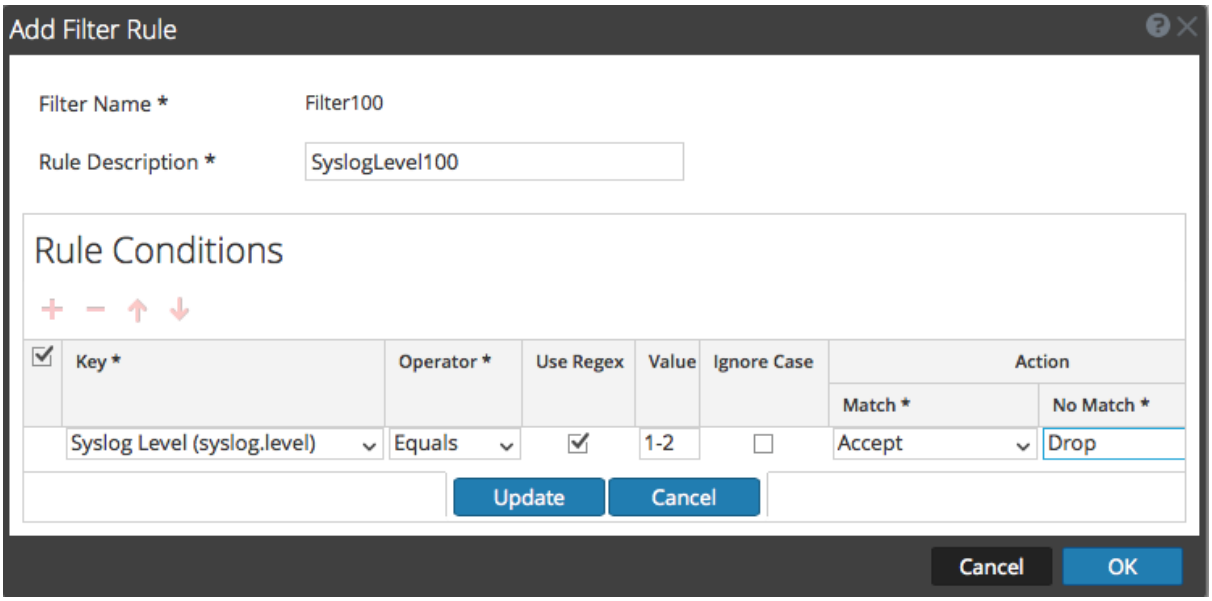


7. Enter a name and description for the new filter and click **Add**.

   The new filter displays in the **Filter** panel.



8. Select the new filter in the **Filters** panel and click + in the **Filter Rules** panel toolbar.
   The **Add Filter Rule** dialog is displayed.
9. Click + under **Rule Conditions**.
10. Add the parameters for this rule and click **Update > OK**.



NetWitness updates the filter with the rule that you defined.

Note: Rules are processed in order from top down until an Action type aborts the processing, or the final rule is checked. Default behavior is to accept the rule if no matches are found.

The following tables describe the parameters for adding a filter rule.

Event Filter Rule "Key" Parameter

The values for the Key field depend on the Collection method to which the filter applies.

Collection Method	Values for the *Key* Field
Checkpoint, File, Netflow, Plugin, SDEE SNMP and VMware	• All Data Fields • Event Source Type • Event Source Name • Source IP • Raw Event
ODBC	• All Data Fields • Event Source Type • Event Source Name • Source IP • Message ID • Message Level
Syslog	• All Data Fields • Event Source Type • Event Source Name • Source IP • Syslog level • Raw Event
Windows	• All Data Fields • Event Source Type • Event Source Name • Source IP • Event ID • Provider • Channel • Computer • UserName • DomainName

Collection Method	Values for the *Key* Field
Windows Legacy	• All Data Fields • Event Source Type • Event Source Name • Source IP • Event ID

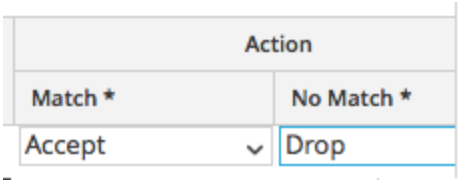
Other Event Filter Rule Parameters

The following table describes all the other available fields for creating an event filter rule.

Field	Description
Operator	Valid values are: • Contains • Equal
Use Regex	Optional. You can select this if you want to use Regex.
Value	Value depends on the key value you selected. For example if you choose **Syslog level** for Key, the value will be a number that denotes the syslog level.
Ignore case	Optional. Select this to ignore the case sensitivity.
Action	Choose actions for message data that matches a condition, or that does not match a condition. See below for more details.

Actions

You can choose a 'match' and 'no-match' action for each rule condition. This screenshot shows a condition that uses the **Accept** action on matches, and **Drop** action on events that do not match the condition.

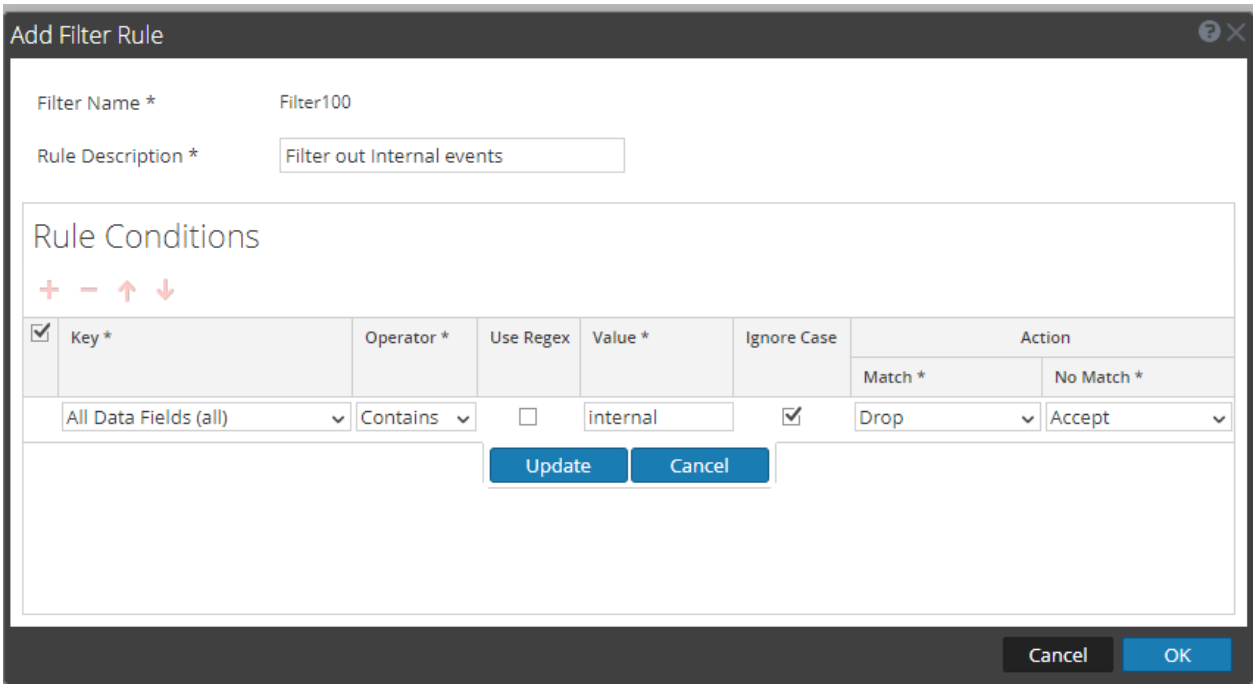


The available actions are as follows:

- **Accept**: the filtered event is included in event logs, and no further rule processing is done for the event: the event will display in the NetWitness user interface during Investigation.

- **Drop**: the filtered event will not be included in event logs, and no further rule processing is done for the event: the event will not display during Investigation.
- **Next condition**: the filtered event moves on to the next rule condition in the rule. If there are no more rule conditions in the current rule, it moves on to the next rule.
- **Next rule**: the filtered event moves on to the next rule. If there are no more rules in the filter, the filter event is included in event logs, and no further rule processing is done for the event (same as **Accept**).

For example, consider the following condition:



In this condition, if the event log contains the string "internal," the event is not included in the event meta, nor is it displayed during Investigation. If the string is not found, the event is included and is displayed during Investigation.

Modify Filter Rules

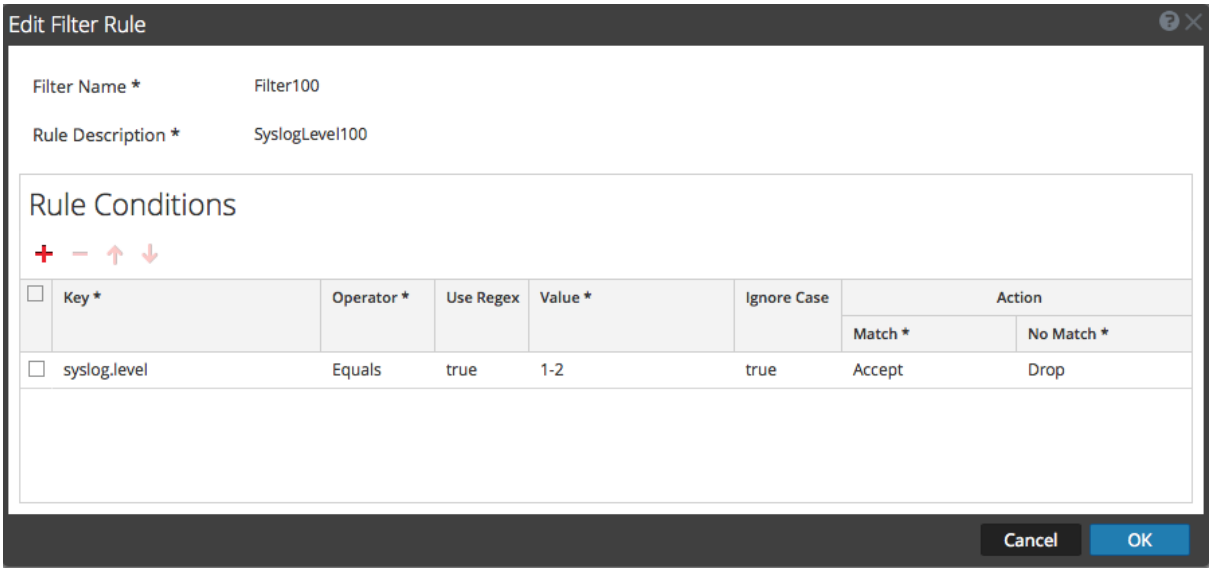
To modify existing filter rules:

1. Go to  **(Admin) > Services**.
2. Select a Log Collection service.
3. Under Actions, select **View > Config** to display the Log Collection configuration parameter tabs.
4. Click the **Event Sources** tab.
5. In the **Event Sources** tab, select any collection method / **Filter** from the drop-down menus.

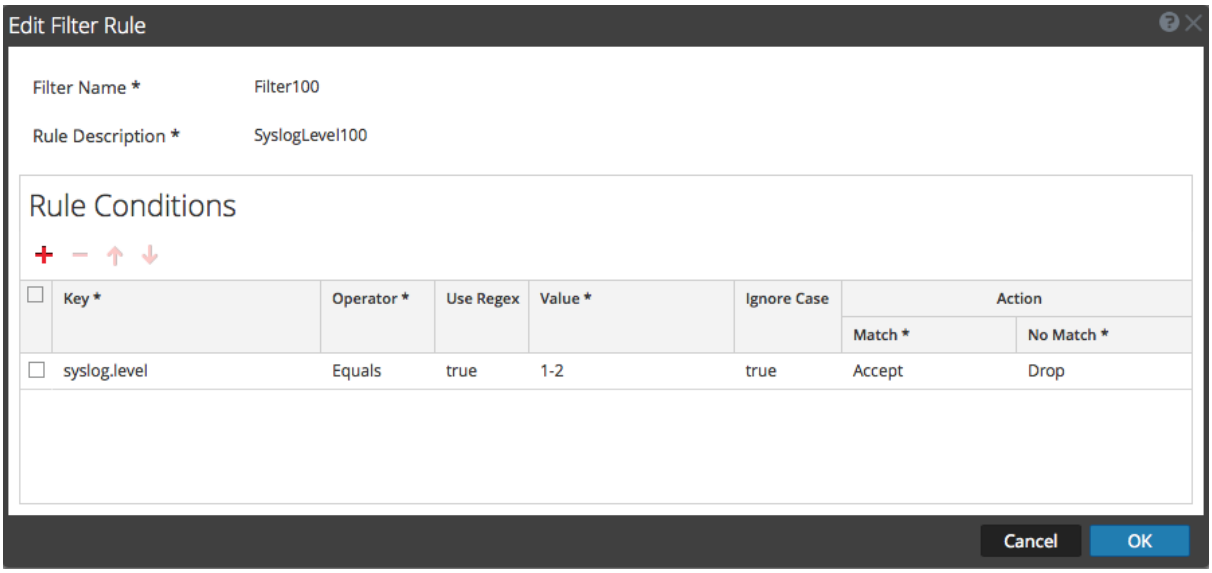
   The following screen shows **Check Point** selected.

   The **Filters** view displays the filters that are configured for the selected collection method, if any.

6. In the **Filter Rules** list, select a rule and click  .
   The **Edit Filter Rule** dialog is displayed.



7. Select the rule condition that you want to modify.



8. Modify the condition parameters that require changes and click **Update** > **OK**.

NetWitness applies the condition parameter changes to the selected filter rule.

Import, Export, Edit, and Test Event Sources in Bulk

This topic describes how to import, export, edit, and test event sources in bulk.

You can use the bulk export option to export the event source details of your current set up and store it. This data can be imported in bulk when you face a problem with your current set up and require the event source data you had.

You can use the bulk edit feature when you have multiple event sources that need a specific modification. You can select all the sources and apply the edit option across them at a time and avoid applying the change one by one.

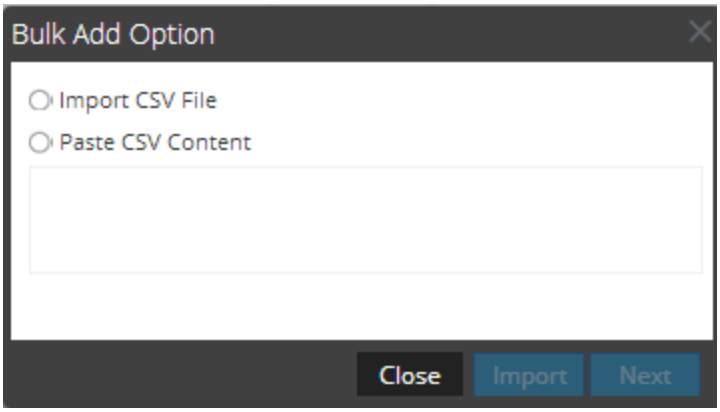
Import Event Sources in Bulk

Warning: When using a spreadsheet program to edit an exported event source CSV file, some data fields like numbers and dates can be re-formatted into the spreadsheet program's native field types. This can cause issues when re-importing this information, as some data fields may be garbled or formatted incorrectly. This can be avoided by importing the CSV file into the spreadsheet program, and specifying all data fields as text values.

To import multiple event sources at once:

1. Go to  **(Admin) > Services**.
2. Select a Log Collection service.
3. Under Actions, select  **> View > Config** to display the Log Collection configuration parameter tabs.
4. Click the **Event Sources** tab.
5. Select **Check Point**, **File**, **Netflow**, **Logstash**, **ODBC**, **Plugins**, **SDEE**, **(Syslog for Remote Collectors) only**, **VMware**, **Windows**, or **Windows Legacy** (SNMP does not have an Import function.).
6. In the **Sources** panel toolbar, click **Import Source**.

   The **Bulk Add Option** dialog is displayed.

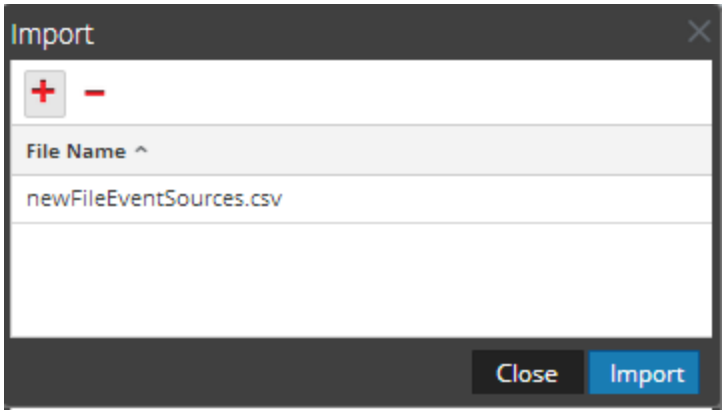


7. Select either **Import CSV File** or **Paste CSV Content**. If you select:

- Import CSV File:
  a. Click **Next**.

     The **Import dialog** is displayed.

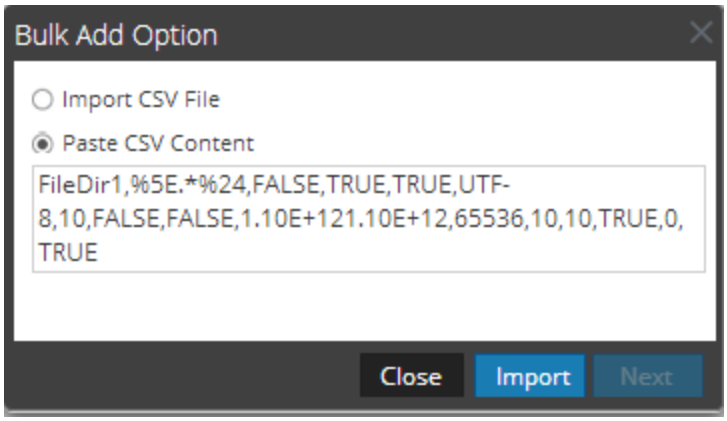
  b. Click **Add** and select a **.csv** file from your network.



  c. Click **Import**.

     The event sources are added to the **Event Source** list.

- Paste CSV Content
  a. Copy the contents of the **.csv** file and paste them into the dialog.



  b. Click **Import**.

     The event sources are added to **Event Source List**.

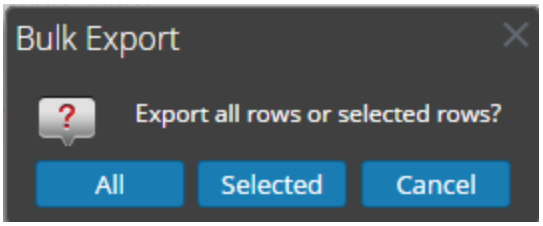
Export Event Sources in Bulk

Warning: When using a spreadsheet program to edit an exported event source CSV file, some data fields like numbers and dates can be re-formatted into the spreadsheet program's native field types. This can cause issues when re-importing this information, as some data fields may be garbled or formatted incorrectly. This can be avoided by importing the CSV file into the spreadsheet program, and specifying all data fields as text values.

To export event source information:

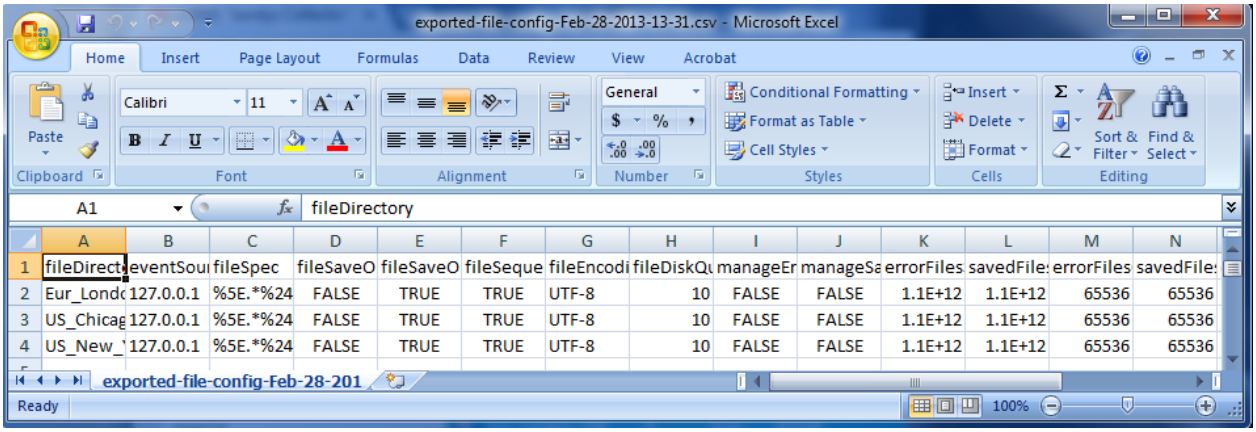
1. Go to  **(Admin) > Services**.
2. Select a Log Collection service.
3. Select  **> View > Config** to display the Log Collection configuration parameter tabs.
4. Click the **Event Sources** tab.
5. Select **Check Point**, **File**, **Netflow**, **Logstash**, **ODBC**, **Plugins**, **SDEE**, **(Syslog for Remote Collectors) only**, **VMware**, **Windows**, or **Windows Legacy** (SNMP does not have an Export function.).
6. In the **Sources** panel, select one or multiple event sources and click **Export Source**.

   The **Bulk Export** dialog is displayed.



7. Based on your selection:
   - **All**: NetWitness exports all event sources to a time-stamped CSV file.
   - **Selected**: NetWitness exports the event source or sources you selected to a time-stamped CSV file.
   - **Cancel**: NetWitness cancels the export.

   The following is an example of a time-stamped CSV file that gets created with the event sources that you selected from the list.



Edit Event Sources in Bulk

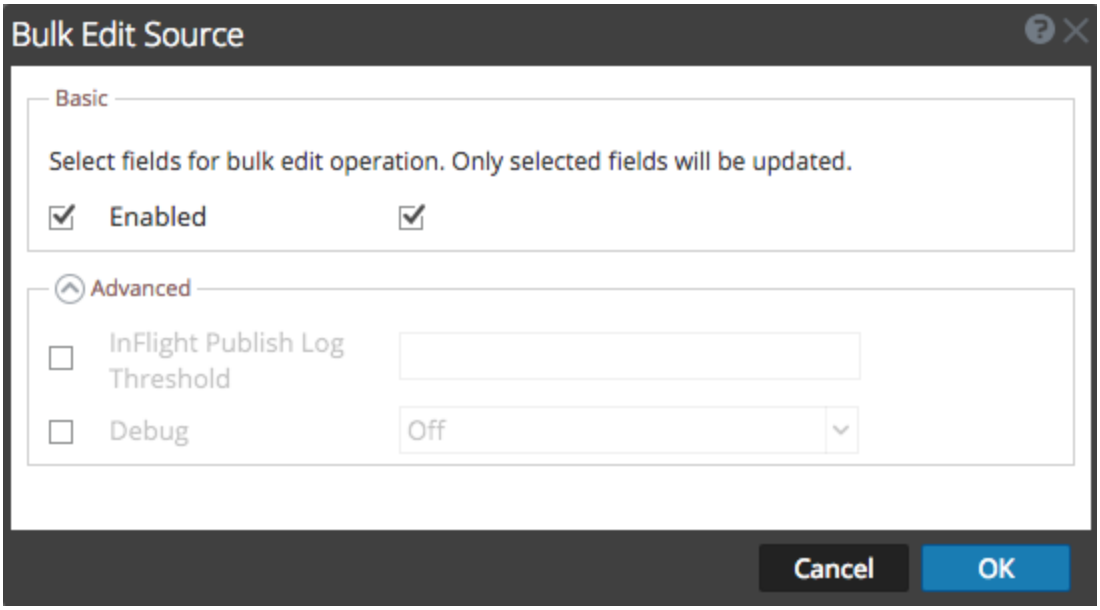
To edit multiple event sources at once:

1. On the **Log Collector Event Sources** tab, select **Check Point**, **File**, **Netflow**, **Logstash**, **ODBC**, **Plugins**, **SDEE**, **Syslog**, **VMware**, **Windows**, or **Windows Legacy** (SNMP does not have

an Edit function.).

2. In the **Sources** panel, select multiple event sources and click .

   The appropriate **Bulk Edit** dialog for the selected event source is displayed. The following figure is an example of **Bulk Edit Source** dialog for File event source parameters.



3. Select the checkbox to the left of the fields that you want to modify (for example, **Debug**).
4. Modify the selected parameters (for example, change Debug from **Off** to **On**).
5. Click **OK**.

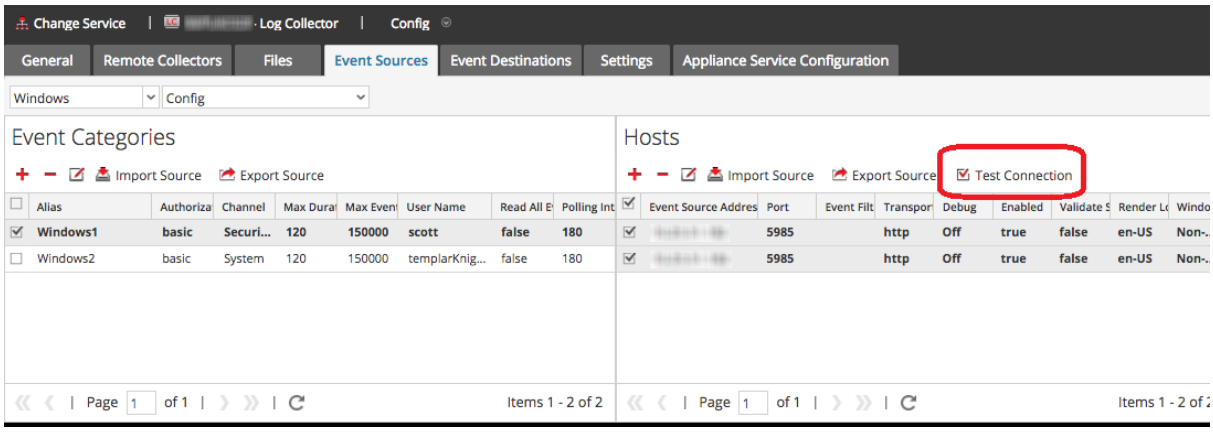
   NetWitness applies the same parameter value change to all of the selected event sources

Test Event Source Connections in Bulk

To test multiple event source connections at once:

1. Go to **(Admin) > Services**.
2. In the **Services** grid, select a **Log Collector** service.
3. Select **> View > Config** to display the Log Collection configuration parameter tabs.
4. Select the **Event Sources** tab, select **Plugins**, **ODBC**, **Logstash** or **Windows** (the other protocols do not have a bulk test connection function).
5. Select one or more:
   - sources from the **Sources** panel for **Plugins** or **ODBC**
   - hosts from **Hosts** panel for **Windows**

   The **Test Connection** button is enabled.



6. Click Test Connection .

   The **Bulk Test Connections** dialog is displayed showing the current status of the test for each source. The status can be waiting, testing, passed or failed.

   If you choose to close the testing before it is completed, the testing stops and the **Bulk Test Connections** dialog closes.

After the testing is complete, the results are displayed in the **Bulk Test Connections** dialog.

See Also

You can use the **Event Sources** module ((Admin) > **Event Sources**) to create groups of event sources, typically imported from a CMDB, and to monitor event sources based on those groups. For details, see the following topics in the *Event Source Management Guide*:

- Import Event Sources
- Export Event Sources
- Bulk Edit Event Source Attributes

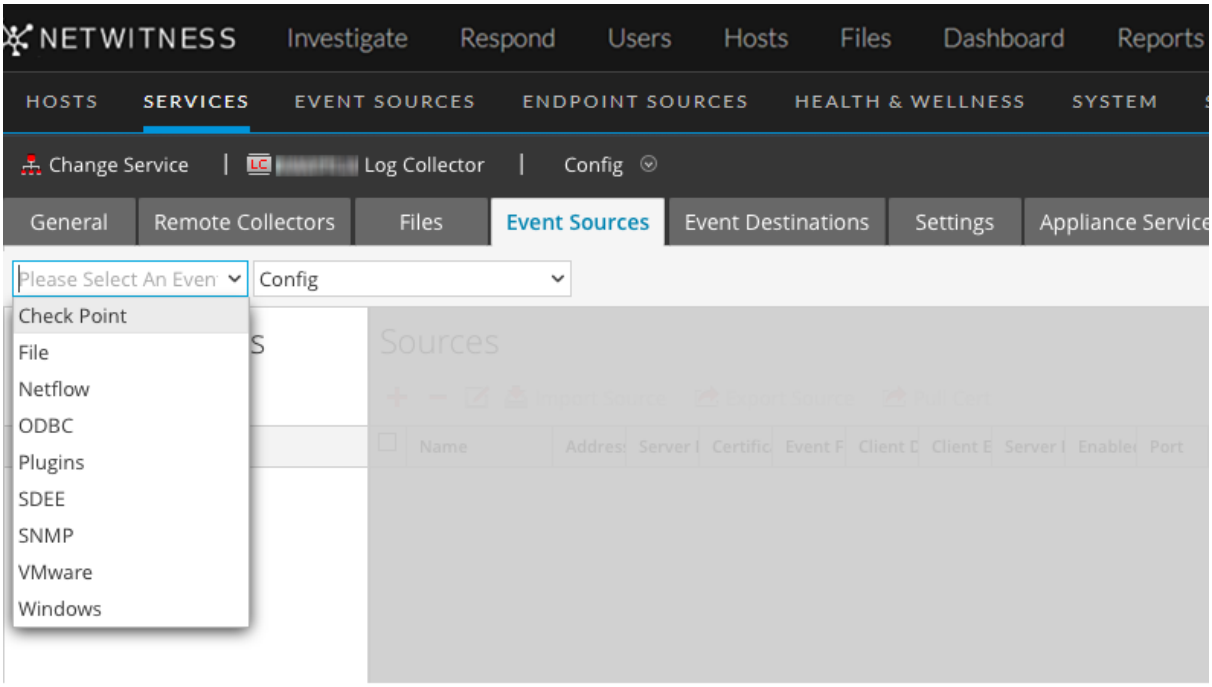
Configure Collection Protocols and Event Sources

This topic tells you how to configure collection protocols and the event sources using those protocols.

You configure the Log Collector to collect event data from your event sources in the Event Sources tab of the Log Collection parameter view.

To configure a collection protocol:

1. Go to **(Admin) > Services** from the NetWitness menu.
2. Select a Log Collection service.
3. Under Actions, select **> View > Config** to display the Log Collection configuration parameter tabs.
4. Click the **Event Sources** tab.



5. Select a collection protocol (for example, **File**) and select **Config**.
6. Click + and select an event source.
7. Select the newly added category and click +.
8. Specify the parameters for the event source. For details, see the individual collection protocol topics.

The following guides provide detailed instructions on how to configure the collection protocols and their associated event sources in NetWitness. Each guide includes an index to configuration instructions for the event sources supported for that collection protocol.

To configure individual collection protocols, see the following topics:

- Configure AWS (CloudTrail) Event Sources in NetWitness
- Configure Azure Event Sources in NetWitness
- Configure Check Point Event Sources in NetWitness
- Configure File Event Sources in NetWitness
- Configure Netflow Event Sources in NetWitness
- Configure ODBC Event Sources in NetWitness
  - Configure Data Source Names (DSNs)
  - Create Custom Typespec for ODBC Collection
  - ODBC Event Source Configuration Parameters
  - ODBC DSNs Event Source Configuration Parameters
- Configure SDEE Event Sources in NetWitness
- Configure SNMP Event Sources in NetWitness
- Configure Syslog Event Sources
- Configure VMware Event Sources in NetWitness
- Configure Windows Event Sources in NetWitness
- Windows Legacy and NetApp Collection Configuration
  - Set Up the Windows Legacy Collector
  - Configure Windows Legacy and NetApp Event Sources
  - Troubleshoot Windows Legacy and NetApp Collection
- Configure Logstash Event Sources in NetWitness

Configure AWS (CloudTrail) Event Sources in NetWitness

This topic tells you how to configure the AWS collection protocol, which collects events from Amazon Web Services (AWS) CloudTrail.

Note: The AWS plugin is meant only for collecting from AWS CloudTrail logs, and not for collecting from arbitrary logs in S3 buckets (under arbitrary directories). The AWS CloudTrail logs are sent in JSON format, as detailed in the AWS documentation here: http://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-event-reference.html.

How AWS Collection Works

The Log Collector service collects events from Amazon Web Services (AWS) CloudTrail. CloudTrail records AWS API calls for an account. The events contain the identity of the API caller, the time of the API call, the source IP address of the API caller, the request parameters, and the response elements returned by the AWS service. The AWS API call history provided by CloudTrail events enables security analysis, resource change tracking, and compliance auditing. CloudTrail uses Amazon S3 for log file storage and delivery. NetWitness copies the log files from the cloud (S3 bucket), and sends the events contained in the files to the Log Collector.

Deployment Scenario

The following figure illustrates how you deploy the AWS Collection Protocol in NetWitness.

Configuration

To configure an AWS (CloudTrail) Event Source:

1. Go to **(Admin) > Services** from the NetWitness menu.
2. Select a Log Collection service.
3. Select **> View > Config** to display the Log Collection configuration parameter tabs.
4. Click the **Event Sources** tab.
5. In the **Event Sources** tab, select **Plugins/Config** from the drop-down menu.
6. In the **Event Categories** panel toolbar, click .
   The **Available Event Source Types** dialog is displayed.

7. Select **cloudtrail**) and click **OK**.

   The newly added event source type is displayed in the **Event Categories** panel.

8. Select the new type in the **Event Categories** panel and click  in the **Sources** toolbar.

   The **Add Source** dialog is displayed.

9. Define parameter values. For details, see AWS Parameters below.
10. Click **Test Connection**.

    The result of the test is displayed in the dialog box. If the test is unsuccessful, edit the device or service information and retry.

    Log Collector takes approximately 60 seconds to return the test results. If it exceeds the time limit, the test times out and the NetWitness displays an error message.

11. If the test is successful, click **OK**.

    The new event source is displayed in the **Sources** panel.

AWS Parameters

The following table describes the **Basic** configuration parameter for AWS collection.

Note: Required parameters are marked with an asterisk. All other parameters are optional.

Parameter	Description
Name *	Name of the event source.
Enabled 	Select the check box to enable the event source configuration to start collection. The check box is selected by default.
Account Id *	Account Identification code of the S3 Bucket

Parameter	Description
S3 Bucket Name *	Name of the AWS (CloudTrail) S3 bucket. Amazon S3 bucket names are globally unique, regardless of the AWS (CloudTrail) region in which you create the bucket. You specify the name at the time you create the bucket. Bucket names should comply with DNS naming conventions. The rules for DNS-compliant bucket names are: • Bucket names must be at least three and no more than 63 characters long. • Bucket names must be a series of one or more labels. Adjacent labels are separated by a single period ".". Bucket names can contain lowercase letters, numbers, and hyphens. Each label must start and end with a lowercase letter or a number. • Bucket names must not be formatted as an IP address (for example, 192.168.5.4). The following examples are **valid** bucket names: • `myawsbucket` • `my.aws.bucket` • `myawsbucket.1` The following examples are **invalid** bucket names: • `.myawsbucket` - Do not start a Bucket Name with a period • `myawsbucket.` - Do not end a Bucket Name with a period • `my..examplebucket` - Only use one period between labels.
Access Key *	Key used to access the S3 bucket. Access Keys are used to make secure REST or Query protocol requests to any AWS service API. Please refer to Manage User Credentials on the Amazon Web Services support site for more information on Access Keys.
Secret Key *	Secret key used to access the S3 bucket.
Region *	Region of the S3 bucket. `us-east-1` is the default value.
Region Endpoint	Specifies the AWS CloudTrail hostname. For example, for an AWS public cloud for us-east region, the Region Endpoint would be s3.amazonaws.com. More information can be found at http://docs.aws.amazon.com/general/latest/gr/rande.html#s3_region. This parameter is necessary to collect CloudTrail logs from AWS Government or Private clouds.
Use Proxy	Enable **Use Proxy** to set proxy for AWS server. By default, it is disabled.
Proxy Server	Enter the proxy name you want to connect to access the AWS server.
Proxy Port	Enter the port number that connects to the proxy server to access the AWS server.
Proxy User	Enter the user name to authenticate with the proxy server.
Proxy Password	Enter the password to authenticate with proxy port.
Start Date *	Starts AWS (CloudTrail) collection from the specified number of days in the past, measured from the current timestamp. The default value is 0, which starts from today. The range is 0–89 days.
Log File Prefix	Prefix of the files to be processed. **Note:** If you set a prefix when you set up your CloudTrail service, make sure to enter the same prefix in this parameter.

Parameter	Description
Cancel	Closes the dialog without adding the AWS (CloudTrail).
OK	Adds the current parameter values as a new AWS (CloudTrail).

The following table describes the **Advanced** configuration parameter for AWS collection.

Parameter	Description
Debug	**Caution:** Only enable debugging (set this parameter to On or Verbose) if you have a problem with an event source and you need to investigate this problem. Enabling debugging will adversely affect the performance of the Log Collector. Enables or disables debug logging for the event source. Valid values are: • **Off** = (default) disabled **On** = enabled • **Verbose** = enabled in verbose mode - adds thread information and source context information to the messages. This parameter is designed to debug and monitor isolated event source collection issues. The debug logging is verbose, so limit the number of event sources to minimize performance impact. If you change this value, the change takes effect immediately (no restart required).
Command Args	Arguments added to the script.
Polling Interval	Interval (amount of time in seconds) between each poll. The default value is 60. For example, if you specify 60, the collector schedules a polling of the event source every 60 seconds. If the previous polling cycle is still underway, it will wait for it to finish that cycle. If you have a large number of event sources that you are polling, it may take longer than 60 seconds for the polling to start because the threads are busy.
SSL Enabled 	Select the check box to communicate using SSL. The security of data transmission is managed by encrypting information and providing authentication with SSL certificates. The check box is selected by default.
Test Connection	Validates the configuration parameters specified in this dialog are correct. For example, this test validates that: • NetWitness can connect with the S3 Bucket in AWS using the credentials specified in this dialog. • NetWitness can download a log file from the bucket (test connection would fail if there were no log files for the entire bucket, but this would be extremely unlikely).

Configure Azure Event Sources in NetWitness

This topic tells you how to configure the Azure collection protocol. Microsoft Azure is a cloud computing platform and infrastructure for building, deploying, and managing applications and services through a global network of Microsoft-managed data centers.

Configuration in NetWitness

For complete details about configuring Azure as an event source, see the Azure Event Source Configuration Guide, available on NetWitness Link.

To configure an Azure Event Source:

1. Go to  **(Admin) > Services** from the NetWitness menu.
2. Select a Log Collection service.
3. Select  **> View > Config** to display the Log Collection configuration parameter tabs.
4. Click the **Event Sources** tab.
5. In the **Event Sources** tab, select **Plugins/Config** from the drop-down menu.
6. In the **Event Categories** panel toolbar, click  .

   The **Available Event Source Types** dialog is displayed.
7. Select **azureaudit**) and click **OK**.

   The newly added event source type is displayed in the **Event Categories** panel.
8. Select the new type in the **Event Categories** panel and click  in the **Sources** toolbar.

   The **Add Source** dialog is displayed.
9. Define parameter values. For details, see Azure Parameters below.
10. Click **Test Connection**.

    The result of the test is displayed in the dialog box. If the test is unsuccessful, edit the device or service information and retry.

    Log Collector takes approximately 60 seconds to return the test results. If it exceeds the time limit, the test times out and the NetWitness displays an error message.
11. If the test is successful, click **OK**.

    The new event source is displayed in the **Sources** panel.

Azure Parameters

This section describes the Azure event source configuration parameters.

Basic Parameters

Note: Required parameters are marked with an asterisk. All other parameters are optional.

Name	Description
Name *	Enter an alpha-numeric, descriptive name for the source. This value is only used for displaying the name on this screen.
Enabled	Select the checkbox to enable the event source configuration to start collection. The checkbox is selected by default.
Client ID *	The Client ID is found the Azure Application Configure tab. Scroll down until you see it.
Client Secret *	When you are configuring the event source, the client secret is displayed when you are creating a key, and you select a duration of validation. Make sure to save this, because you will only be able to see it once, and it cannot be retrieved later.
API Resource Base URL *	Enter `https://management.azure.com/`. Be sure to include the trailing slash (/).
Federation Metadata Endpoint *	In your Azure application, click the **View Endpoints** button (near the bottom of the pane). There are a lot of links that all begin with the same string. Compare the URLs and find the common string that begins most of them. This common string is the endpoint that you need to enter here.
Subscription ID *	You can find this in the Microsoft Azure dashboard: click on Subscriptions at the bottom of the list on the left.
Tenant Domain *	Go to the active directory and click on the directory. In the URL, the tenant domain is the string directly following **manage.windowsazure.com/**. The tenant domain is the string up to and including the **.com**.
Resource Group Names *	In Azure, select Resource groups from the left navigation pane, then select your group.
Start Date *	Choose the date from which to start collecting. Default's to the current date.
Test Connection	Checks the configuration parameters specified in this dialog to make sure they are correct.

Advanced Parameters

Click  next to **Advanced** to view and edit the advanced parameters, if necessary.

Name	Description
Polling Interval	Interval (amount of time in seconds) between each poll. The default value is **180**. For example, if you specify 180, the collector schedules a polling of the event source every 180 seconds. If the previous polling cycle is still underway, the collector waits for that cycle to finish. If you have a large number of event sources that you are polling, it may take longer than 180 seconds for the polling to start because the threads are busy.

Name	Description
Max Duration Poll	Maximum duration, in seconds, of a polling cycle. A zero value indicates no limit.
Max Events Poll	The maximum number of events per polling cycle (how many events collected per polling cycle).
Max Idle Time Poll	Maximum duration, in seconds, of a polling cycle. A zero value indicates no limit.
Command Args	Optional arguments to be added to the script invocation.
Debug	**Caution:** Only enable debugging (set this parameter to On or Verbose) if you have a problem with an event source and you need to investigate this problem. Enabling debugging will adversely affect the performance of the Log Collector. Enables or disables debug logging for the event source. Valid values are: • **Off** = (default) disabled • **On** = enabled • **Verbose** = enabled in verbose mode - adds thread information and source context information to the messages. This parameter is designed to debug and monitor isolated event source collection issues. If you change this value, the change takes effect immediately (no restart required). The debug logging is verbose, so limit the number of event sources to minimize performance impact.

Configure Check Point Event Sources in NetWitness

This topic tells you how to configure the Check Point collection protocol, which collects events from Check Point event sources.

This protocol collects events from Check Point event sources using OPSEC LEA. OPSEC LEA is the Check Point Operations Security Log Export API that facilitates the extraction of logs.

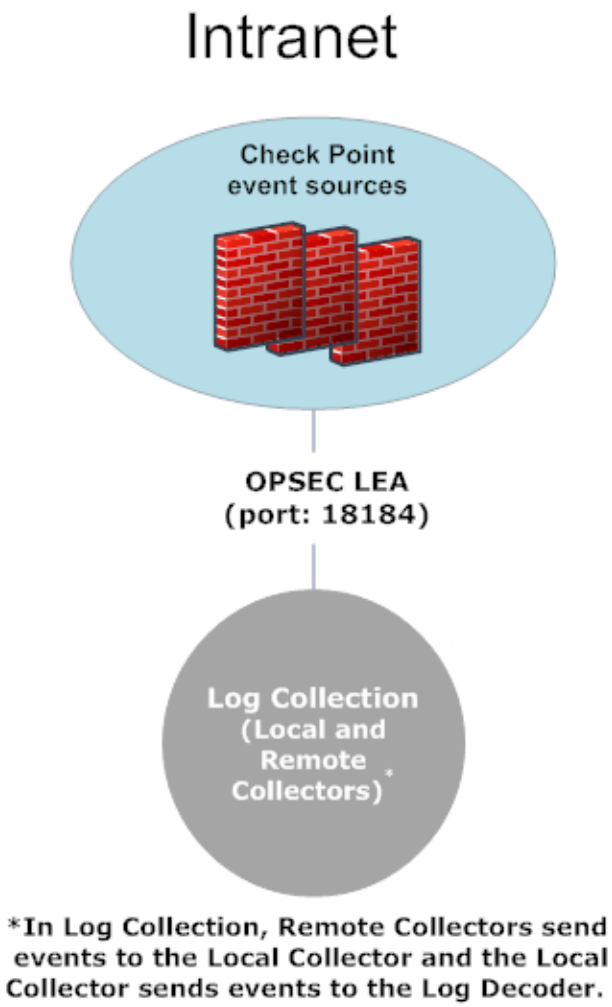
How Check Point Collection Works

The Log Collector service collects events from Check Point event sources using OPSEC LEA. OPSEC LEA is the Check Point Operations Security Log Export API that facilitates the extraction of logs.

Note: OPSEC LEA (Log Export API) supports extraction of logs from Check Point event sources configured with a SHA-256 or SHA-1 certificate.

Deployment Scenario

The following figure illustrates how you deploy the Check Point Collection Protocol in NetWitness.



Configuration in NetWitness

To configure a Check Point Event Source:

1. Go to **(Admin) > Services** from the NetWitness menu.
2. Select a Log Collection service.
3. Select **> View > Config** to display the Log Collection configuration parameter tabs.
4. Click the **Event Sources** tab.
5. In the **Event Sources** tab, select **Check Point/Config** from the drop-down menu.
6. In the **Event Categories** panel toolbar, click .
   The **Available Event Source Types** dialog is displayed.

7. Select a check point event source type and click **OK**.

   The newly added event source type is displayed in the **Event Categories** panel.

8. Select the new type in the **Event Categories** panel and click  in the **Sources** toolbar.

   The **Add Source** dialog is displayed.

9. Define parameter values. For details, see Check Point Parameters below.
10. Click **Test Connection**.

    The result of the test is displayed in the dialog box. If the test is unsuccessful, edit the device or service information and retry.

    Log Collector takes approximately 60 seconds to return the test results. If it exceeds the time limit, the test times out and the NetWitness displays an error message.

11. If the test is successful, click **OK**.

    The new event source is displayed in the **Sources** panel.

Check Point Parameters

This section describes the Check Point event source configuration parameters.

Basic Parameters

Parameter	Description
Name*	Name of the event source.
Address*	IP Address of the Check Point server.
Server Name*	Name of the Check Point server.
Certificate Name	Certificate name for secure connections to use when the transport mode is https. If set, the certificate must exist in the certificate trust store that you created using the Settings tab. Select a certificate from the drop-down list. The file naming convention for Check Point event source certificates is **checkpoint_*name-of-event-source***.
Client Distinguished	Enter the Client Distinguished Name from the Check Point server.
Client Entity Name	Enter the Client Entity Name from the Check Point server.
Server Distinguished	Enter the Server Distinguished Name from the Check Point server.
Enabled	Select the check box to enable the event source configuration to start collection. The check box is selected by default.

Parameter	Description
Pull Certificate	Select the checkbox to pull a certificate for first time. Pulling a certificate makes it available from the trust store.
Certificate Server Address	IP Address of the server on which the certificate resides. Defaults to the event source address.
Password	Only active when you select the Pull Certificate checkbox for first time. Password required to pull the certificate. The password is the activation key created when adding an OPSEC application to Check Point on the Check Point server.

Determine Advanced Parameter Values for Check Point Collection

You use less system resources when you configure a Check Point event source connection to stay open for a specific time and specific event volume (transient connection). NetWitness defaults to the following connection parameters that establish a transient connection:

- Polling Interval = **180** (3 minutes)
- Max Duration Poll = **120** (2 minutes)
- Max Events Poll = **5000** (5000 events per polling interval)
- Max Idle Time Poll = **0**

For very active Check Point event sources, it is a good practice to set up a connection that stays open until you stop collection (persistent connection). This ensures that Check Point collection maintains the pace of the events generated by these active event sources. The persistent connection avoids restart and connection delays and prevents Check Point collection from lagging behind event generation.

To establish a persistent connection for a Check Point event source, set the following parameters to the following values:

- Polling Interval = **-1**
- Max Duration Poll = **0**
- Max Events Poll = **0**
- Max Idle Time Poll = **0**

Parameter	Description
Port	Port on the Check Point server that Log Collector connects to. Default value is 18184.

Parameter	Description
Collect Log Type	Type of logs that you want to collect: Valid values are: • **Audit** - collects audit events. • **Security** - collects security events. If you want to collect both audit and security events, you must create a duplicate event source. For example, first you would create an event source with Audit selected pulling a certificate into the trust store for this event source. Next you would create another event source with the same values except that you would select Security for the Collect Log Type and you would select the same certificate in Certificate Name that you pulled when you set up the first set of parameters for this event source and you would make sure that Pull Certificate was not selected.
Collect Logs From	When you set up a Check Point event source, NetWitness collects events from the current log file. Valid values are: • **Now** - Start collecting logs now (at this point in time in the current log file). • **Start of Log** - Collect logs from the beginning of the current log file. If you choose "Start of Log" for this parameter value, you may collect a very large amount of data depending on how long the current log file has been collecting events. Note that this option is effective only for the first collection session.
Polling Interval	Interval (amount of time in seconds) between each poll. The default value is **180**. For example, if you specify 180, the collector schedules a polling of the event source every 180 seconds. If the previous polling cycle is still underway, it will wait for it to finish that cycle. If you have a large number of event sources that you are polling, it may take longer than 180 seconds for the polling to start because the threads are busy.
Max Duration Poll	The maximum duration of polling cycle (how long the cycle lasts) in seconds.
Max Events Poll	The maximum number of events per polling cycle (how many events collected per polling cycle).
Max Idle Time Poll	Maximum idle time, in seconds, of a polling cycle. 0 indicates no limit.**> 300** is the default value.
Forwarder	Enables or disables the Check Point server as a forwarder. By default it is disabled.
Log Type (Name Value Pair)	Logs from the event source in Name Value format. By default it is disabled.

Parameter	Description
Debug	**Caution:** Only enable debugging (set this parameter to "On" or "Verbose") if you have a problem with an event source and you need to investigate this problem. Enabling debugging will adversely affect the performance of the Log Collector. Enables and disables debug logging for the event source. Valid values are: • **Off** = (default) disabled • **On** = enabled • **Verbose** = enabled in verbose mode - adds thread information and source context information to the messages. This parameter is designed to debug and monitor isolated event source collection issues. The debug logging is verbose, so limit the number of event sources to minimize performance impact. If you change this value, the change takes effect immediately (no restart required).

Verify Check Point Collection is Working

The following procedure illustrates how you can verify that Check Point collection is working from the **Administration > Health & Wellness > Event Source Monitoring** tab.

To verify Check Point collection from the Event Source Monitoring tab:

1. Access the **Manage** tab from the  **(Admin) > Event Sources** view.
2. Find a checkpoint event source in the **Event Sources** column.
3. Look for activity in the **Total Count** column to verify that Check Point collection is accepting events.

To verify Check Point collection from the Investigation > Events view:

The following procedure illustrates how you can verify that Check Point collection is working from the **Investigation > Events** view.

1. Access the **Investigation > Events** view.
2. Select the Log Decoder (for example, **LD1**) collecting Check Point events in the **Investigate a Device** dialog.
3. Look for a Check Point event source parser (for example, **checkpointfw1**) in the **device.type** field in the **Details** column to verify that Check Point collection is accepting events.

Note: If the logs from the VSX Checkpoint firewall server are collected by the Log Collector checkpoint service, to translate the VSX IP in the logs to **ip.orig** meta, you must add the VSX hostname and the VSX IP address to the `/etc/hosts` file in the Log Collector.

Configure File Event Sources in NetWitness

This topic tells you how to configure the File collection protocol:

Note: In NetWitness 11.4 and later, you can perform File Log collection for many event sources using Endpoint Agents, thus simplifying the collection process. For details, see the *NetWitness Endpoint Configuration Guide*. For a list of which event sources are supported, see the section "Currently Supported File Log Event Source Types."

Configure a File Event Source

To configure a File Event Source:

1. Go to  **(Admin) > Services** from the NetWitness menu.
2. Select a Log Collection service.
3. Under Actions, select  **> View > Config** to display the Log Collection configuration parameter tabs.
4. Click the **Event Sources** tab.
5. In the **Event Sources** tab, select **File/Config** from the drop-down menu.
6. In the **Event Categories** panel toolbar, click  .
   The **Available Event Source Types** dialog is displayed.
7. Select a file event source type and click **OK**.
   The newly added event source type is displayed in the **Event Categories** panel.
8. Select the new type in the **Event Categories** panel and click  in the **Sources** toolbar.
   The **Add Source** dialog is displayed.
9. Add a **File Directory** name and modify any other parameters that require changes. For details, see File Collection Parameters below.
10. To get the public key and enter it into the dialog box, do the following:
    a. Select and copy the public key from the Event Source by running: `cat ~/.ssh/id_rsa.pub`
    b. Paste the public key in the **Eventsource SSH Key** field.
11. Click **OK**.

You need to restart file collection for your changes to take effect.

Stop and Restart File Collection

After you add a new event source that uses file collection, you must stop and restart the NetWitness File Collection service. This is necessary to add the key to the new event source.

File Collection Parameters

The following table provides descriptions of the File Collection source parameters.

The following table describes the **Basic** configuration parameter for File collection.

Note: Required parameters are marked with an asterisk. All other parameters are optional.

Name	Description
File Directory*	Collection directory (for example, **Eur_London100**) into which the File event source places its files. Valid value is a character string that is conforms to the following regular expression: **[_a-zA-Z][_a-zA-Z0-9]*** This means that the file directory must start with a letter followed by numbers, letters, and underscores. Do not modify this parameter after you start collecting event data. After you create the collection, the Log Collector creates the work, save, and error sub-directories under the collection directory.
Address*	IP address of the event source. Valid value is an **IPv4 address**, **IPv6 address**, or a **hostname** including a fully-qualified domain name.
File Spec	Regular expression. For example, **^.*$** = process everything.
File Encoding	Character encoding used by the syslog senders to this port. Defaults to UTF-8. **Note:** It is safe to leave this as UTF-8, since UTF-8 handles ASCII characters as well, and most senders have their encoding set to UTF-8. NetWitness has tested the following values: • EUC-KR • SJIS • GB3212/GBK • ISO_8859-1 (German) • ISO_8859-7 (Greek)
Cancel	Closes the dialog without making adding an event source type.
OK	Adds the parameters for the event source.

The following table describes the **Advanced** configuration parameter for File collection.

Name	Description

Name	Description
Ignore Encoding Conversion Errors	Select the check box to ignore encoding conversion errors and ignore invalid data. The check box is selected by default. **Caution:** This may cause parsing and transformation errors.
File Disk Quota	Determines when to stop saving files regardless of the **Save On Error** and **Save On Success** parameter settings. For example, a value of 10 indicates that when there is less than 10% available disk left, the Log Collector stops saving files to reserve enough space for your estimated normal collection processing. **Caution:** Available disk refers to a partition where the base collection directory is mounted. If the Log Decoder server has a 10TB disk size and 2TB is allocated to base collection directory, then setting this value to 10 causes log collection to stop when less than 0.2TB (10% of 2TB) of space is left. It does not mean 10% of 10TB. Valid value is a number in the **0** to **100** range. **10** is the default.
Sequential Processing	Sequential processing flag: • Select the check box (default) to process event source files in collection order. • Do not select the checkbox to process event source files in parallel.
Save On Error	Save on error flag. Check the checkbox to retain the **eventsource collection** file when the Log Collector it encounters an error. The check box is selected by default.
Save On Success	Save **eventsource collection** file after processing flag. Check the checkbox to save the eventsource collection file after processing it. The check box is not selected by default.
Eventsource SSH Key	SSH public key used to upload files for this event source. Please refer to the *Generate Key Pair on Event Source and Import Public Key to Log Collector* section in the Install and Update the SFTP Agent Guide for instructions on generating keys. **Note:** If File collection is stopped, NetWitness does not update the authorized_keys file with the SSH public key that you add or modify in this parameter. You must restart File collection to update the public key. You can add or modify the value of the public key in this parameter in multiple File event sources without File collection running, but NetWitness will not update the **authorized_keys** file until File collection is restarted.
Manage Error Files	By default, the Log Collector uses the **File Disk Quota** parameter to ensure that the disk does not fill up with error files. If you set this parameter to **true**, you can specify one of these: • Maximum space allotted to error files in the **Error Files Size** parameter. • Maximum number of error files allowed in **Error Files Count** parameter. A reduction percent is also specified, which tells the system how much to reduce when the maximum is reached. Select the check box to manage error files. The check box is not selected by default.

Name	Description
Error Files Size	Only valid if the **Manage Error Files** and **Save On Error** parameters are set to true. Specifies to what extent NetWitness saves error files. The value that you specify is the maximum total size of all the files in the error directory. Valid value is a number in **0** to **281474976710655** range. You specify these values in either **Kilobytes**, **Megabytes**, or **Gigabytes**. **100 Megabytes** is the default. If you change this parameter, the change does not take effect until you restart collection or restart the Log Collector service.
Error Files Count	Only valid if the **Manage Error Files** and **Save On Error** parameters are set to true. Maximum number of error files allowed in the error directory. Valid value is a number in **0** to **65536** range. **65536** is the default. If you change this parameter, the change does not take effect until you restart collection or restart the Log Collector service.
Error Files Reduction %	Percent amount by size or count of the error files that the Log Collector service removes when the maximum size or count has been reached. The service removes the oldest files first. Valid value is a number in the **0** to **100** range. **10** is the default.
Manage Saved Files	Select the check box to manage saved files. The check box is not selected by default. By default, the Log Collector uses the **File Disk Quota** parameter to ensure that the disk does not fill up with saved files. If check this check box, you can specify one of these: • Maximum space allotted to saved files in the **Saved Files Size** parameter. • Maximum number of saved files allowed in **Saved Files Count** parameter. A reduction percent is also specified, which tells the system how much to reduce when the maximum is reached.
Saved Files Size	Only valid if the **Manage Saved Files** and **Save On Success** parameters are set to true. Maximum total size of all the files in the save directory. Valid value is a number in the **0** to **281474976710655** range. You specify these values in either **Kilobytes**, **Megabytes**, or **Gigabytes**. **100 Megabytes** is the default. If you change this parameter, the change does not take effect until you restart collection or restart the Log Collector service.
Saved Files Count	Only valid if the **Manage Saved Files** and **Save On Success** parameters are set to true. Maximum number of saved files in the save directory. Valid value is a number in **0** to **65536** range. **65536** is the default. If you change this parameter, the change does not take effect until you restart collection or restart the Log Collector service.
Saved File Reduction %	Percent amount by size or count of the saved files that the Log Collector service removes when the maximum size or count has been reached. The service removes the oldest files first. Valid value is a number in the **0** to **100** range. **10** is the default.

Name	Description
Debug	**Caution:** Only enable debugging (set this parameter to **On** or **Verbose**) if you have a problem with an event source and you need to investigate this problem. Enabling debugging will adversely affect the performance of the Log Collector. Enables or disables debug logging for the event source. Valid values are: • **Off** = (default) disabled • **On** = enabled • **Verbose** = enabled in verbose mode - adds thread information and source context information to the messages. This parameter is designed to debug and monitor isolated event source collection issues. The debug logging is verbose, so limit the number of event sources to minimize performance impact. If you change this value, the change takes effect immediately (no restart required).

Configure Logstash Event Sources in NetWitness

You can configure the Logstash collection protocol.

Note: Make sure you start Syslog collection to send logs from Logstash to destination host.

IMPORTANT:
- Do not change logstash.yml file as it breaks the functionality.
- Do not change sincedb_path input configuration. If you change the sincedb_path, the back up and restore functionality breaks.

To configure a Logstash Event Source:

1. Go to  **(Admin) > Services** from the NetWitness menu.
2. Select a Log Collection service.
3. Under Actions, select  **> View > Config** to display the Log Collection configuration parameter tabs.
4. Click the **Event Sources** tab.
5. In the **Event Sources** tab, select **Logstash/Config** from the drop-down menu.
6. In the **Event Categories** panel toolbar, click  .

   The **Available Event Source Types** dialog is displayed: currently, **Custom** is the only available entry.
7. Select **Custom** and click **OK**.

The newly added event source type is displayed in the **Event Categories** panel.

8. Select the new type in the **Event Categories** panel and click  in the **Sources** toolbar.
   The **Add Source** dialog is displayed.
9. Fill in the fields, based on the Logstash event source you are adding. General details about the available parameters are described below in Logstash Collection Parameters.
10. Click **OK**.

Note: Once you configure the Logstash you must restart syslog collection for the changes to take effect.

Logstash Collection Parameters

The following tables provides descriptions of the Logstash Collection source parameters.

Note: Items that are followed by an asterisk (*) are required.

Basic Parameters

Custom Event Source

The following table lists the custom event source parameters.

Name	Description
Name *	Enter an alpha-numeric, descriptive name for the source. This value is only used for displaying the name on this screen.
Enabled	Select the checkbox to enable the event source configuration to start collection. The checkbox is selected by default.
Description	Enter a text description for the event source.
Input Configuration *	An input plugin enables a specific source of events to be read by Logstash. The following code represents an example input plugin. Paste in the text for your input plugin.
Filter Configuration *	A filter plugin performs intermediary processing on an event. Paste in any filter details for your Logstash event source.
Event Destination *	NetWitness consumes Syslog from Log Collectors and Log Decoders: specify the specific Log Collector or Log Decoder to which you want to send the information.
Test Connection	Checks the configuration parameters specified in this dialog to make sure they are correct.

Beats Event Source

The following table lists the beats event source parameters.

Name	Description
Name *	Enter an alpha-numeric, descriptive name for the source. This value is only used for displaying the name on this screen.
Enabled	Select the check box to enable the event source configuration to start collection. The check box is selected by default.
Description	Enter a text description for the event source.
Port Number*	Enter the port number (for example, 5044) that you configured for your event sources.
Linux-Audit	Select the checkbox to enable processing for Linux audit.
Linux-System	Select the checkbox to enable processing for Linux system.
Ngnix	Select the checkbox to enable processing for Nginx.
Event Destination*	Select the NetWitness Log Collector or Log Decoder to which event needs to be send from the drop-down list.
Test Configuration	Checks the configuration parameters specified in this dialog to make sure they are correct.

Export Connector Event Source

The following table lists the custom export connector event source parameters.

Name	Description
Name *	Enter an alpha-numeric, descriptive name for the source. This value is only used for displaying the name on this screen.
Enabled	Select the check box to enable the event source configuration to start collection. The check box is selected by default.
Description	Enter a text description for the event source.
Host*	Select the hostname of the Decoder or Log Decoder for data aggregation from the drop-down list.
Username*	Username used to access the Decoder or Log Decoder for data aggregation.

Name	Description
Authentication	**Note:** If you upgrade from NetWitness Platform 11.6.0.0 to 11.6.1.0, automatic key is generated and stored in the key store management for the password set in Logstash pipeline configuration. You can view the key instead of password in the **Authentication** field. Select the authentication type used for data aggregation. By default, **SSL** field is enabled, if you select **trusted authentication**. **Note:** For trusted authentication, make sure you add the PEM file at `/etc/pki/nw/node/node-cert.pem` to the source Decoders REST APIs (`/sys/trustpeer` and `/sys/caupload`).
SSL	Select the check box to communicate using SSL. The security of data transmission is managed by encrypting information and providing authentication with SSL certificates. By default, SSL option is enabled, if you select **trusted authentication** type in the **Authentication** field.
Decoder Type*	Decoder Type is a read only field and it is auto populated when you select the Host.
Output Configuration*	Logstash pipeline output configuration that send the input events.
Test Configuration	Checks the configuration parameters specified in this dialog to make sure they are correct.

Advanced Parameters

Click next to **Advanced** to view and edit the advanced parameters, if necessary.

Name	Description
Debug	**Caution:** Only enable debugging (set this parameter to On or Verbose) if you have a problem with an event source and you need to investigate this problem. Enabling debugging will adversely affect the performance of the Log Collector. **Caution:** Enables or disables debug logging for the event source. Valid values are: • **Off** = (default) disabled • **On** = enabled • **Verbose** = enabled in verbose mode - adds thread information and source context information to the messages. This parameter is designed to debug and monitor isolated event source collection issues. If you change this value, the change takes effect immediately (no restart required). The debug logging is verbose, so limit the number of event sources to minimize performance impact.
Beats SSL (This option is applicable only for beats typespec)	Select this checkbox to communicate using beats SSL. The security of data transmission is managed by encrypting information and providing authentication with SSL certificates. This check box is not selected by default. **Note:** Ensure that you copy the server SSL certificate and the key (generated in your system) to `/etc/logstash/pki` on Log Collector, which is used during SSL connection. `/etc/logstash/pki` is a path in the Log Collector node.
Certificate (This option is applicable only for beats typespec)	Select the name of a server SSL certificate located at `/etc/logstash/pki`.
Key (This option is applicable only for beats typespec)	Select the name of a server SSL key located at `/etc/logstash/pki`.

Name	Description
SSL Enabled	• For custom typespec - Select the check box to communicate using SSL. The security of data transmission is managed by encrypting information and providing authentication with SSL certificates. This check box is selected by default. • For beats typespec - Select the check box to communicate using SSL. The security of data transmission is managed by encrypting information and providing authentication with SSL certificates. This check box is selected by default. • For export connector typespec - Select the check box for Logstash to communicate with Packet Decoder and Log Decoder in SSL and Non-SSL mode. This check box is not selected by default.
Export Logs (This option is applicable only for export_ connector typespec)	Select the check box to export logs from Log Decoders.
Starting Session (This option is not applicable for export_ connector typespec)	Specify the session id from which you want to start the aggregation from instead of the default.
Include Metas (This option is not applicable for export_ connector typespec)	Specify the list of metas separated by comma that you want to aggregate. The default metas such as time, did, sessionid are collected in addition to the metas you added for aggregation.
Exclude Metas (This option is not applicable for export_ connector typespec)	Specify the list of metas separated by comma that you want to exclude from aggregration.
Query (This option is not applicable for export_ connector typespec)	Specify the query so the data matching the query is only aggregated. For example, select * where user.dst = 'john'.

Name	Description
Enable Metrics Collection (This option is applicable only for export_ connector typespec)	Enables metrics collection in Elastic. **IMPORTANT:** If you enable the metrics collection, you must provide the Elastic host, username, and password.
Elastic Host (This option is applicable only for export_ connector typespec)	Specify the Elastic host to forward metrics.
Elastic Port (This option is applicable only for export_ connector typespec)	Port number through which metrics are forwarded.
Elastic Username (This option is applicable only for export_ connector typespec)	Specify the name of the Elastic search.
Elastic Password (This option is applicable only for export_ connector typespec)	**Note:** If you upgrade from NetWitness Platform 11.6.0.0 to 11.6.1.0, automatic key is generated and stored in the key store management for the password set in Logstash pipeline configuration. You can view the key instead of password in the **Elastic Password** field. Specify the Elastic search password.
Minutes Back (This option is applicable only for export_ connector typespec)	Starts collecting data from last xx minutes. For example, if the value is set to 30 minutes, Log Collector starts collecting the logs and metas starting from last 30 minutes.

Name	Description
Persistent Queue (This option is applicable only for export_ connector typespec	Persistent queue stores the message queue on disk to avoid data loss. By default, this option is not enabled.
Additional Custom Configuration	Use this text box for any additional configuration, in case you have multiple inputs or another set of outputs to send somewhere in addition to a NetWitness Log Collector or Log Decoder. For example, you can configure the data to be sent to Elasticsearch. In this case each event that is sent to Netwitness Platform will also be send to Elasticsearch.
Required Plugins	Specify the required plugins in a comma separated list. **Note:** Backup and restore is not supported for custom plugins. **Note:** If the test connection failed due to required plugin is not installed, you must install the required plugin, for more information, see Install or Manage Logstash Plugin.
Pipeline Workers	Number of pipeline worker threads allocated for logstash pipeline.
Ports (This option is not applicable for export_ connector typespec)	Enter a port number (for example, 5000 or UDP:5000, TCP:5000) , and ensure the Enabled box is checked. This allows the plugins to collect logs over the network (For example, UDP, TCP). **IMPORTANT:** If you are configuring beats event source, make sure you provide beats event source port (For example, 5044) in the advance configuration even if you have updated the port in the basic parameters.
Exclude Metas (This option is not applicable for export_ connector typespec)	Specify the list of metas separated by comma that you want to exclude from aggregration.
Query (This option is not applicable for export_ connector typespec)	Specify the query so the data matching the query is only aggregated. For example, select * where user.dst = 'john'.

Note: You can monitor the health and throughput for Logstash pipeline using Logstash Input Plugin Overview dashboard. For more information, see "New Health and Wellness Dashboards" topic in the *System Maintenance Guide*.

View logstash collection in the Investigation > Events view

To view logstash collection in the Events view

1. Go to the **Investigation** > **Events view**.
2. Select a Log Decoder service
   (for example, LD1) which collects logstash events, from the drop-down list.
3. Select the time range.
4. Click  .
5. Look for events with a device.type value that matches the one defined in the Logstash pipeline.

   Note: Though some meta are parsed by default, a custom parser or Log Parser Rules are required for full parsing.

Install or Manage Logstash Plugin

By default, Logstash related plugins are installed when Logstash is installed. In addition, you can add or customize the plugins based on your requirement.

List All Lostash Plugins

You can list all the Logstash plugins available on your environment using the following command:

```
/usr/share/logstash/bin/logstash-plugin list --verbose
```

Install New Logstash Plugin

You can install new plugin using the following command:

```
/usr/share/logstash/bin/logstash-plugin install <plugin-name>
```

For example, see the following command:

```
/usr/share/logstash/bin/logstash-plugin install logstash-input-github
```

Manage Logstash Plugin

You can manage the existing plugins using the following commands:

- To update all Logstash plugins:

  ```
  /usr/share/logstash/bin/logstash-plugin update
  ```

- To update specific Logstash plugin:

  ```
  /usr/share/logstash/bin/logstash-plugin update <plugin-name>
  ```

Configure Keystore Management

Keystore management allows you to securely store secret values (key and password). These keys are used as a placeholder for authentication credentials within a Logstash pipeline configuration.

To configure keystore management:

1. Go to  (**ADMIN**) > **Services** from the NetWitness Platform menu.
2. Select a Log Collection service.
3. Under Actions, select  > **View** > **Config** to display the Log Collection configuration parameter tabs.
4. Click the **Event Sources** tab.
5. In the **Event Sources** tab, select **Logstash > Keystore management** from the drop-down menu.
6. Click  .
7. In the **key** field, enter the name of the key.
8. In the **password** field, enter the password.
9. Click **Save**.

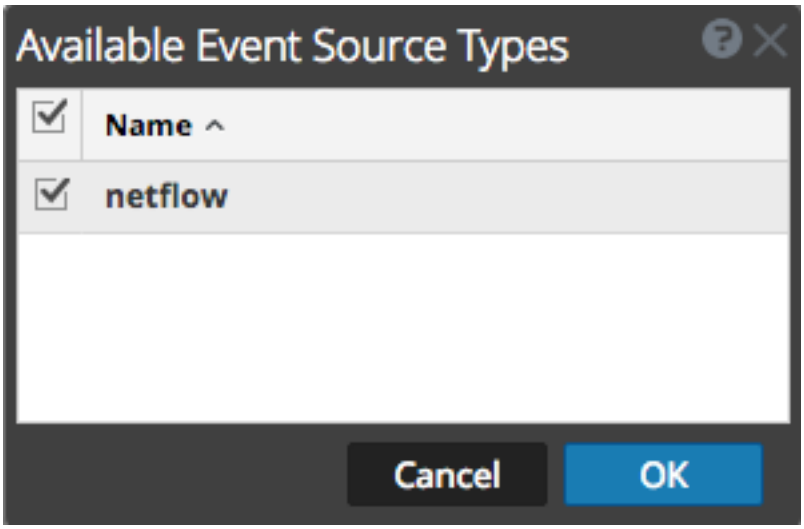
Configure Netflow Event Sources in NetWitness

This topic tells you how to configure the Netflow collection protocol.

Configure a Netflow Event Source

To configure a Netflow Event Source:

1. Go to  **(Admin) > Services** from the NetWitness menu.
2. Select a Log Collection service.
3. Select  **> View > Config** to display the Log Collection configuration parameter tabs.
4. Click the **Event Sources** tab.
5. In the **Event Sources** tab, select **Netflow/Config** from the drop-down menu.
6. In the **Event Categories** panel toolbar, click  .
   The **Available Event Source Types** dialog is displayed.
7. Select the **netflow** event source type and click **OK**.



The newly added event source type is displayed in the **Event Categories** panel.

8. Select the new type in the **Event Categories** panel and click + in the **Sources** toolbar.
   The **Add Source** dialog is displayed.
9. Enter a port number in the **Port** field, and ensure the Enabled box is checked.

   Note: NetWitness opens the 2055, 4739, 6343, and 9995 ports on the firewall by default. You can open other ports for Netflow if required.

   For details of other parameters, see Netflow Collection Parameters below.
10. Click **OK**.

The new event source is displayed in the list.

Netflow Collection Parameters

The following table provide descriptions of the basic Netflow Collection parameters.

Name	Description
Port	Specify the port number configured for the Netflow event source. NetWitness opens the 2055, 4739, 6343, and 9995 ports for Netflow by default. You can open other ports for Netflow if required.
Enabled	Select the check box to enable the event source configuration to start collection. The check box is selected by default.
Cancel	Closes the dialog without making adding an event source type.
OK	Adds the parameters for the event source.

The following table provide descriptions of the advanced Netflow Collection parameters.

Name	Description
InFlight Publish Log Threshold	Establishes a threshold that, when reached, NetWitness generates a log message to help you resolve event flow issues. The Threshold is the size of the netflow event messages currently flowing from the event source to NetWitness . Valid values are: • **0 (default)** - disables the log message. • **100-100000000** - generates a log message when this Log Collector has processed the specified number of netflow events. For example, if you set this value to 100, NetWitness generates a log message when 100 netflow events of the specific netflow version (v5 or v9) have been processed.
Debug	**Caution:** Only enable debugging (set this parameter to On or Verbose) if you have a problem with an event source and you need to investigate this problem. Enabling debugging will adversely affect the performance of the Log Collector . Enables or disables debug logging for the event source. Valid values are: • **Off** = (default) disabled • **On** = enabled • **Verbose** = enabled in verbose mode - adds thread information and source context information to the messages. This parameter is designed to debug and monitor isolated event source collection issues. The debug logging is verbose, so limit the number of event sources to minimize performance impact. If you change this value, the change takes effect immediately (no restart required).

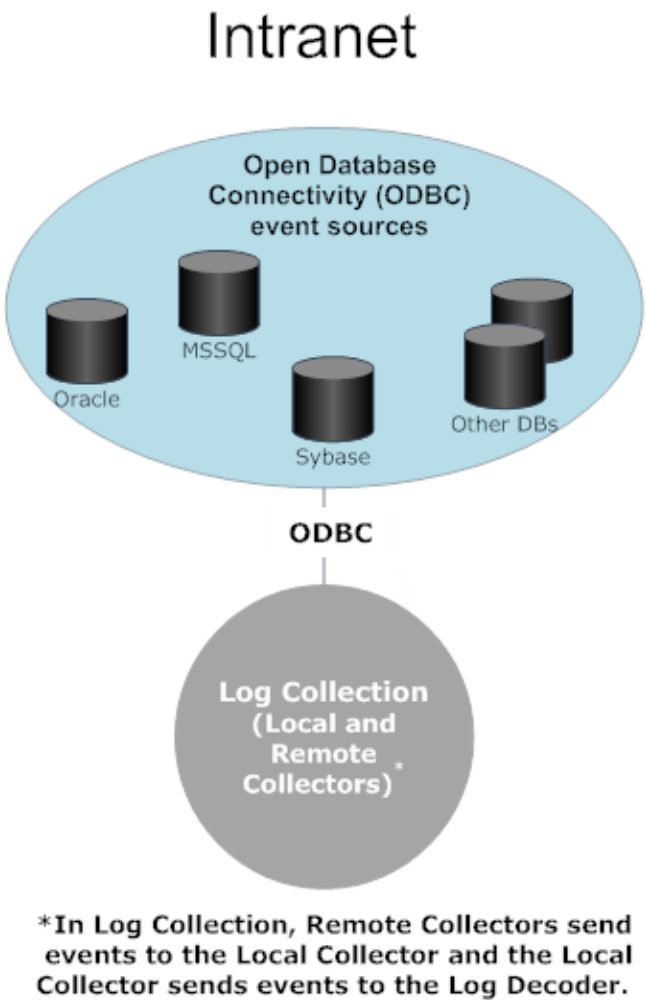
ODBC

Configure ODBC Event Sources in NetWitness

This topic tells you how to configure ODBC collection protocol which collects events from event sources that store audit data in a database using the Open Database Connectivity (ODBC) software interface.

Deployment Scenario

The following figure illustrates how you deploy the ODBC Collection Protocol in NetWitness.



Configure an ODBC Event Source

To configure an ODBC event source, you need to configure an event source type, and also choose a DSN template.

Configure a DSN

The following procedure describes how to add a DSN from an existing DSN template. For other procedures related to DSNs, see Configure Data Source Names (DSNs).

Configure a DSN (Data Source Name):

1. Go to **(Admin) > Services**.
2. In the **Services** grid, select a **Log Collector** service.
3. Click under **Actions** and select **View** > **Config**.
4. In the Log Collector **Event Sources** tab, select **ODBC/DSNs** from the drop-down menu.

5. The DSNs panel is displayed with the existing DSNs, if any.
6. Click  to open the **Add DSN** dialog.
7. Choose a DSN Template from the drop down menu and enter a name for the DSN. (You use the name when you set up the ODBC event source type.) If required, click 

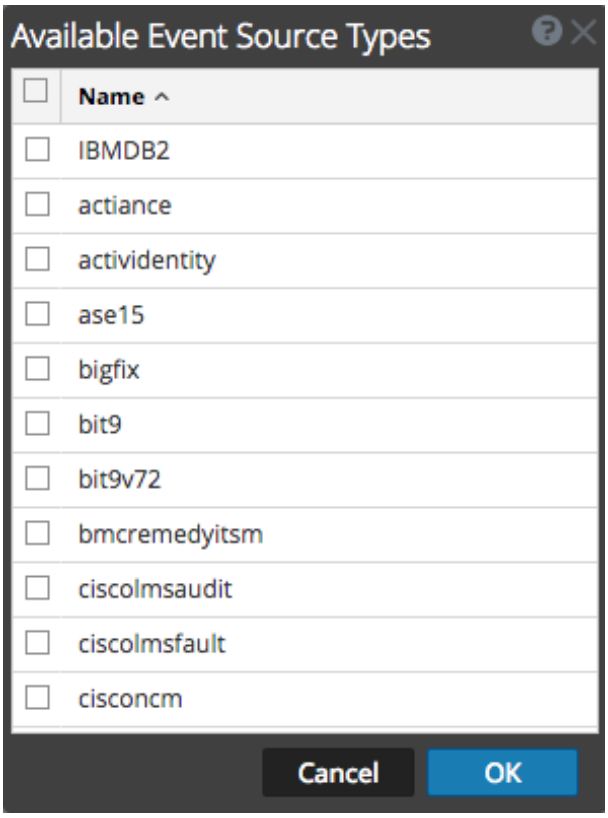
to add or delete DSN templates.
8. Fill in the parameters and click **Save**.

Add an Event Source Type

For details on parameters used in the following procedure, see ODBC Event Source Configuration Parameters.

To configure an ODBC Event Source Type:

1. Go to  **(Admin) > Services**.
2. Select a Log Collection service.
3. Under Actions, select  **> View > Config** to display the Log Collection configuration parameter tabs.
4. Click the **Event Sources** tab.
5. In the **Event Sources** tab, select **ODBC/Config** from the drop-down menu.
6. In the **Event Categories** panel toolbar, click  .
   The **Available Event Source Types** dialog is displayed.

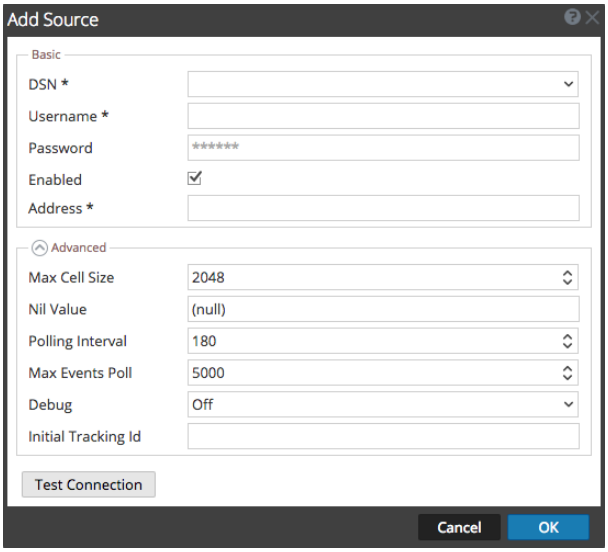


7. Select an event source category (for example **mssql** and click **OK**.

   The newly added event source type is displayed in the **Event Categories** panel.

8. Select the new type in the **Event Categories** panel and click  in the **Sources** toolbar.

   The **Add Source** dialog is displayed.



9. Select a DSN from the drop down list, specify or modify the other parameters as required, and click **OK**.
10. Click **Test Connection**.

    The result of the test is displayed in the dialog box. If the test is unsuccessful, edit the DSN information and retry.

    Note: Log Collector takes approximately 60 seconds to return the test results. If it exceeds the time limit, the test times out and the NetWitness server displays an error message.

11. If the test is successful, click **OK**.

The newly defined DSN is displayed in the **Sources** panel.

Configure Data Source Names (DSNs)

This topic tells you how to create and maintain DSNs for ODBC Collection.

Context

Open Database Connectivity (ODBC) event sources require Data Source Names (DSNs) so you need to define DSNs with their associate value pairs for ODBC event source configuration.

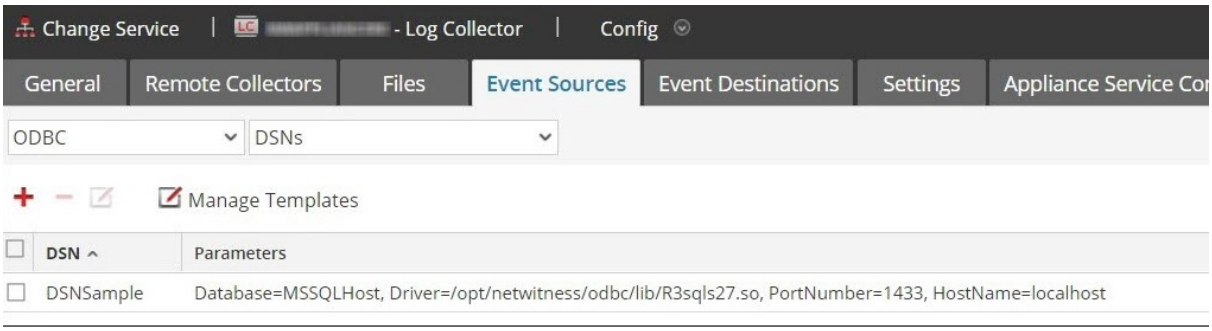
Navigate to the DSN Panel

To add or edit DSNs or DSN templates, first navigate to the appropriate screen.

To navigate to the DSN templates panel:

1. Go to  **(Admin) > Services**.
2. In the **Services** grid, select a **Log Collector** service.
3. Click  under **Actions** and select **View > Config**.
4. In the **Log Collector Event Sources** tab, select **ODBC/DSNs** from the drop-down menu.

   The **DSNs** panel is displayed with the DSNs that are added, if any.



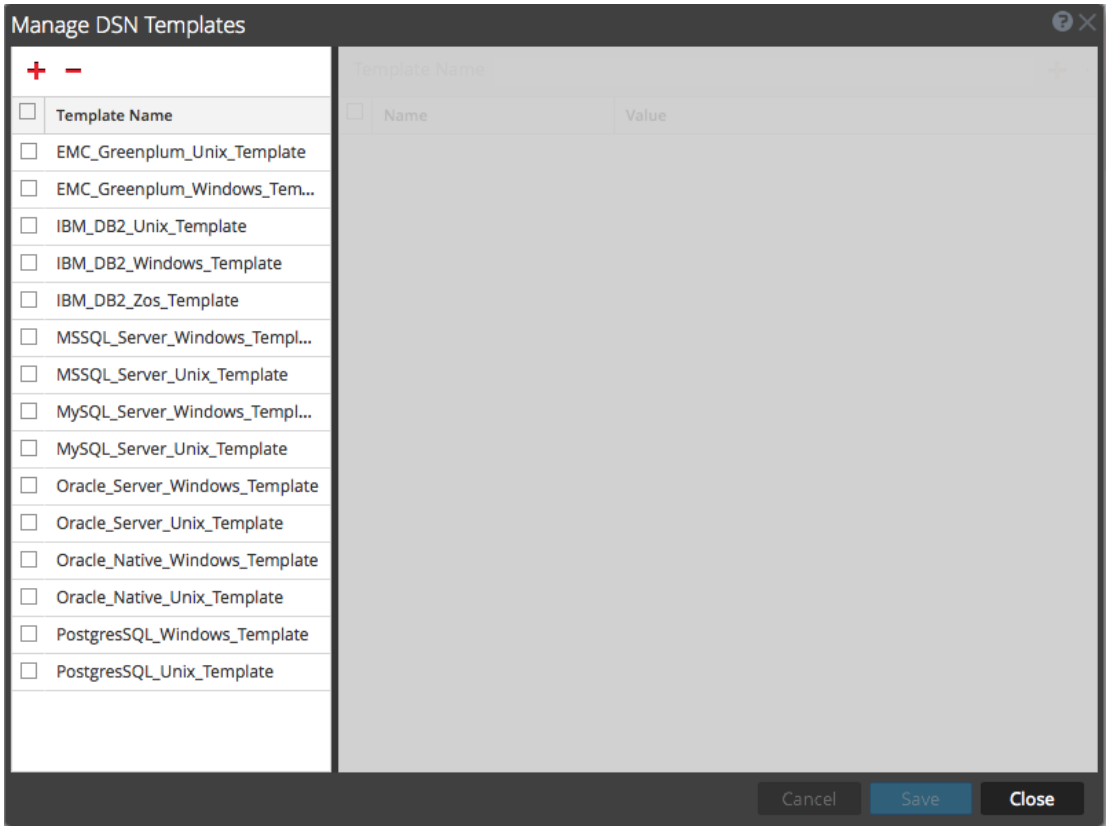
From this screen, you can perform the following actions:

- Add a new DSN template
- Add a DSN from an existing template
- Add a DSN by editing an existing DSN template
- Remove a DSN or DSN template

Add a New DSN Template

If none of the predefined DSN templates fit your needs, use this procedure to add a DSN template.

1. From the DSNs panel, click Manage Templates.
   The **Manage DSN Templates** dialog is displayed.

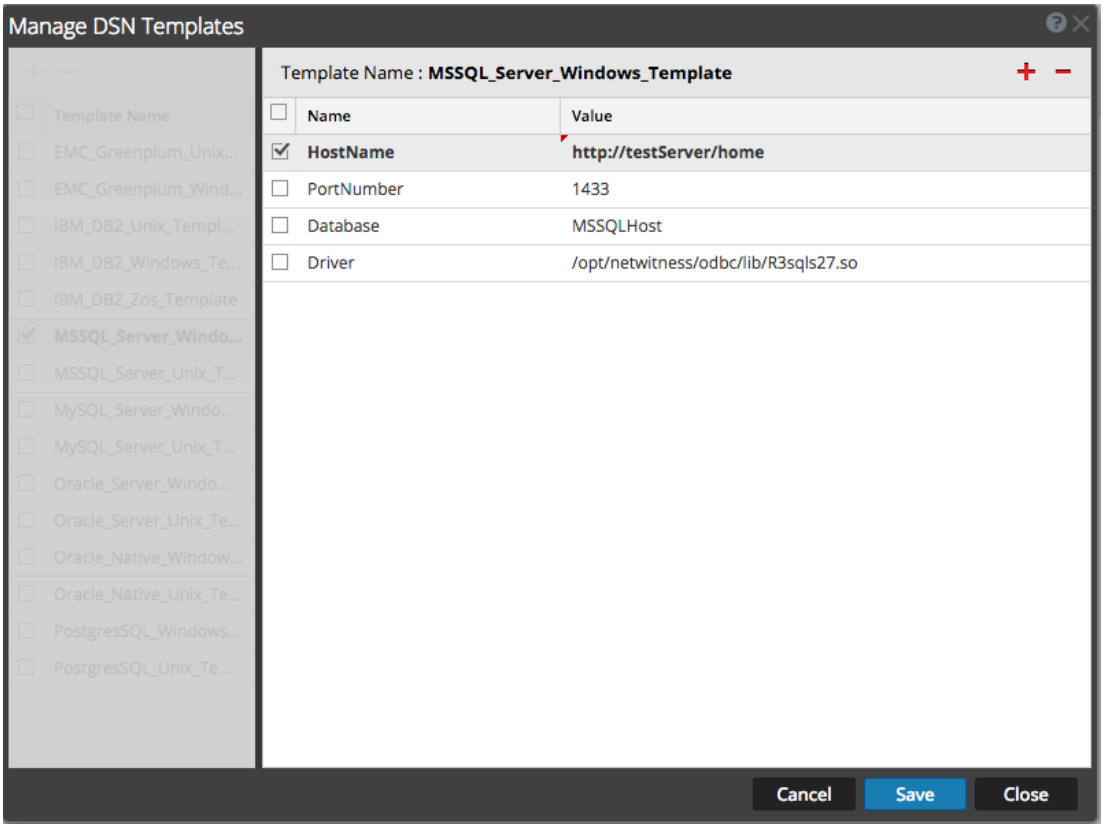


Note: NetWitness provides default templates on the left side panel that you can use while adding a new DSN.

2. Click + .
   The right panel is activated.

3. Specify a template name and click + on the right panel to add parameters.

4. Specify the parameters. Click **Save**.



The new DSN template is added in the **Manage DSN Templates** list.

Add a DSN from an existing template

You can select an existing template, and fill in the parameters for your needs.

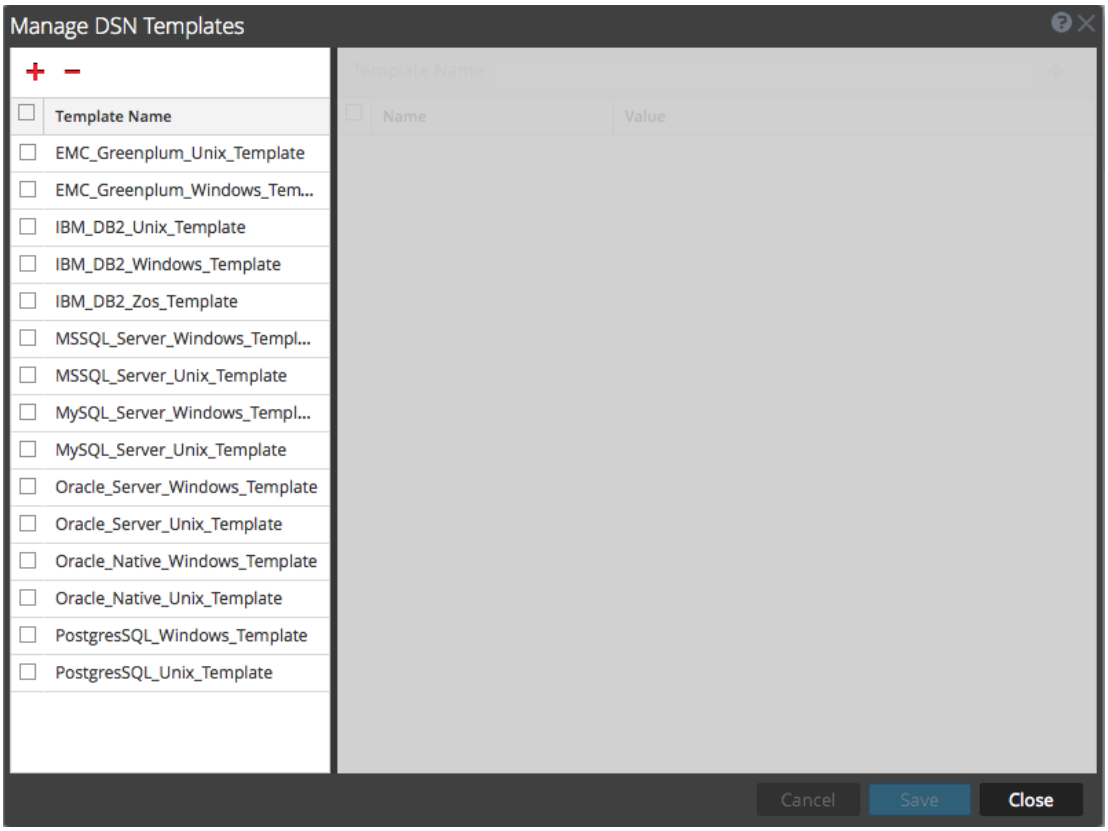
1. From the DSNs panel, click + to open the Add DSN dialog box.
   The **Add DSN** dialog is displayed with existing DSNs, if any
2. Choose a DSN Template from the drop down menu and enter a name for the DSN. (You use the name when you set up the ODBC event source type.)
3. Fill in the parameters and click **Save**.

Your DSN is added to the list of DSNs.

Add a New DSN by editing an existing DSN template

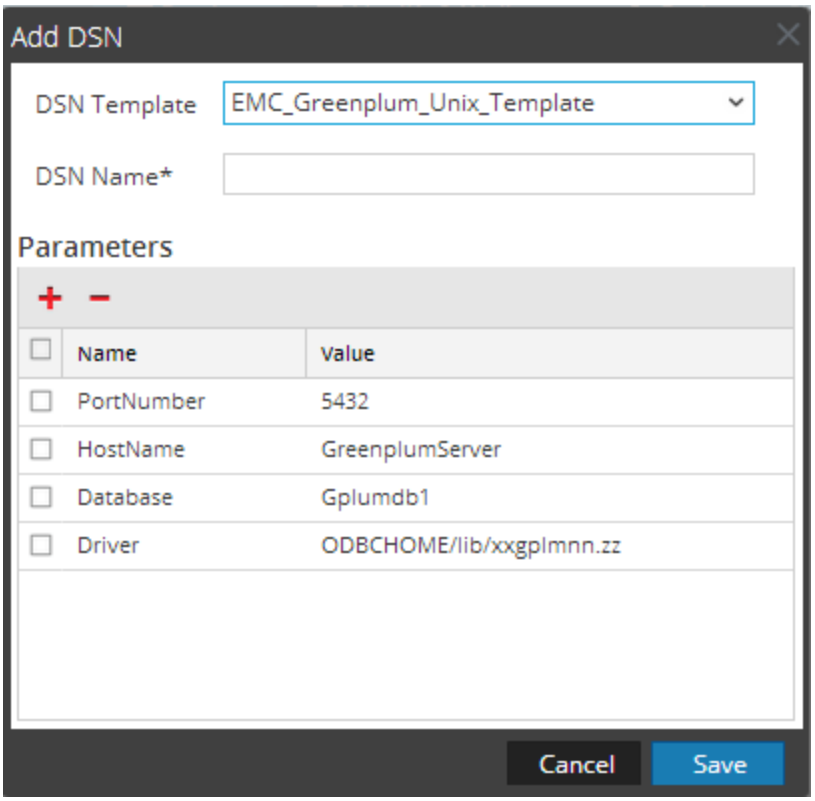
You can add a DSN by updating an existing DSN template to fit your needs.

1. From the DSNs panel, click Manage Templates.
   The **Manage DSN Templates** dialog is displayed.



2. Select the existing template that you want to modify.

   The right panel is activated, and the default parameters for the selected template are displayed.



3. Specify a name in the **DSN Name** field.
4. Add, delete or edit the default parameters.
5. Once you have the set of required parameters, click **Save**, then **Close**.
6. Choose the DSN Template that you updated from the drop down menu and enter a name for the DSN. (You use the name when you set up the ODBC event source type.)
7. Fill in the parameters and click **Save**.

Your DSN is added to the list of DSNs.

Remove a DSN or DSN template

If you no longer use a DSN or a DSN template, you can remove it from the system.

To remove an existing DSN:

1. From the DSNs panel, select an existing DSN.
2. Click —.

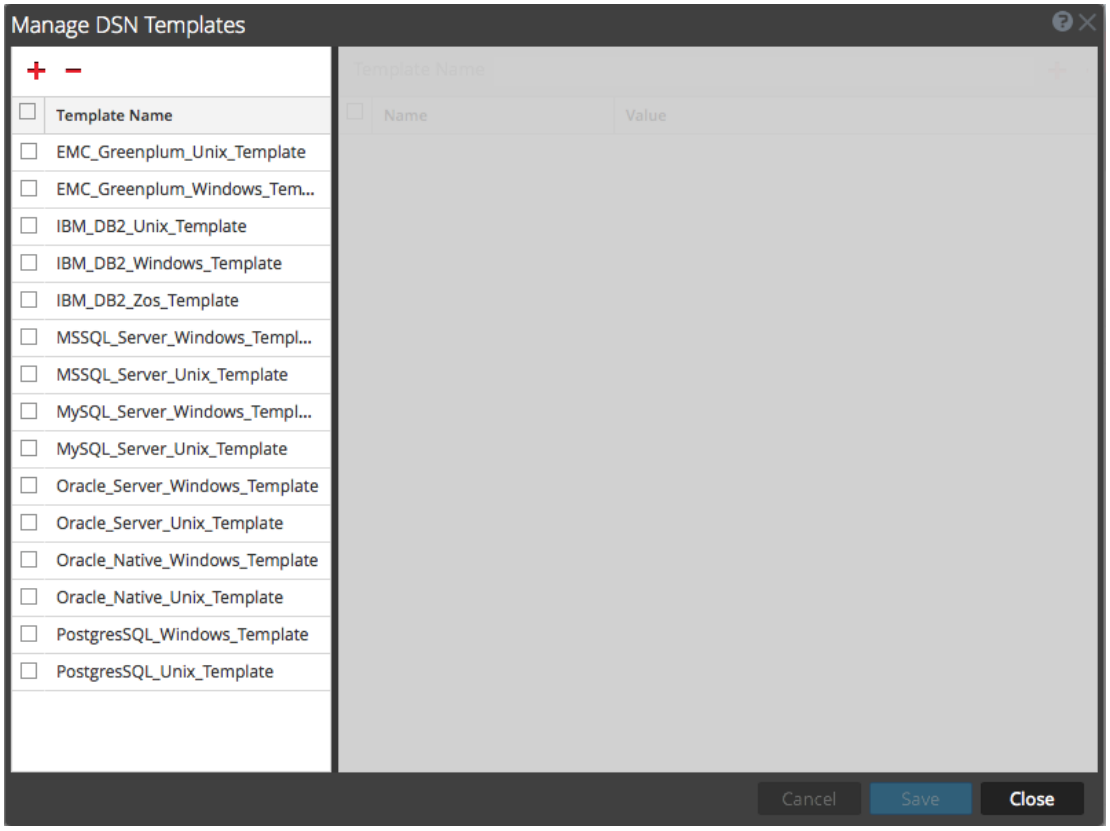
   A Warning message appears, asking whether you are sure you want to delete the DSN.
3. To delete the DSN, click **Yes**. Alternatively, to cancel the deletion, click **No**.

If you confirmed the deletion, the selected DSN is removed from the system.

To remove an existing DSN Template:

1. From the DSNs panel, click Manage Templates.

   The **Manage DSN Templates** dialog is displayed.



2. From the DSNs panel, select an existing DSN Template.
3. Click —.

   A Confirmation message appears, asking whether you are sure you want to delete the DSN Template.
4. To delete the DSN Template, click **Yes**. Alternatively, to cancel the deletion, click **No**.

If you confirmed the deletion, the selected DSN Template is removed from the system.

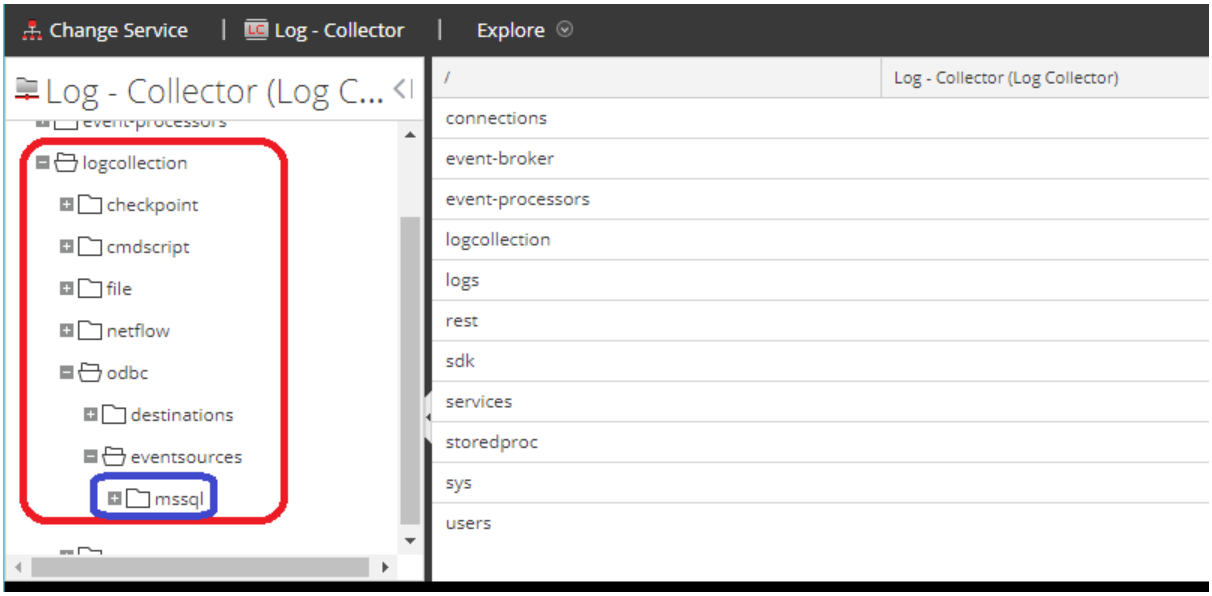
Configure device.ip meta for ODBC Data Source

For any ODBC event source, you can choose to have the ODBC collector populate the **device.ip** meta value with either the event source IP address or the actual source IP on which logs are being collected.

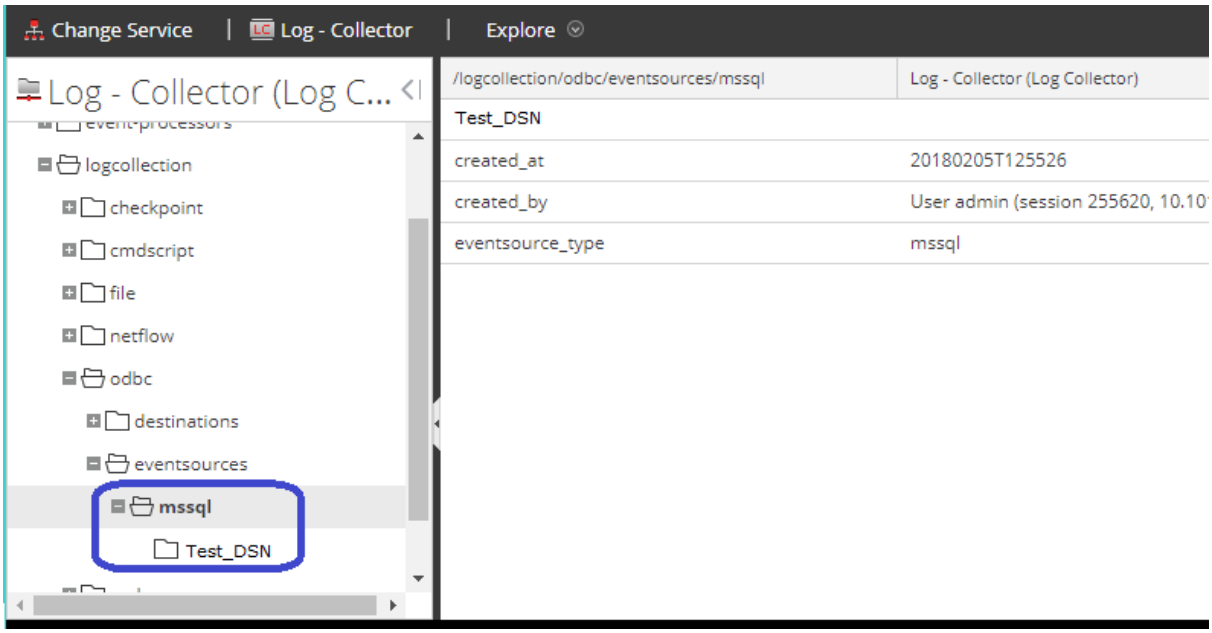
To view or set this parameter:

1. Go to **ADMIN > Services**.
2. In the **Services** grid, select a **Log Collector** service.
3. Click under **Actions** and select **View > Explore**.
4. Navigate to **logcollection > odbc > eventsources**.

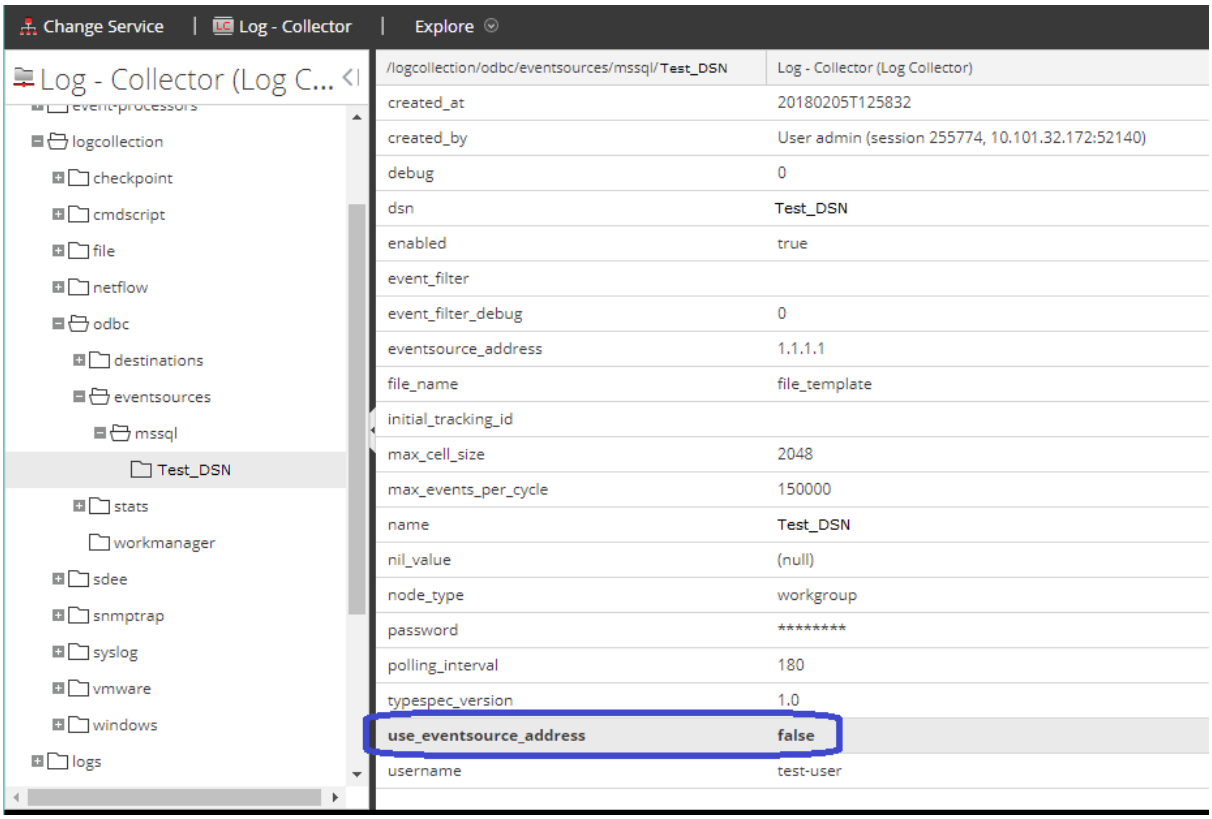
There are entries for each ODBC event source that you have configured in NetWitness. For example, for this installation, the only ODBC event source currently configured is MS SQL:



5. Click + next to an event source to expand it and see its DSN entry.



6. Click the DSN entry (in this example **Test_DSN**) to show the parameters.
7. The **use_eventsource_address** parameter is listed.



- **False**: actual source IP address is used. This is the default value.
- **True**: event source IP address is used.

8. Click on the value (in this case **False**), and type in the new value.

> **Note:** If you type anything other than **true** or **false** (case does not matter), you receive an error saying that the value you entered cannot be set.

Any changes you make take effect immediately.

Create Custom Typespec for ODBC Collection

NetWitness uses type specification (typespec) files for ODBC and file collection. These files act on raw log files, and are used for two main purposes:

- Define where in the log file data resides. For instance, some log files contain header information that is not considered data to be parsed.
- Replace certain types of characters that the log parser cannot parse correctly. For instance, the tab character can sometimes cause problems.

This topic tells you how to create a custom typespec for the Log Collector. The topic includes:

- Create Custom typespec procedure
- ODBC Collection typespec syntax

- Sample ODBC Collection typespec files

Create Custom Typespec

To create a custom typespec file:

1. Open an SFTP client (for example, WinSCP) and connect to a Log Collector or remote Log Collector.
2. Navigate to `/etc/netwitness/ng/logcollection/content/collection/odbc`, and copy an existing file, for example **bit9.xml**.
3. Modify the file according to your requirements. See ODBC Collection Typespec Syntax for details.
4. Rename and save the file to the same directory.
5. Restart the Log Collector.

Note: You will not be able to see new Event Source type in NetWitness until you restart the Log Collector.

ODBC Collection Typespec Syntax

The following table describes the typespec parameters.

Parameter	Description
name	The display name of your ODBC event source (for example, **activeidentity**). NetWitness displays this name in the **Sources** panel of the **View > Config > Events Sources** tab. Valid value is an alphanumeric string. You cannot use - (dashes), _(underscores), or spaces. The name must be unique across all typespec files in the folder.
type	Event source type: **odbc**. Do not modify this line.
prettyName	User-defined name for the event source. You can use the same value as name (for example, apache) or use a more descriptive name.
version	Version of this typespec file. Default value is 1.0.
author	Person who created the typespec file. Replace author-name with your name.
description	Formal description of the event source. Replace formal-description with your description of the event source.
<device> Section	
parser	This optional parameter contains the name of the log parser. This value forces the Log Decoder to use the specified log parser when parsing logs from this event source. **Note:** Please leave the field blank when unsure of the log parser to be used.
name	Name your ODBC event source (for example, **ActivIdentity ActivCard AAA Server**).
maxVersion	The version number of the event source (for example, **6.4.1**).
description	Description of the event source.
<collection> Section	

Parameter	Description
odbc	The syntax under <odbc> is used for event collection and processing. You can provide multiple queries for the same event source type by adding <query> tags.
query	This section contains the details of the query used to collect information from the event source.
tag	The prefix tag you want to add to events during transformation (for example **ActivIdentity**).
outputDelimiter	Specify the delimiter to use to separate fields. Specify any of the following values: • || (piping) • ^ (caret) • , (comma) • : (colon) • **0x20** (to represent a space)
interval	Specify the number of seconds between events. Default value is **60**.
dataQuery	Specify the query to fetch data from the ODBC eventsource database for SQL-syntax. For example: `SELECT acceptedrejected, servername, serveripa, sdate, millisecond, suid, groupname, ipa, reason, info1, info2, threadid FROM A_AHLOG WHERE sdate > '%TRACKING%' ORDER BY sdate`
maxTrackingQuery	The query used on the initial pull of events to identify the starting point within the data set to begin pulling logs from. After the initial pull, this query is no longer used, unless the **maxTracking** value has been reset or altered. For example: `SELECT MAX(Event_Id) from ExEvents`
trackingColumn	The tracking column value used when the ODBC collector pulls a new set of events.

Sample ODBC Collection Typespec Files

The following sample is the typespec file for the IBM ISS SiteProtector event source.

```
<?xml version="1.0" encoding="UTF-8"?>
<typespec>

   <name>siteprotector4_x</name>
   <type>odbc</type>
   <prettyName>SITEPROTECTOR4_X</prettyName>
   <version>1.0</version>
   <author>Administrator</author>
   <description>Collects events from SiteProtector</description>

   <device>
      <name>Internet Security Systems, Inc. RealSecure SiteProtector v 2.0</name>
      <maxVersion>2.0</maxVersion>
```

```
      <description></description>
      <parser>iss</parser>
   </device>

   <configuration>
   </configuration>

   <collection>
      <odbc>
         <query>
            <tag></tag>
            <outputDelimiter></outputDelimiter>
            <interval></interval>
            <dataQuery></dataQuery>
            <maxTrackingQuery></maxTrackingQuery>
            <trackingColumn></trackingColumn>
            <levelColumn></levelColumn>
            <eventIdColumn></eventIdColumn>
            <addressColumn></addressColumn>
         </query>
      </odbc>
   </collection>
</typespec>
```

The following sample is the typespec file for the Bit9 Security Platform event source.

```
<?xml version="1.0" encoding="UTF-8"?>
<typespec>

   <name>bit9</name>
   <type>odbc</type>
   <prettyName>BIT9</prettyName>
   <version>1.0</version>
   <author>Administrator</author>
   <description>Bit9 Events</description>

   <device>
      <name>Bit9</name>
      <parser>bit9</parser>
   </device>

   <configuration>
   </configuration>
```

```
   <collection>
      <odbc>
         <query>
            <tag>BIT9</tag>
            <outputDelimiter>||</outputDelimiter>
            <interval>10</interval>
            <dataQuery>
            SELECT
            Timestamp,
            Event_Id,
            Computer_Id,
            File_Catalog_Id,
            Root_File_Catalog_Id,
            Priority,
            Type,
            Subtype,
            IP_Address,
            User_Name,
            Process,
            Description
            FROM
            ExEvents
            WHERE
            Event_Id > '%TRACKING%'
            </dataQuery>
            <trackingColumn>Event_Id</trackingColumn>
            <maxTrackingQuery>SELECT MAX(Event_Id) from ExEvents</maxTrackingQuery>
            <eventIdColumn></eventIdColumn>
         </query>
      </odbc>
   </collection>
</typespec>
```

Troubleshoot ODBC Collection

You can troubleshoot problems and monitor ODBC collection by reviewing the ODBC collector log informational, warning, and error messages to during execution of collection.

Each ODBC log messages includes the:

- Timestamp
- Category: `debug`, `info`, `warning`, or `failure`
- collection method = `OdbcCollection`
- ODBC event source type (GOTS-name) = Generic ODBC Type Specification name that you configured for the event source.
- collection function completed or attempted (for example, `[processing]`)
- ODBC event source name (DSN-name) = Data Source Name that you configured for the event source.
- description (for example, how many events the Log Collector collected)
- tracking ID = the Log Collector position in the target database table.

The following example illustrates the message you would receive upon successful collection of an ODBC event:

```
2014-July-25 17:21:25 info (OdbcCollection) : [event-source] [processing]
[event-source] Published 100 ODBC events: last tracking id: 2014-July-25
13:22:00.280
```

The following example illustrates a message you may receive upon unsuccessful collection of an ODBC event:

Log Message	`timestamp failure (OdbcCollection: [event-source] [processing] [event-source-type] Failed during doWork: Unable to prepare statement: state: S0002; error-code:208; description: [RSA] [ODBC-driver][event-source-type]Invalid object name 'object-name'.`
Possible Cause	ODBC collection failed while accessing the ODBC Driver or the target database.
Solutions	Validate the DSN value pairs for the events source.

Configure SDEE Event Sources in NetWitness

This topic tells you how to configure the SDEE collection protocol.

To add an SDEE Event Source:

1. Go to  **(Admin) > Services**.
2. Select a Log Collection service.
3. Under Actions, select  **> View > Config** to display the Log Collection configuration parameter tabs.
4. Click the **Event Sources** tab.

5. In the **Event Sources** tab, select **SDEE/Config** from the drop-down menu.

   The Event Categories panel displays the SDEE event sources that are configured, if any.

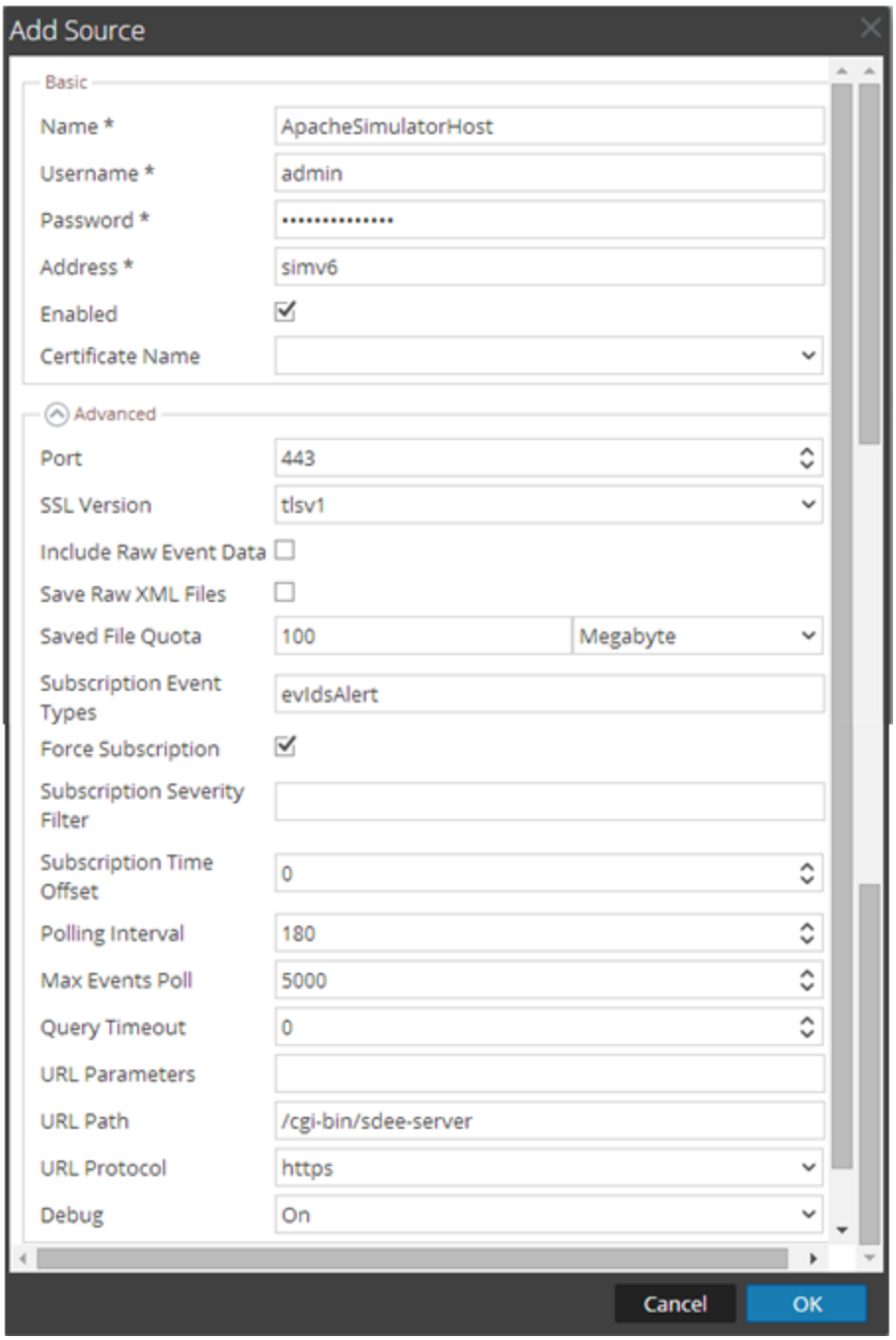
6. In the **Event Categories** panel toolbar, click  .

   The **Available Event Source Types** dialog is displayed.

7. Select an event source type and click **OK**.

   The newly added event source type is displayed in the **Event Categories** panel.

8. Select the new type in the Event Categories panel and click  in the Sources panel toolbar. The Add Source dialog is displayed.



9. Add a Name, Username, Address, and Password, and modify any other parameters that require changes, and click **OK**.

Configure SNMP Event Sources in NetWitness

This topic tells you how to configure the SNMP collection protocol.

Configure the SNMP Trap Event Source

To add the SNMP Event Source:

Note: If you have previously added the **snmptrap** type, you cannot add it again. You can edit it, or manage users.

1. Go to **(Admin) > Services** from the NetWitness menu.
2. Select a Log Collection service.
3. Under Actions, select **> View > Config** to display the Log Collection configuration parameter tabs.
4. Click the **Event Sources** tab.
5. In the **Event Sources** tab, select **SNMP/Config** from the drop-down menu.
6. In the **Event Categories** panel toolbar, click .
   The **Available Event Source Types** dialog is displayed.
7. Select the **snmptrap** event source type and click **OK**.
   The newly added event source type is displayed in the **Event Categories** panel.
8. Select **snmptrap** in the Event Categories panel.
9. Select **snmptrap** in the Sources panel and then click the Edit icon, , to edit the parameters.
10. Update any of the parameters that you need to change and click **OK**.

(Optional) Configure SNMP Users

If you are using SNMPv3, follow this procedure to update and maintain the SNMP v3 users.

To configure SNMPv3 Users:

1. Go to **(Admin) > Services**.
2. In the **Services** grid, select a **Log Collector** service.
3. Click under **Actions** and select **View** > **Config**.
4. In the Log Collector **Event Sources** tab, select **SNMP/SNMP v3 User Manager** from the drop-down menu.

The SNMPv3 User panel is displayed with the existing users, if any.

5. Click  to open the **Add SNMP User** dialog.
6. Fill in the dialog with the necessary parameters. The available parameters are described below.

SNMP User Parameters

The following table describes the parameters that you need to enter when you create an SNMPv3 user.

Note: Required parameters are marked with an asterisk. All other parameters are optional.

Parameter	Description
Username *	User name (or more accurately in SNMP terminology, security name). NetWitness uses this parameter and the **Engine ID** parameter to create a user entry in the SNMP engine of the collection service. The **Username** and **Engine ID** combination must be unique (for example, **logcollector**).
Engine ID	(Optional) Engine ID of the event source. For all event sources sending SNMP v3 traps to this collection service, you must add the username and engine id of the sending event source. For all event sources sending SNMPv3 informs, you must add just the username with a blank engine id.
Authentication Type	(Optional) Authentication protocol. Valid values are as follows: • **None** (default) - only security level of **noAuthNoPriv** can be used for traps sent to this service • **SHA** - Secure Hash Algorithm • **MD5 - Message Digest Algorithm** **IMPORTANT:** DO NOT USE: do not select MD5, as it conflicts with the Log Collector running in FIPS mode.
Authentication Passphrase	Optional if you do not have the **Authentication Type** set. Authentication passphrase.
Privacy Type	(Optional) Privacy protocol. You can only set this parameter if Authentication Type parameter is set. Valid values are as follows: • **None** (default) • **AES** - Advanced Encryption Standard • **DES - Data Encryption Standard** **IMPORTANT:** DO NOT USE: do not select DES, as it conflicts with the Log Collector running in FIPS mode.

Parameter	Description
Privacy Passphrase	Optional if you do not have the **Privacy Type** set. Privacy passphrase.
Close	Closes the dialog without adding the SNMPv3 user or saving modifications to the parameters.
Save	Adds the SNMPv3 user parameters or saves modifications to the parameters.

Configure Syslog Event Sources

This topic tells you how to configure Syslog event sources for the Log Collector.

Note: Prior to NetWitness 11.3, you did not configure Syslog Collection for Local Log Collectors: syslog collection was only configurable for Remote Collectors. You ***can*** configure Syslog for local Log Collectors that are on version 11.3 or later.

Configure a Syslog Event Source

For Remote or Virtual Log Collectors, syslog listeners for UDP on port 514, TCP on port 514 and SSL on port 6514 are created by default. You should not change the SSL settings on the TCP and SSL listeners. If you need SSL certificate verification, create a new event source type to listen on a different port.

Note: For local Log Collectors, you cannot create syslog listeners on ports 514 and 6514: these ports are used by the Log Decoder service.

To configure the Log Collector for Syslog collection:

1. Go to  **(Admin) > Services**.
2. In the Services grid, select a Log Collector, and from the Actions menu, choose  **> View > Config**.
3. Select the **Event Sources** tab.
4. Select **Syslog/Config** from the drop-down menu.

   The Event Categories panel displays the Syslog event sources that are configured, if any.

   Note: For NetWitness, some Syslog event sources are available by default. In this case, you can proceed to step 6.

5. In the Event Categories panel toolbar, click  .

   The Available Event Source Types dialog is displayed.
6. Select either **syslog-tcp** or **syslog-udp**. You can set up either or both, depending on the needs of your organization.

7. Select the new type in the Event Categories panel and click  in the Sources panel toolbar.

   The Add Source dialog is displayed.

8. Enter the port number, and select **Enabled**. Optionally, configure any of the Advanced parameters as necessary.

   Click **OK** to accept your changes and close the dialog box.

Once you configure one or both syslog types, the Log Decoder or Log Collector collects those types of messages from all available event sources. So, you can continue to add Syslog event sources to your system without needing to do any further configuration in NetWitness.

Character Encodings

For each protocol (TCP, UDP), you need to instantiate a separate syslog listener for each character encoding.

For example, assume that you have legacy event sources sending syslog in UDP and TCP with EUC-KR and EUC-JP encodings. You would need to configure four listeners for the Log Collector, and configure the syslog event sources to send to separate ports as follows:

- UDP ***<port1>*** for EUC-KR
- UDP ***<port2>*** for EUC-JP
- TCP ***<port3>*** for EUC-KR
- TCP ***<port4>*** for EUC-JP

Alternatively, you could use separate Remote/Virtual Log Collectors, each using the default port 514.

Syslog Parameters

The following tables describe the available basic and advanced parameters for Syslog configuration.

Note: Required parameters are marked with an asterisk. All other parameters are optional.

Basic Parameters

Name	Description
Port*	Enter the port number that you configured for your event sources.
Enabled	Select the check box to enable the event source configuration to start collection. The check box is selected by default.
SSL Receiver	**Note:** This parameter applies to NetWitness version 11.1 and newer. It is available only for the **syslog-tcp** Event Category. If you select the check box, the event source accepts SSL/TLS connections only. Also, if you change this setting, you must stop and restart Syslog collection for the change to become effective.

Name	Description
Encoding	Character encoding used by the syslog senders to this port. Defaults to UTF-8. **Note:** It is safe to leave this as UTF-8, since UTF-8 handles ASCII characters as well, and most senders have their encoding set to UTF-8. NetWitness has tested the following values: • EUC-KR • SJIS • GB3212/GBK • ISO_8859-1 (German) • ISO_8859-7 (Greek)

Advanced Parameters

Name	Description
Inflight Publish Log Threshold	Establishes a threshold that, when reached, NetWitness generates a log message to help you resolve event flow issues. The Threshold is the size of the syslog event messages currently flowing from the event source to NetWitness. Valid values are: • **0** (default) - disables the log message • **100-100000000** - generates log message when the syslog event messages currently flowing from the event source to NetWitness are within the 100 to 100000000 byte range.
Maximum Receivers	Maximum number of receiver resources used to process collected syslog events. The default value is **2**.
Event Filter	Select a filter. Refer to Configure Event Filters for a Collector for instructions on how to define filters.

Name	Description
Debug	**Caution:** Only enable debugging (set this parameter to "On" or "Verbose") if you have a problem with an event source and you need to investigate this problem. Enabling debugging will adversely affect the performance of the Log Collector. Enables or disables debug logging for the event source. Valid values are: • **Off** = (default) disabled • **On** = enabled • **Verbose** = enabled in verbose mode - adds thread information and source context information to the messages. This parameter is designed to debug and monitor isolated event source collection issues. The debug logging is verbose, so limit the number of event sources to minimize performance impact. If you change this value, the change takes effect immediately (no restart required).
SSL Verify Mode	**Note:** This parameter applies to NetWitness version 11.1 and newer. It is available only for the **syslog-tcp** Event Category. This setting is relevant only if the **SSL Receiver** setting is selected. If you change the SSL Verify Mode, you must stop and restart Syslog collection for the change to become effective. Available options: • **verify-none**: (default) The server does not verify the client's certificate, if any. A client can connect without presenting a certificate. • **verify-peer**: The server verifies the client's certificate, if any. A client can connect without presenting a certificate. **Note:** If verification fails, a warning is logged but the messages will still be accepted. • **verify-peer-fail-if-no-cert**: The client must present a certificate and the server will verify it. **Note:** If you use this mode, the client's CA certificate ***must*** be uploaded to the Log Collector's truststore using the REST API at `http://LC-ip-address:50101/sys/caupload`
Certificate Directory Path	You can configure custom certificates for the syslog listener on Log Collectors. For details, see (Optional) Configure Custom Certificates on Log Collectors.

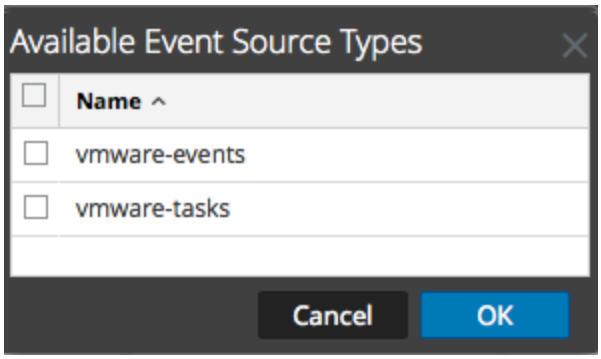
Configure VMware Event Sources in NetWitness

This topic tells you how to configure the VMware collection protocol.

To add a VMware Event Source:

1. Go to **(Admin) > Services**.
2. Select a Log Collection service.
3. Under Actions, select **> View > Config** to display the Log Collection configuration parameter tabs.
4. Click the **Event Sources** tab.
5. In the Log Collector **Event Sources** tab, select **VMware/Config** from the drop-down menu.

   The Event Categories panel displays the VMware event sources that are configured, if any.
6. Click to open the **Available Event Source Types** dialog.



7. Select **vmware-events** or **vmware-tasks** from the Available Event Source Types dialog and click **OK**.

   The VMware available event source types are as follows:

   - **vmware-events:** Setup vmware-events to collect events from vCenter Servers and ESX/ESXi servers.
   - **vmware-tasks:** (Optional) Setup vmware-tasks to collect tasks from vCenter Servers.
8. Select the new type in the Event Categories panel, and click + in the Sources toolbar.
9. Add a Name, Username and Password, and modify any other parameters that require changes.

   Caution: If you need to enter the domain name as part of the Username, you must use a double-backslash as a separator. For example, if the domain|username is corp\smithj, you must specify **corp\\smithj**.
10. Click **OK** to save your changes.

Configure Windows Event Sources in NetWitness

This topic tells you how to configure the Windows collection protocol.

In NetWitness, you need to configure the Kerberos Realm, and then add the Windows Event Source type.

To configure the Kerberos Realm for Windows collection:

1. Go to **(Admin) > Services**.
2. Select a Log Collection service.
3. Under Actions, select **> View > Config** to display the Log Collection configuration parameter tabs.
4. Click the **Event Sources** tab.
5. Select **Windows/Kerberos Realm** from the drop-down menu.
6. In the Kerberos Realm Configuration panel toolbar, click to add a new realm.
   The Add Kerberos Domain dialog is displayed.
7. Fill in the parameters, using the guidelines below.

Parameter	Details
Kerberos Realm Name	Enter the realm name, in all caps. For example, DSNETWORKING.COM. Note that the Mappings parameter is automatically filled with variations on the realm name.
KDC Host Name	Enter the name of the Domain Controller. ***Do not*** use a fully qualified name here: just the host name for the DC. **Note:** Make sure that the log collector is configured as a DNS client for the corporate DNS server. Otherwise, the Log Collector will not know how to find the Kerberos Realm.
Admin Server	(Optional) The name of the Kerberos Administration Server in FQDN format.
Mappings	This parameter is automatically filled after you enter the realm name.

8. Click **Save** to add the Kerberos domain.

To add a Windows Event Source:

1. Go to **(Admin) > Services**.
2. Select a Log Collection service.
3. Under Actions, select **> View > Config** to display the Log Collection configuration parameter tabs.
4. Click the **Event Sources** tab.
5. In the Log Collector **Event Sources** tab, select **Windows/Config** from the drop-down menu.
   The Event Categories panel displays the VMware event sources that are configured, if any.

Next, continue from the current screen to add a Windows Event Category and type.

To configure the Windows Event Type:

1. Select **Windows/Config** from the drop-down menu.
2. In the Event Categories panel toolbar, click  to add a source.

   The Add Source dialog is displayed.
3. Fill in the parameters, using the guidelines below.

Parameter	Details
Alias	The windows domain, referred to as Alias, is the configuration parameter that the Log Collector uses to group event sources. These event source type groups (for example, **domain2**, **domain3**, and **domain4**) categorize the event sources you have configured.
Authorization Method	The authentication method. Valid values are: • **Basic** (default) • **Negotiate** - Negotiates authentication between Kerberos and NTLM (Microsoft Windows NT LAN Manager). For security reasons, NetWitness supports Kerberos exclusively. Select this method.
Channel	For most event sources that use Windows collection, you want to collect from the **Security**, **System**, and **Application** channels. This is a comma-separated list of channels from which NetWitness collects events. The default value for this parameter is: `System, Application, Security` You can use parentheses to include and exclude event IDs. The exclude filter must have a ^ between the channel name and the event ID. You must separate event IDs with a |. For example, **Application^(211|300), System(1010|1012)** excludes the 211 and 300 Application events and includes the 1010 and 1012 System events. A channel is a named stream of events that transports them from an event publisher to an event log file. There are many predefined Windows channels. The following are examples of some of these channels: • **System** - applications that run under system service accounts (installed system services), drivers, or a component or application that has events that relate to the health of the system. • **Application** - all user-level applications. This channel is unsecured and it is open to any application. If an application has extensive information, you should define an application-specific channel for it. • **Security** - the Windows Audit Log (event log) used exclusively for the Windows Local Security Authority.
User Name	Enter the account name for the Windows user account that you set up earlier for communicating with NetWitness. Note that you need to enter the full account name, which includes the domain. For example, `rsalog@DSNETWORKING.COM`. **Note:** For negotiate authentication, this must be the Kerberos principal name in the name@kerberos_domain format. For example, **logcollector@LAB30.LOCAL**.
Password	Enter the correct password for the user account for the event source. The password is encrypted internally and is displayed in its encrypted form.

Parameter	Details
Read All Events	Select this box to read all historical event data from a channel. • **Unchecked** (default) - Log Collector does not collect from all historical event data for a specified channel. • **Checked** - Log Collector collects from all historical event data from a specified channel.
Max Events Per Cycle	(Optional). NetWitness recommends that you set this value to 0, which collects everything.
Polling Interval	(Optional). For most users, a value of **60** should work well.
Render Events	Select this box to request rendered events from the event source. • **Checked** (default) - Log Collector requests rendered events from the event source. • **Unchecked** - Log Collector does not request rendered events from the event source.

4. Click **OK** to add the source.

   The newly added Windows event source is displayed in the Event Categories panel.

5. Select the new event source in the Event Categories panel.

   The **Hosts** panel is activated.

6. Click  in the Hosts panel toolbar.

7. Fill in the parameters, using the guidelines below.

Parameter	Details
Event Source Address	Enter the IP address or host name for the Windows host.
Port	Accept the default value, **5985**.
Transport Mode	Enter **http**.
Enabled	Ensure the box is checked.

8. Click **Test Connection**.

Note: You should be able to successfully test the connection, even if the Windows service is not running.

For more information on any of the previous steps, see the following Help topics in the NetWitness User Guide:

- Configure Windows Collection
- Microsoft WinRM Configuration Guide
- Test and Troubleshoot Microsoft WinRM Guide

Configure RFC5424 Format

The RFC5424 format is supported for all event types. This format is enabled through the 'Enable RFC5424' flag. The default value of this flag is 'False'. It can be set to either 'True' or 'False' to enable or disable RFC5424 respectively.

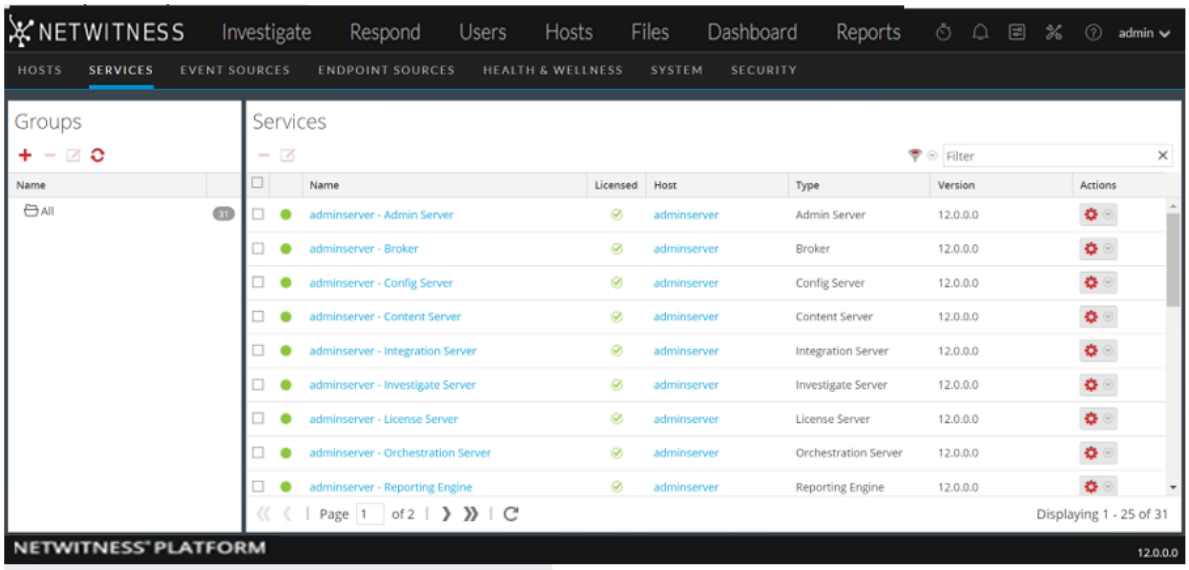
Flag Value	Description
True	Log is sent to Log Decoder in RFC5424 format
False	Log is sent to Log Decoder in Z Connector format

Note: When the log is sent to Log Decoder in RFC 5424 format, the disk utilization for Netwitness database is slightly higher and this may impact the data retention

To enable RFC5424 flag:

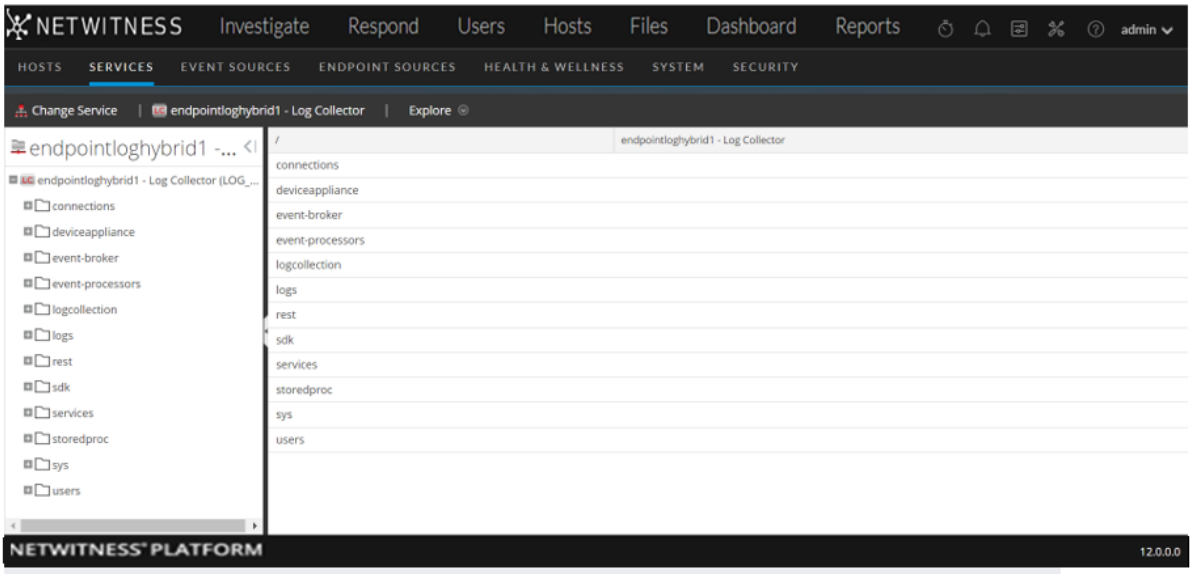
1. Go to **(Admin) > Services** from the NetWitness menu.

   The Services panel displays the list of services.

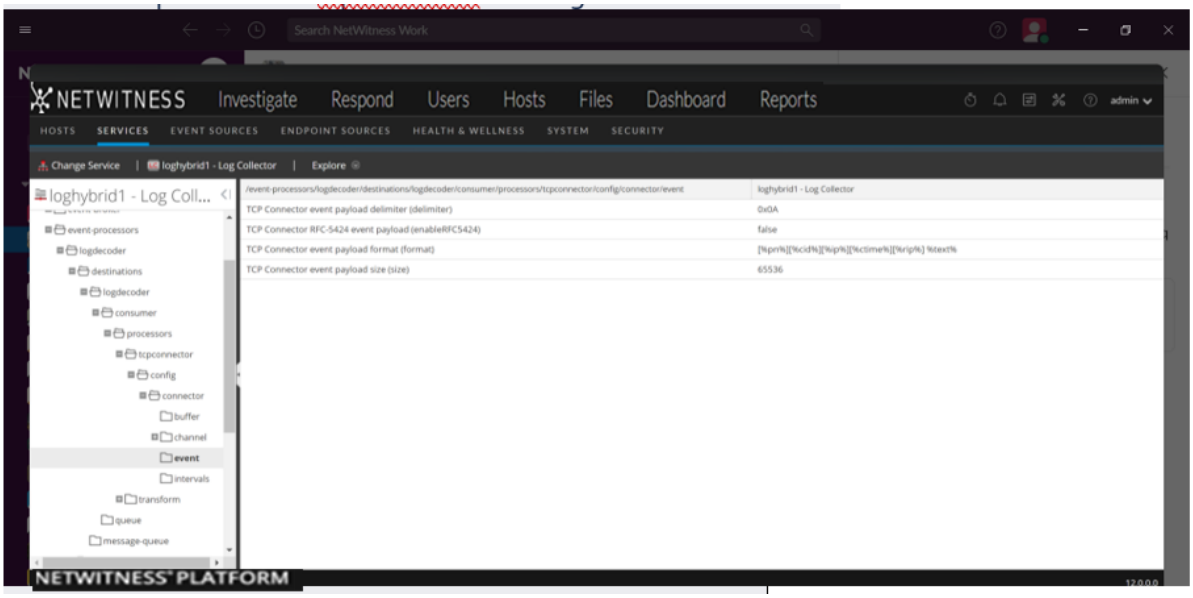


2. Select a Log Collection service.

3. Click > **View** > **Explore**.



4. On the left panel, go to **(event-processors)** > **logdecoder** > **destinations** > **logdecoder** > > > processors > **tcpconnector** > **config** > **connector** > **event**



5. Click **TCP connector RFC-5424 event payload (enable RFC2454)**.
6. Change the value to true to enable the flag.
7. Restart the **Log Collector** service.

Windows Legacy and NetApp Collection Configuration

This **Windows Legacy** protocol collects events from Windows Legacy (Windows 2003 or earlier event sources) and CIFS Auditing events from NetApp ONTAP event sources.

You must deploy Log Collection, that is set up a Local Collector and Windows Legacy Remote Collector, before you can configure the Windows Legacy collection protocol.

How Legacy Windows and NetApp Collection Works

You use the Windows Legacy collection protocol to configure NetWitnessto collection events from:

- Legacy Microsoft Windows event sources (Window 2003 and earlier event sources)
- NetApp event sources

Window 2003 and Earlier Event Sources

Legacy Windows event sources are older Windows versions (such as Windows 2000 and Window 2003). The Windows Legacy collection protocol collects from Windows event sources that are already configured for enVision collection without having to reconfigure them. You set up these event sources under the windows event source type.

NetApp Event Sources

NetApp appliances running Data ONTAP support a native auditing framework that is similar to Windows Servers. When configured, this auditing framework generates and saves audit events in Windows.evt file format. The Windows Legacy collection protocol supports collection of events from such NetApp.evt files. You set up these event sources under the netapp_evt event source type.

The NetApp Data ONTAP appliance is configured to generate CIFS Auditing events and save them periodically as.evt files in a format that includes the timestamp in the filename. Refer to the Network Appliance Data ONTAP Event Source Configuration Guide on NetWitness Link for details. The collection protocol saves the timestamp of the last processed.evt filename to keep track of collection status.

Net App Specific Parameters

Most of the parameters that you maintain in Add/Edit Source dialog apply to both Windows Legacy and Net App events sources.

The following two parameters are unique to NetApp event sources.

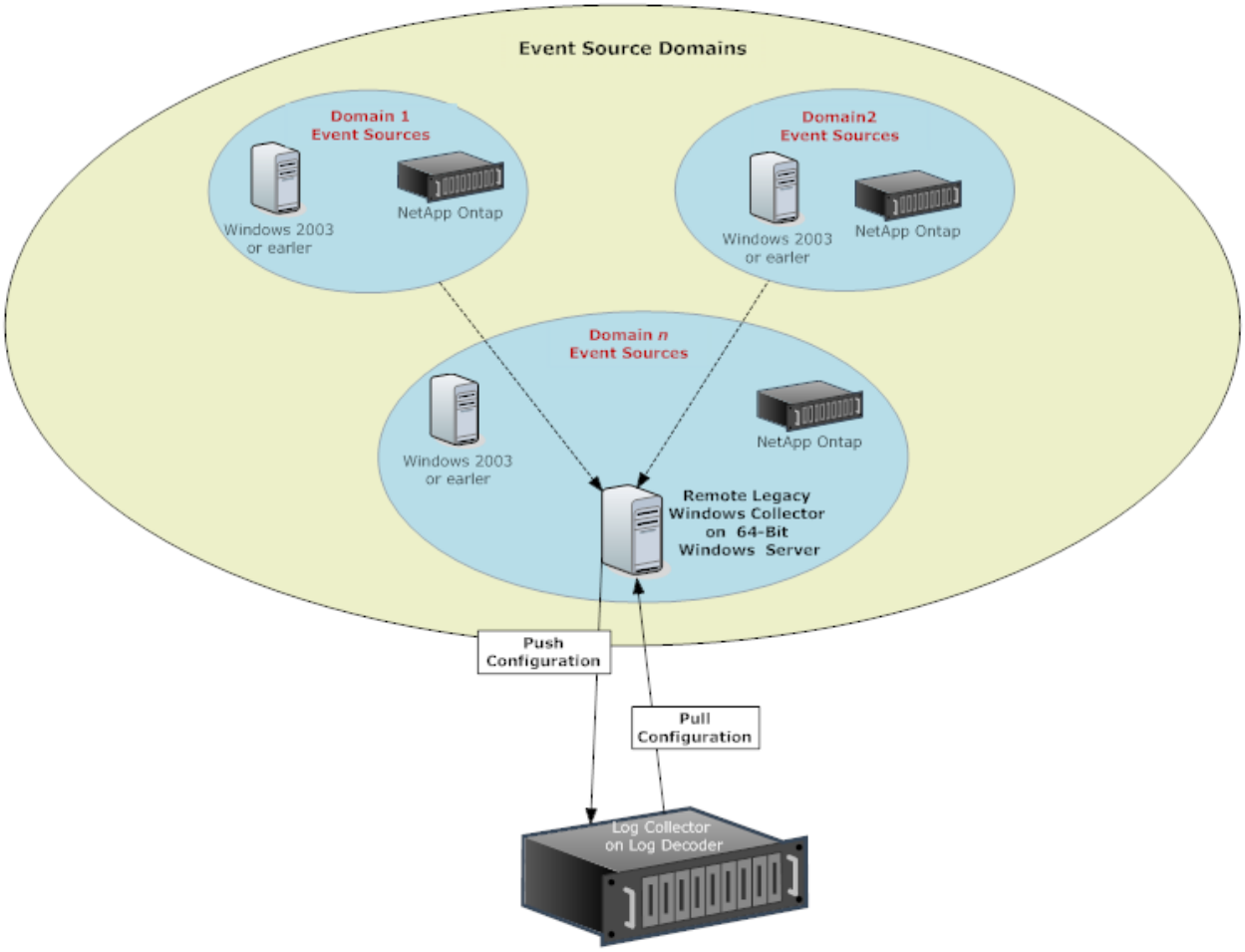
- **Event Directory Path** - The NetApp appliance generates event data and saves it in.evt files in a shareable directory on the NetApp appliance. NetWitness requires you to specify this directory path in the Event Directory Path parameter
- **Event File Prefix** - Similar to the Event Directory Path, NetWitness requires you to specify the prefix (for example, adtlog.) of the event data.evt files so that NetWitness can process this data.

In each polling cycle, NetWitness browses the configured NetApp shared path for the **.evt** files that you identified with the Event Directory Path and Event File Prefix parameters. NetWitness:

- Sorts Files matching the event-file-prefix.YYMMDDhhmmss.evt format in ascending order.
- Uses the timestamp of the last file processed to determine the files that still need processing. If NetWitness finds a partially processed file, it skips the events already processed.

Deployment Scenario

The Windows Legacy collection protocol collects event data from Windows 2003 or earlier, and NetApp ONTAP appliance, event sources. The Windows Legacy Remote Collector is the SA Legacy Windows Collector installed on physical or virtual Windows 2008 64-bit server in your event source domain.



Set Up the Windows Legacy Collector

This topic tells you where to find the executable and instructions required to install or upgrade the Windows Legacy collector in your Windows Legacy domain or domains.

You install the NetWitness Windows Legacy collector on a physical or virtual Windows 2008 R2 SP1 64-Bit server using the **NWLegacyWindowsCollector-11.*version-number.exe***. You download the **NWLegacyWindowsCollector-11.*version-number.exe*** from NetWitness Link. Please refer to the *NetWitness 11.x Windows Legacy Collection Upgrade & Installation Instructions* for the details on how to install or upgrade Windows Legacy collection.

Note: The Microsoft Management Console (MMC) should be closed during the installation process.

Configure Windows Legacy and NetApp Event Sources

This topic tells you how to configure Windows Legacy event sources in NetWitness.

The Windows Legacy collection protocol collects event data from Windows 2003 or earlier event sources, and from NetApp event sources.

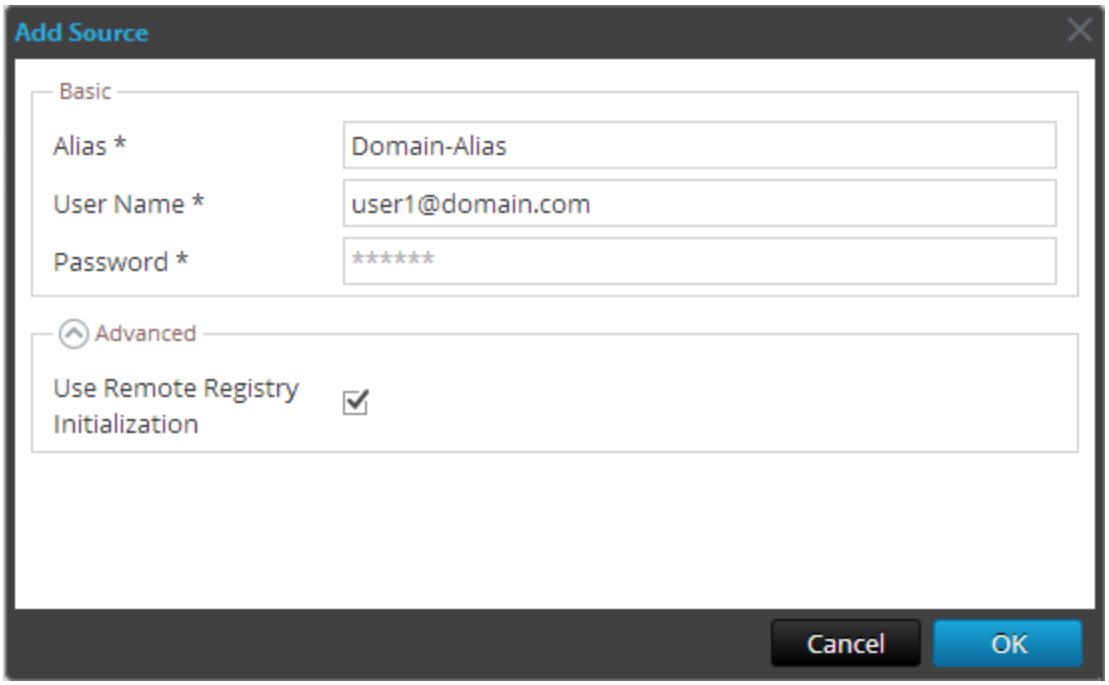
Prerequisites

Before you configure a Windows Legacy event source, make sure that you have:

1. Installed the NetWitness Windows Legacy Remote Collector on a physical or virtual Windows 2008 64-bit server.
2. Added this Windows Legacy Remote Collector to NetWitness.

Add a Windows Legacy Event Source

1. Access the Services view by selecting  **(Admin) > Services** from the NetWitness menu.
2. In the **Services** grid, select a **Windows Legacy Log Decoder** service.
3. Under Actions, select  **> View > Config** to display the Log Collection configuration parameter tabs.
4. Click the **Event Sources** tab.
5. In the **Event Sources** tab, select one of the following options from the drop-down menu.
   - Windows Legacy/Windows.
   - Windows Legacy/NetApp.
6. Configure the alias:
   a. Click  in the **Event Categories** panel toolbar.
   The **Add Source** dialog is displayed.
   b. Specify values for the parameters and click **OK**.



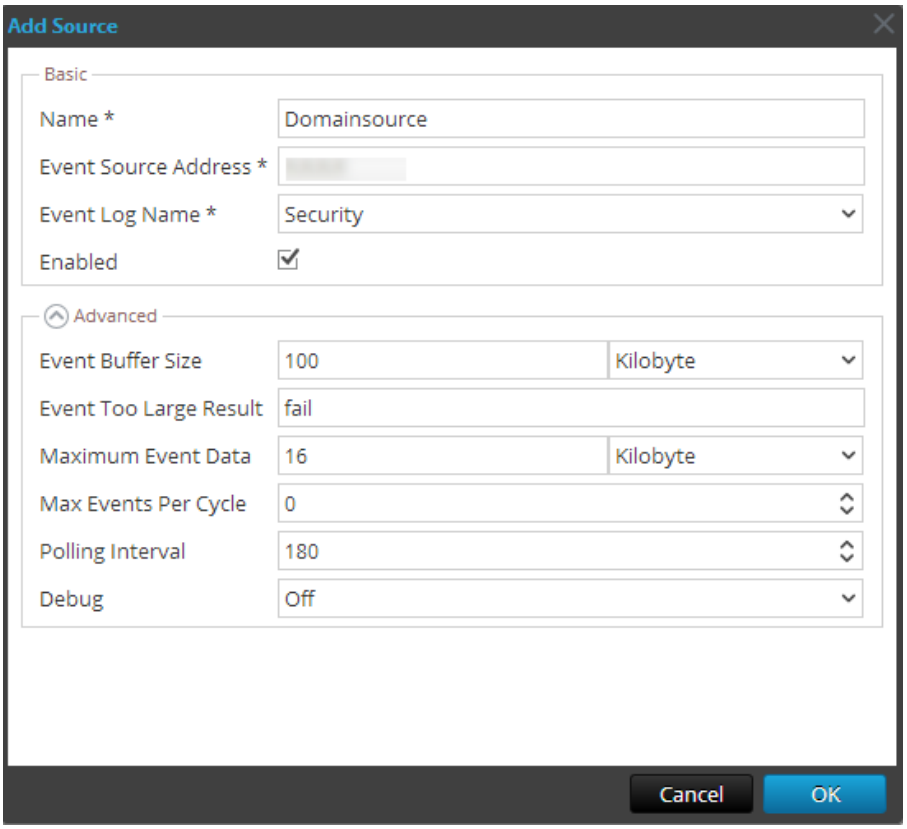
Note: By default, **Remote Registry Initialization** is selected. For details, see Remote Registry Access below.

The newly added windows event source type is displayed in the **Event Categories** panel.

7. Add the event source:

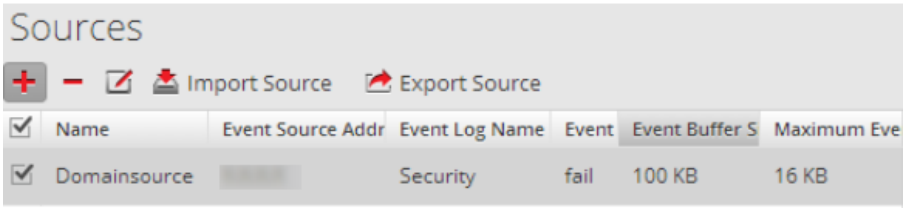
   a. Select the new alias in the **Event Categories** panel and click + in the **Source** panel toolbar. The **Add Source** dialog is displayed.

   b. Specify values for the event source parameters and click **OK**.



For details, see Windows Legacy Configuration Parameters below.

The newly added Windows event source is displayed in the **Event Categories** panel.



Remote Registry Access

Windows Legacy Collector performs an initial verification of the event source before collecting data. By default, Windows Legacy Collector uses Windows Management Instrumentation (WMI) method to perform this initial verification. If you enable Remote registry access method, Windows Legacy Collector performs a remote registry query to verify the event source.

Configure Push or Pull between Log Collector and Windows Legacy Collector

You can configure the Windows Legacy Collector to push event data to a Local Collector, or you can configure a Local Collector to pull event data from the Windows Legacy Collector.

To configure a Local Collector or the Windows Legacy Collector:

1. Go to **(Admin) > Services**.
2. Select a Local Collector or the Windows Legacy Collection service.
3. Under Actions, select **> View > Config** to display the Log Collection configuration parameter tabs.
4. Depending on your selection in step 2:
   - If you selected a Local Collector, the **Remote Collectors** tab is displayed. Select the Windows Legacy Collector from which the Local Collector pulls events in this tab.
   - If you selected a Windows Legacy Collector, the **Local Collectors** are displayed. Select the Local Collectors to which the Windows Legacy Collector pushes events in this tab.

Windows Legacy Configuration Parameters

The following table describes the basic parameters for a Windows Legacy event source.

Note: Required parameters are marked with an asterisk. All other parameters are optional.

Feature	Description
Name*	The name of the event source. Valid value is a name in the **[_a-zA-Z] [_a-zA-Z0-9]*** range. You can use a dash "**-**" as part of the name.
Event Source Address*	IP address of the event source. Valid value is an IPv4 address, IPv6 address, or a hostname including a fully qualified domain name. NetWitness defaults to **127.0.0.1**. Log Collector converts the hostname to lower-case letters to prevent duplicate entries.
Event Log Name	The name of the event log from which to collect event data (for example, **System**, **Application**, or **Security**). The following are examples of some of these channels: • **System** - applications that run under system service accounts (installed system services), drivers, or a component or application that has events that relate to the health of the system. • **Application** - all user-level applications. This channel is unsecured and it is open to any application. If an application has extensive information, you should define an application-specific channel for it. • **Security** - the Windows Audit Log (event log) used exclusively for the Windows Local Security Authority.
Enabled	Select this checkbox to collect from this event source. If you do not check this checkbox, the Log Collector does not collect events from this event source.

Feature	Description
Event Directory Path	NetApp **.evt or.evtx** files directory path. This must be the UNC path. The NetApp generates event data and saves it in.evt or.evtx files in a shareable directory on the NetApp appliance. • In each polling cycle, Log Collector browses the configured NetApp shared path for the.evt files that you identified with the **Event Directory Path** and **Event File Prefix parameters**. Log Collector : ◦ sorts files that match the **event-file-prefix.YYMMDDhhmmss.evt** format in ascending order. ◦ uses the timestamp of the last file processed to determine the files that still need processing. If Log Collector finds a partially processed file, it skips the events already processed. • In each polling cycle, Log Collector browses the configured NetApp shared path for the **.evtx** files that you identified with the **Event Directory Path** and **Event File Prefix parameters**. Log Collector : ◦ sorts files that match the **event-file-prefix.YYMMDDhhmmssms.evtx** format in ascending order. ◦ uses the timestamp of the last file processed to determine the files that still need processing. If Log Collector finds a partially processed file, it skips the events already processed.
Event File Prefix	Prefix of the **.evt** files (for example, **adtlog**.) saved in the **Event Directory Path**.
Cancel	Closes the dialog without adding the Windows Legacy event source.
OK	Adds the current parameter values as a new event source

The following table describes the advanced parameters for a Windows Legacy event source.

Feature	Description
Event Buffer Size	Maximum size of the data the Log Collector pulls from the event source for each request. Valid value is a number in **0** to **511** Kilobytes range. You specify this value in **Kilobytes**.
Event Too Large Result	Tells Log Collector what to do if an event is too large for the event buffer.
Maximum Event Data	Maximum size of event data to include in the output. Valid value is a number in **0** to **511Kilobytes** range. You specify this value in **Kilobytes** or **Megabytes**. • 1 Kilobyte - 100 Megabytes • 0 = do not include event data in the output.
Max Events Per Cycle	The maximum number of events per polling cycle (how many events collected per polling cycle).
Polling Interval	Interval (amount of time in seconds) between each poll. The default value is **180**. For example, if you specify 180, the collector schedules a polling of the event source every 180 seconds. If the previous polling cycle is still underway, it will wait for it to finish that cycle. If you have a large number of event sources that you are polling, it may take longer than 180 seconds for the polling to start because the threads are busy.

Feature	Description
Debug	**Caution:** Only enable debugging (set this parameter to On or Verbose) if you have a problem with an event source and you need to investigate this problem. Enabling debugging will adversely affect the performance of the Log Collector. Enables or disables debug logging for the event source. Valid values are: • **Off** = (default) disabled • **On** = enabled • **Verbose** = enabled in verbose mode - adds thread information and source context information to the messages. This parameter is designed to debug and monitor isolated event source collection issues. If you change this value, the change takes effect immediately (no restart required). Limit the number of event sources for which you use Verbose debugging to minimize performance impact.

Troubleshoot Windows Legacy and NetApp Collection

This topic highlights possible problems that you may encounter with Windows Legacy Collection (LWC) and suggested solutions to these problems.

Note: In general, you receive more robust log messages by disabling SSL.

Protocol Restart Problems

Problem	Possible Causes	Solutions
You restart the Legacy Windows collection protocol, but NetWitness is not receiving events.	The logcollector service is stopped.	Restart the **logcollector** service. 1. Log on to the **Windows Legacy Remote Collector**. 2. Go to **Start > Administrative Tools > Task Scheduler** and click on **Task Scheduler Library**. 3. In the right panel, look for the **restartnwlogcollector** task and make sure that it is running. 4. If this is not the case, right-click **restartnwlogcollector** and select **Run**.

Installation Problems

If you see any of the following messages in the **MessageBroker.log**, you may have issues.

Log Messages	Any message that contains "rabbitmq"
Possible Cause	RabbitMQ service may not be running. Port **5671** may not be opened.
Solutions	Make sure that the RabbitMQ service is running. Make sure that port **5671** is open.
Log Messages	Error: Adding logcollector user account. Error: Adding administrator tag to logcollector account. Error: Adding logcollection vhost. Error: Setting permissions to logcollector account in all vhosts.
Possible Cause	**rabbitmq-server** was not running when installer tried to create users and vhosts.

Solutions	Make sure that the **RabbitMQ** service is running and run below commands manually.
	`rabbitmqctl -q add_user logcollector netwitness`
	`rabbitmqctl -q set_user_tags logcollector administrator`
	`rabbitmqctl -q add_vhost logcollection`
	`rabbitmqctl -q set_permissions -p / logcollector ".*" ".*" ".*"`
	`rabbitmqctl -q set_permissions -p logcollection logcollector ".*" ".*" ".*"`

Windows Legacy Federation Script Issues

If you see any of the following messages in the federation script log, you may have issues.

Problem	Possible Symptoms	Solutions
Federation script started, but the LWC service went down.	NetWitness log shows connection failure exceptions with Windows Legacy Collector.	This issue is fixed automatically after restarting the Windows Legacy service.

Problem	Possible Symptoms	Solutions

LWC is running, but RabbitMQ service is down or restarting.	Federation log file at Windows Legacy side displays an error message about RabbitMQ service being down.
	The log file to look at is: **C:\NetWitness\ng\logcollector**
	The following error message is logged in case RabbitMQ is not running:
	`"Unable to connect to node logcollector@localhost: nodedown"`
	The following diagnostics messages are displayed:
	`attempted to contact: [logcollector@localhost]`
	`logcollector@localhost: * connected to epmd (port 4369) on localhost * epmd reports: node 'logcollector' not running at all other nodes on localhost: ['rabbitmqctl-4084'] * suggestion: start the node`
		Run the **federation.bat** script manually at LWC. To run the **federate.bat** script manually, perform the following steps:
		1. Go to folder C:**\Program Files\NwLogCollector** where the Windows Legacy instance is installed.
		2. Locate the file **federate.bat i**n this folder. Select the file and right click.
		3. Select **Run as Administrator**.
		4. To monitor the log file, navigate to **C:\NetWitness\ng\logcollector\federate.log** while the **federate.bat** script is being executed.
		Note: Make sure the log file does not show any errors while the script is being executed.

RabbitMQ service is down on the NetWitness side.	NetWitness User Interface pages do not work.	Restart RabbitMQ service.
Customer receives a Health and Wellness notification, or the following Health and Wellness Alarm is displayed: "Communication failure between Master NetWitness Host and a Remote Host" with LWC Host as the Remote IP.	**Federate.bat** script failed to run successfully.	If the **Federate.bat** script did not run correctly, run it manually as described previously.

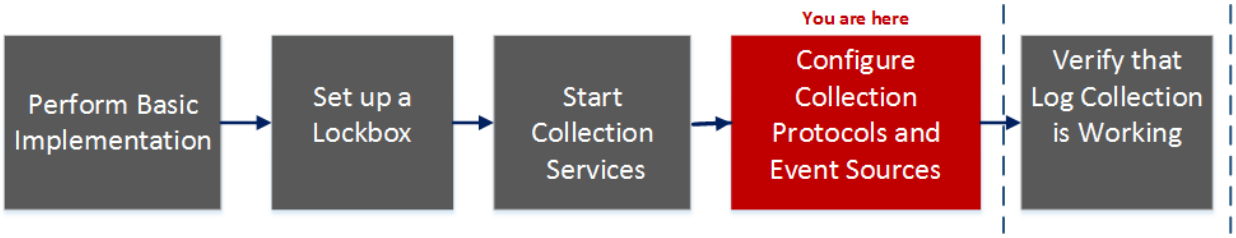
Reference

AWS Parameters

This topic provides an overview of the AWS collection configuration parameters for deploying a remote log collection service (VLC) in an Amazon Web Services (AWS) environment.

Workflow

This workflow illustrates the basic tasks needed to start collecting events through Log Collection.



What do you want to do?

Role	I Want to...	Documentation
Administrator	Perform basic Log Collection implementation	Basic Implementation
Administrator	Set up a lockbox to maintain lockbox settings.	Set Up a Lockbox
Administrator	Start Log Collection services.	Start Collection Services
Administrator	*Configure Log Collection protocols and event sources.	Configure Collection Protocols and Event Sources
Administrator	Verify that Log Collection is working.	Verify That Log Collection Is Working

*You can perform this task here.

Related Topics

- Configure AWS (CloudTrail) Event Sources in NetWitness

The following table describes the **Basic** configuration parameter for AWS collection.

Note: Required parameters are marked with an asterisk. All other parameters are optional.

Parameter	Description
Name *	Name of the event source.
Enabled	Select the check box to enable the event source configuration to start collection. The check box is selected by default.
Account Id *	Account Identification code of the S3 Bucket

Parameter	Description
S3 Bucket Name *	Name of the AWS (CloudTrail) S3 bucket. Amazon S3 bucket names are globally unique, regardless of the AWS (CloudTrail) region in which you create the bucket. You specify the name at the time you create the bucket. Bucket names should comply with DNS naming conventions. The rules for DNS-compliant bucket names are: • Bucket names must be at least three and no more than 63 characters long. • Bucket names must be a series of one or more labels. Adjacent labels are separated by a single period ".". Bucket names can contain lowercase letters, numbers, and hyphens. Each label must start and end with a lowercase letter or a number. • Bucket names must not be formatted as an IP address (for example, 192.168.5.4). The following examples are **valid** bucket names: • `myawsbucket` • `my.aws.bucket` • `myawsbucket.1` The following examples are **invalid** bucket names: • `.myawsbucket` - Do not start a Bucket Name with a period • `myawsbucket.` - Do not end a Bucket Name with a period • `my..examplebucket` - Only use one period between labels.
Access Key *	Key used to access the S3 bucket. Access Keys are used to make secure REST or Query protocol requests to any AWS service API. Please refer to Manage User Credentials on the Amazon Web Services support site for more information on Access Keys.
Secret Key *	Secret key used to access the S3 bucket.
Region *	Region of the S3 bucket. `us-east-1` is the default value.
Region Endpoint	Specifies the AWS CloudTrail hostname. For example, for an AWS public cloud for us-east region, the Region Endpoint would be s3.amazonaws.com. More information can be found at http://docs.aws.amazon.com/general/latest/gr/rande.html#s3_region. This parameter is necessary to collect CloudTrail logs from AWS Government or Private clouds.
Use Proxy	Enable **Use Proxy** to set proxy for AWS server. By default, it is disabled.
Proxy Server	Enter the proxy name you want to connect to access the AWS server.
Proxy Port	Enter the port number that connects to the proxy server to access the AWS server.
Proxy User	Enter the user name to authenticate with the proxy server.
Proxy Password	Enter the password to authenticate with proxy port.
Start Date *	Starts AWS (CloudTrail) collection from the specified number of days in the past, measured from the current timestamp. The default value is 0, which starts from today. The range is 0–89 days.
Log File Prefix	Prefix of the files to be processed. **Note:** If you set a prefix when you set up your CloudTrail service, make sure to enter the same prefix in this parameter.

Parameter	Description
Cancel	Closes the dialog without adding the AWS (CloudTrail).
OK	Adds the current parameter values as a new AWS (CloudTrail).

The following table describes the **Advanced** configuration parameter for AWS collection.

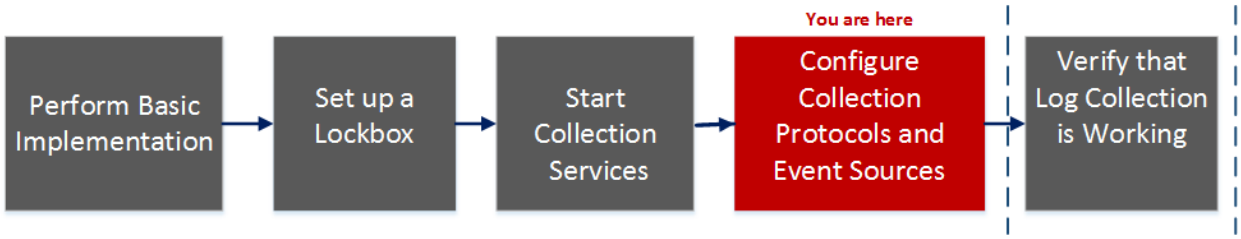
Parameter	Description
Debug	**Caution:** Only enable debugging (set this parameter to On or Verbose) if you have a problem with an event source and you need to investigate this problem. Enabling debugging will adversely affect the performance of the Log Collector. Enables or disables debug logging for the event source. Valid values are: • **Off** = (default) disabled **On** = enabled • **Verbose** = enabled in verbose mode - adds thread information and source context information to the messages. This parameter is designed to debug and monitor isolated event source collection issues. The debug logging is verbose, so limit the number of event sources to minimize performance impact. If you change this value, the change takes effect immediately (no restart required).
Command Args	Arguments added to the script.
Polling Interval	Interval (amount of time in seconds) between each poll. The default value is 60. For example, if you specify 60, the collector schedules a polling of the event source every 60 seconds. If the previous polling cycle is still underway, it will wait for it to finish that cycle. If you have a large number of event sources that you are polling, it may take longer than 60 seconds for the polling to start because the threads are busy.
SSL Enabled 	Select the check box to communicate using SSL. The security of data transmission is managed by encrypting information and providing authentication with SSL certificates. The check box is selected by default.
Test Connection	Validates the configuration parameters specified in this dialog are correct. For example, this test validates that: • NetWitness can connect with the S3 Bucket in AWS using the credentials specified in this dialog. • NetWitness can download a log file from the bucket (test connection would fail if there were no log files for the entire bucket, but this would be extremely unlikely).

Azure Parameters

Microsoft Azure is a cloud computing platform and infrastructure for building, deploying, and managing applications and services through a global network of Microsoft-managed data centers.

Workflow

This workflow illustrates the basic tasks needed to start collecting events through Log Collection.



What do you want to do?

Role	I Want to...	Documentation
Administrator	Perform basic Log Collection implementation	Basic Implementation
Administrator	Set up a lockbox to maintain lockbox settings.	Set Up a Lockbox
Administrator	Start Log Collection services.	Start Collection Services
Administrator	*Configure Log Collection protocols and event sources.	Configure Collection Protocols and Event Sources
Administrator	Verify that Log Collection is working.	Verify That Log Collection Is Working

*You can perform this task here.

Related Topics

- Configure Azure Event Sources in NetWitness

Azure Event Source Configuration Parameters

This topic describes the Azure event source configuration parameters.

Basic Parameters

Note: Required parameters are marked with an asterisk. All other parameters are optional.

Name	Description
Name *	Enter an alpha-numeric, descriptive name for the source. This value is only used for displaying the name on this screen.

Name	Description
Enabled	Select the checkbox to enable the event source configuration to start collection. The checkbox is selected by default.
Client ID *	The Client ID is found the Azure Application Configure tab. Scroll down until you see it.
Client Secret *	When you are configuring the event source, the client secret is displayed when you are creating a key, and you select a duration of validation. Make sure to save this, because you will only be able to see it once, and it cannot be retrieved later.
API Resource Base URL *	Enter `https://management.azure.com/`. Be sure to include the trailing slash (/).
Federation Metadata Endpoint *	In your Azure application, click the **View Endpoints** button (near the bottom of the pane). There are a lot of links that all begin with the same string. Compare the URLs and find the common string that begins most of them. This common string is the endpoint that you need to enter here.
Subscription ID *	You can find this in the Microsoft Azure dashboard: click on Subscriptions at the bottom of the list on the left.
Tenant Domain *	Go to the active directory and click on the directory. In the URL, the tenant domain is the string directly following **manage.windowsazure.com/**. The tenant domain is the string up to and including the **.com**.
Resource Group Names *	In Azure, select Resource groups from the left navigation pane, then select your group.
Start Date *	Choose the date from which to start collecting. Default's to the current date.
Test Connection	Checks the configuration parameters specified in this dialog to make sure they are correct.

Advanced Parameters

Click next to **Advanced** to view and edit the advanced parameters, if necessary.

Name	Description
Polling Interval	Interval (amount of time in seconds) between each poll. The default value is **180**. For example, if you specify 180, the collector schedules a polling of the event source every 180 seconds. If the previous polling cycle is still underway, the collector waits for that cycle to finish. If you have a large number of event sources that you are polling, it may take longer than 180 seconds for the polling to start because the threads are busy.
Max Duration Poll	Maximum duration, in seconds, of a polling cycle. A zero value indicates no limit.

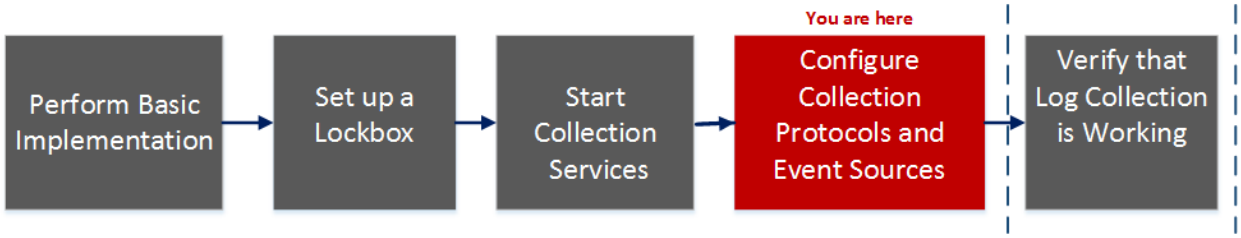
Name	Description
Max Events Poll	The maximum number of events per polling cycle (how many events collected per polling cycle).
Max Idle Time Poll	Maximum duration, in seconds, of a polling cycle. A zero value indicates no limit.
Command Args	Optional arguments to be added to the script invocation.
Debug	**Caution:** Only enable debugging (set this parameter to On or Verbose) if you have a problem with an event source and you need to investigate this problem. Enabling debugging will adversely affect the performance of the Log Collector. Enables or disables debug logging for the event source. Valid values are: • **Off** = (default) disabled • **On** = enabled • **Verbose** = enabled in verbose mode - adds thread information and source context information to the messages. This parameter is designed to debug and monitor isolated event source collection issues. If you change this value, the change takes effect immediately (no restart required). The debug logging is verbose, so limit the number of event sources to minimize performance impact.

Check Point Parameters

The Check Point Collection protocol collects events from Check Point event sources using OPSEC LEA. OPSEC LEA is the Check Point Operations Security Log Export API that facilitates the extraction of logs.

Workflow

This workflow illustrates the basic tasks needed to start collecting events through Log Collection.



What do you want to do?

Role	I Want to...	Documentation
Administrator	Perform basic Log Collection implementation	Basic Implementation
Administrator	Set up a lockbox to maintain lockbox settings.	Set Up a Lockbox
Administrator	Start Log Collection services.	Start Collection Services
Administrator	*Configure Log Collection protocols and event sources.	Configure Collection Protocols and Event Sources
Administrator	Verify that Log Collection is working.	Verify That Log Collection Is Working

*You can perform this task here.

Related Topics

- Configure Check Point Event Sources in NetWitness

Check Point Collection Configuration Parameters

Note: Required parameters are marked with an asterisk. All other parameters are optional.

Basic Parameters

Parameter	Description
Name*	Name of the event source.
Address*	IP Address of the Check Point server.

Parameter	Description
Server Name*	Name of the Check Point server.
Certificate Name	Certificate name for secure connections to use when the transport mode is https. If set, the certificate must exist in the certificate trust store that you created using the Settings tab. Select a certificate from the drop-down list. The file naming convention for Check Point event source certificates is **checkpoint_*name-of-event-source***.
Client Distinguished	Enter the Client Distinguished Name from the Check Point server.
Client Entity Name	Enter the Client Entity Name from the Check Point server.
Server Distinguished	Enter the Server Distinguished Name from the Check Point server.
Enabled	Select the check box to enable the event source configuration to start collection. The check box is selected by default.
Pull Certificate	Select the checkbox to pull a certificate for first time. Pulling a certificate makes it available from the trust store.
Certificate Server Address	IP Address of the server on which the certificate resides. Defaults to the event source address.
Password	Only active when you select the Pull Certificate checkbox for first time. Password required to pull the certificate. The password is the activation key created when adding an OPSEC application to Check Point on the Check Point server.

Determine Advanced Parameter Values for Check Point Collection

You use less system resources when you configure a Check Point event source connection to stay open for a specific time and specific event volume (transient connection). NetWitness defaults to the following connection parameters that establish a transient connection:

- Polling Interval = **180** (3 minutes)
- Max Duration Poll = **120** (2 minutes)
- Max Events Poll = **5000** (5000 events per polling interval)
- Max Idle Time Poll = **0**

For very active Check Point event sources, it is a good practice to set up a connection that stays open until you stop collection (persistent connection). This ensures that Check Point collection maintains the pace of the events generated by these active event sources. The persistent connection avoids restart and connection delays and prevents Check Point collection from lagging behind event generation.

To establish a persistent connection for a Check Point event source, set the following parameters to the following values:

- Polling Interval = **-1**
- Max Duration Poll = **0**
- Max Events Poll = **0**
- Max Idle Time Poll = **0**

Parameter	Description
Port	Port on the Check Point server that Log Collector connects to. Default value is 18184.
Collect Log Type	Type of logs that you want to collect: Valid values are: • **Audit** - collects audit events. • **Security** - collects security events. If you want to collect both audit and security events, you must create a duplicate event source. For example, first you would create an event source with Audit selected pulling a certificate into the trust store for this event source. Next you would create another event source with the same values except that you would select Security for the Collect Log Type and you would select the same certificate in Certificate Name that you pulled when you set up the first set of parameters for this event source and you would make sure that Pull Certificate was not selected.
Collect Logs From	When you set up a Check Point event source, NetWitness collects events from the current log file. Valid values are: • **Now** - Start collecting logs now (at this point in time in the current log file). • **Start of Log** - Collect logs from the beginning of the current log file. If you choose "Start of Log" for this parameter value, you may collect a very large amount of data depending on how long the current log file has been collecting events. Note that this option is effective only for the first collection session.
Polling Interval	Interval (amount of time in seconds) between each poll. The default value is **180**. For example, if you specify 180, the collector schedules a polling of the event source every 180 seconds. If the previous polling cycle is still underway, it will wait for it to finish that cycle. If you have a large number of event sources that you are polling, it may take longer than 180 seconds for the polling to start because the threads are busy.
Max Duration Poll	The maximum duration of polling cycle (how long the cycle lasts) in seconds.
Max Events Poll	The maximum number of events per polling cycle (how many events collected per polling cycle).
Max Idle Time Poll	Maximum idle time, in seconds, of a polling cycle. 0 indicates no limit.**> 300** is the default value.
Forwarder	Enables or disables the Check Point server as a forwarder. By default it is disabled.

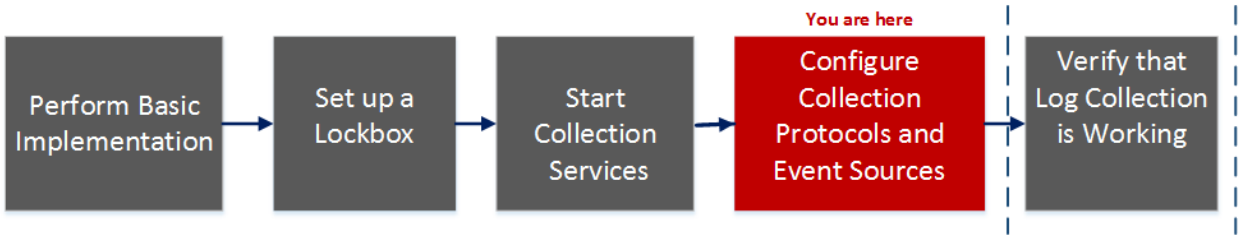
Parameter	Description
Log Type (Name Value Pair)	Logs from the event source in Name Value format. By default it is disabled.
Debug	**Caution:** Only enable debugging (set this parameter to "On" or "Verbose") if you have a problem with an event source and you need to investigate this problem. Enabling debugging will adversely affect the performance of the Log Collector. Enables and disables debug logging for the event source. Valid values are: • **Off** = (default) disabled • **On** = enabled • **Verbose** = enabled in verbose mode - adds thread information and source context information to the messages. This parameter is designed to debug and monitor isolated event source collection issues. The debug logging is verbose, so limit the number of event sources to minimize performance impact. If you change this value, the change takes effect immediately (no restart required).

File Parameters

This topic describes the File Collection configuration parameters.

Workflow

This workflow illustrates the basic tasks needed to start collecting events through Log Collection.



What do you want to do?

Role	I Want to...	Documentation
Administrator	Perform basic Log Collection implementation	Basic Implementation
Administrator	Set up a lockbox to maintain lockbox settings.	Set Up a Lockbox
Administrator	Start Log Collection services.	Start Collection Services
Administrator	*Configure Log Collection protocols and event sources	Configure Collection Protocols and Event Sources
Administrator	Verify that Log Collection is working.	Verify That Log Collection Is Working

*You can perform this task here.

Related Topics

- Configure File Event Sources in NetWitness

File Collection Event Source Parameters

The following table provides descriptions of the File Collection source parameters.

The following table describes the **Basic** configuration parameter for File collection.

Note: Required parameters are marked with an asterisk. All other parameters are optional.

Name	Description

Name	Description
File Directory*	Collection directory (for example, **Eur_London100**) into which the File event source places its files. Valid value is a character string that is conforms to the following regular expression: **[_a-zA-Z][_a-zA-Z0-9]*** This means that the file directory must start with a letter followed by numbers, letters, and underscores. Do not modify this parameter after you start collecting event data. After you create the collection, the Log Collector creates the work, save, and error sub-directories under the collection directory.
Address*	IP address of the event source. Valid value is an **IPv4 address**, **IPv6 address**, or a **hostname** including a fully-qualified domain name.
File Spec	Regular expression. For example, **^.*$** = process everything.
File Encoding	Character encoding used by the syslog senders to this port. Defaults to UTF-8. **Note:** It is safe to leave this as UTF-8, since UTF-8 handles ASCII characters as well, and most senders have their encoding set to UTF-8. NetWitness has tested the following values: • EUC-KR • SJIS • GB3212/GBK • ISO_8859-1 (German) • ISO_8859-7 (Greek)
Cancel	Closes the dialog without making adding an event source type.
OK	Adds the parameters for the event source.

The following table describes the **Advanced** configuration parameter for File collection.

Name	Description
Ignore Encoding Conversion Errors	Select the check box to ignore encoding conversion errors and ignore invalid data. The check box is selected by default. **Caution:** This may cause parsing and transformation errors.

Name	Description
File Disk Quota	Determines when to stop saving files regardless of the **Save On Error** and **Save On Success** parameter settings. For example, a value of 10 indicates that when there is less than 10% available disk left, the Log Collector stops saving files to reserve enough space for your estimated normal collection processing. **Caution:** Available disk refers to a partition where the base collection directory is mounted. If the Log Decoder server has a 10TB disk size and 2TB is allocated to base collection directory, then setting this value to 10 causes log collection to stop when less than 0.2TB (10% of 2TB) of space is left. It does not mean 10% of 10TB. Valid value is a number in the **0** to **100** range. **10** is the default.
Sequential Processing	Sequential processing flag: • Select the check box (default) to process event source files in collection order. • Do not select the checkbox to process event source files in parallel.
Save On Error	Save on error flag. Check the checkbox to retain the **eventsource collection** file when the Log Collector it encounters an error. The check box is selected by default.
Save On Success	Save **eventsource collection** file after processing flag. Check the checkbox to save the eventsource collection file after processing it. The check box is not selected by default.
Eventsource SSH Key	SSH public key used to upload files for this event source. Please refer to the *Generate Key Pair on Event Source and Import Public Key to Log Collector* section in the Install and Update the SFTP Agent Guide for instructions on generating keys. **Note:** If File collection is stopped, NetWitness does not update the authorized_keys file with the SSH public key that you add or modify in this parameter. You must restart File collection to update the public key. You can add or modify the value of the public key in this parameter in multiple File event sources without File collection running, but NetWitness will not update the **authorized_keys** file until File collection is restarted.
Manage Error Files	By default, the Log Collector uses the **File Disk Quota** parameter to ensure that the disk does not fill up with error files. If you set this parameter to **true**, you can specify one of these: • Maximum space allotted to error files in the **Error Files Size** parameter. • Maximum number of error files allowed in **Error Files Count** parameter. A reduction percent is also specified, which tells the system how much to reduce when the maximum is reached. Select the check box to manage error files. The check box is not selected by default.

Name	Description
Error Files Size	Only valid if the **Manage Error Files** and **Save On Error** parameters are set to true. Specifies to what extent NetWitness saves error files. The value that you specify is the maximum total size of all the files in the error directory. Valid value is a number in **0** to **281474976710655** range. You specify these values in either **Kilobytes**, **Megabytes**, or **Gigabytes**. **100 Megabytes** is the default. If you change this parameter, the change does not take effect until you restart collection or restart the Log Collector service.
Error Files Count	Only valid if the **Manage Error Files** and **Save On Error** parameters are set to true. Maximum number of error files allowed in the error directory. Valid value is a number in **0** to **65536** range. **65536** is the default. If you change this parameter, the change does not take effect until you restart collection or restart the Log Collector service.
Error Files Reduction %	Percent amount by size or count of the error files that the Log Collector service removes when the maximum size or count has been reached. The service removes the oldest files first. Valid value is a number in the **0** to **100** range. **10** is the default.
Manage Saved Files	Select the check box to manage saved files. The check box is not selected by default. By default, the Log Collector uses the **File Disk Quota** parameter to ensure that the disk does not fill up with saved files. If check this check box, you can specify one of these: • Maximum space allotted to saved files in the **Saved Files Size** parameter. • Maximum number of saved files allowed in **Saved Files Count** parameter. A reduction percent is also specified, which tells the system how much to reduce when the maximum is reached.
Saved Files Size	Only valid if the **Manage Saved Files** and **Save On Success** parameters are set to true. Maximum total size of all the files in the save directory. Valid value is a number in the **0** to **281474976710655** range. You specify these values in either **Kilobytes**, **Megabytes**, or **Gigabytes**. **100 Megabytes** is the default. If you change this parameter, the change does not take effect until you restart collection or restart the Log Collector service.
Saved Files Count	Only valid if the **Manage Saved Files** and **Save On Success** parameters are set to true. Maximum number of saved files in the save directory. Valid value is a number in **0** to **65536** range. **65536** is the default. If you change this parameter, the change does not take effect until you restart collection or restart the Log Collector service.
Saved File Reduction %	Percent amount by size or count of the saved files that the Log Collector service removes when the maximum size or count has been reached. The service removes the oldest files first. Valid value is a number in the **0** to **100** range. **10** is the default.

Name	Description
Debug	**Caution:** Only enable debugging (set this parameter to **On** or **Verbose**) if you have a problem with an event source and you need to investigate this problem. Enabling debugging will adversely affect the performance of the Log Collector. Enables or disables debug logging for the event source. Valid values are: • **Off** = (default) disabled • **On** = enabled • **Verbose** = enabled in verbose mode - adds thread information and source context information to the messages. This parameter is designed to debug and monitor isolated event source collection issues. The debug logging is verbose, so limit the number of event sources to minimize performance impact. If you change this value, the change takes effect immediately (no restart required).

Log Collection Service System View

A Log Collector is a service that runs on a Log Decoder host (referred to as a Local Collector) or sends events from a Remote Collector to a Local Collector, and is configured and managed in a similar way to a Log Decoder.

To access the Log Collection Service System view, go to (Admin) > Services and select a Log Collector service, then select View > System.

Workflow

This workflow illustrates the basic tasks needed to start collecting events through Log Collection.

What do you want to do?

Role	I want to ...	Documentation
Administrator	Perform basic Log Collection implementation	Basic Implementation
Administrator	Set up a lockbox to maintain lockbox settings	Set Up a Lockbox
Administrator	*Start Log Collection Services.	Start Collection Services
Administrator	Configure Log Collection protocols and event sources.	Configure Collection Protocols and Event Sources
Administrator	Verify that Log Collection is working.	Verify That Log Collection Is Working

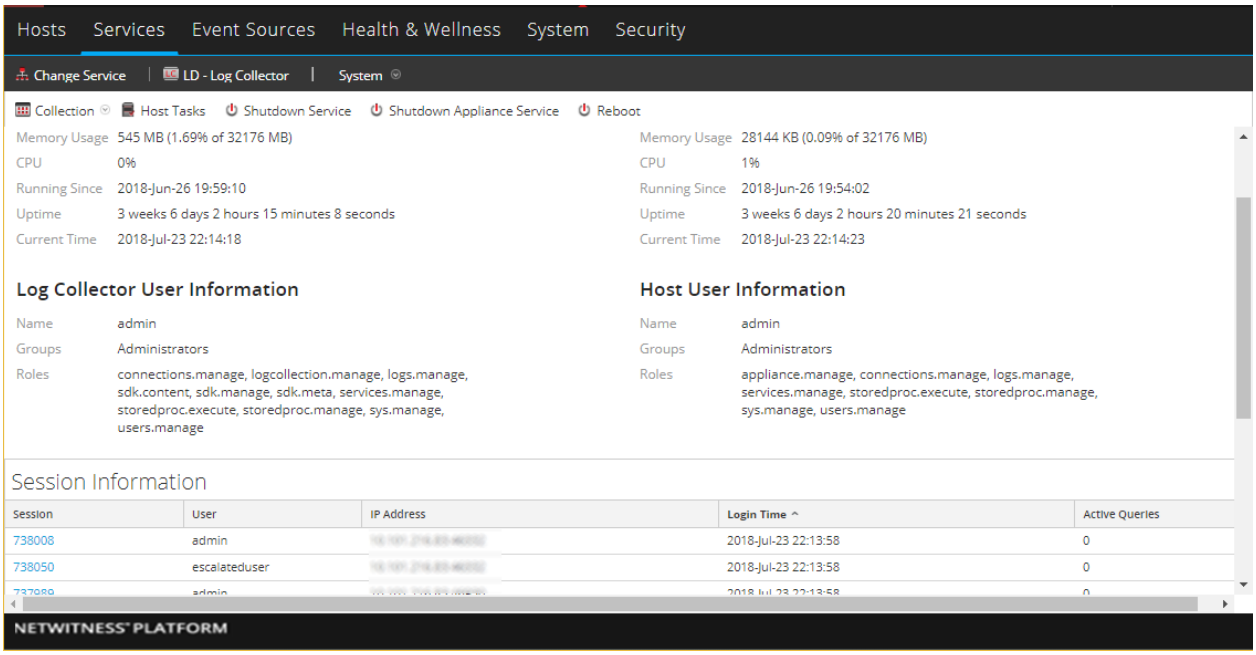
*You can perform this task here.

Related Topics

Basic Implementation

Quick Look

From the Log Collector Service Information Toolbar, you can manage event data using the Collection icon to start event data from a stopped protocol or stop collecting data from a started protocol. From the Host Tasks icon, you can select tasks that you want to run. You can also shutdown your service and reboot your service from the Service Information Toolbar.


Hosts
Services
Event Sources
Health & Wellness
System
Security
Change Service
LD - Log Collector
System
Collection
Host Tasks
Shutdown Service
Shutdown Appliance Service
Reboot
Memory Usage 545 MB (1.69% of 32176 MB)
CPU 0%
Running Since 2018-Jun-26 19:59:10
Uptime 3 weeks 6 days 2 hours 15 minutes 8 seconds
Current Time 2018-Jul-23 22:14:18
Memory Usage 28144 KB (0.09% of 32176 MB)
CPU 1%
Running Since 2018-Jun-26 19:54:02
Uptime 3 weeks 6 days 2 hours 20 minutes 21 seconds
Current Time 2018-Jul-23 22:14:23
Log Collector User Information
Name admin
Groups Administrators
Roles connections.manage, logcollection.manage, logs.manage, sdk.content, sdk.manage, sdk.meta, services.manage, storedproc.execute, storedproc.manage, sys.manage, users.manage
Host User Information
Name admin
Groups Administrators
Roles appliance.manage, connections.manage, logs.manage, services.manage, storedproc.execute, storedproc.manage, sys.manage, users.manage
Session Information
Session
User
IP Address
Login Time
Active Queries
738008 admin 2018-Jul-23 22:13:58 0
738050 escalateduser 2018-Jul-23 22:13:58 0
NETWITNESS PLATFORM

ODBC Event Source Configuration Parameters

This topic tells you how to configure ODBC collection protocol which collects events from event sources that store audit data in a database using the Open Database Connectivity (ODBC) software interface.

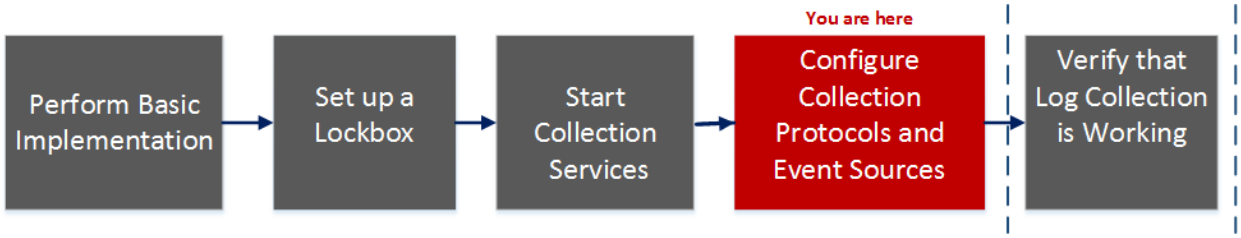
Access ODBC Configuration Parameters

To access the ODBC Event Source Configuration Parameters:

1. Go to **(Admin) > Services** from the NetWitness menu.
2. Select a Log Collection service.
3. Select **> View > Config** to display the Log Collection configuration parameter tabs.
   The **Service Config** view is displayed with the Log Collector **General** tab open.
4. Click the **Event Sources** tab, and select **ODBC/Config** from the drop-down menu.

Workflow

This workflow illustrates the basic tasks needed to start collecting events through Log Collection.



What do you want to do?

Role	I Want to...	Documentation
Administrator	Perform basic Log Collection implementation.	Basic Implementation
Administrator	Set up a lockbox to maintain lockbox settings.	Set Up a Lockbox
Administrator	Start Log Collection services.	Start Collection Services
Administrator	*Configure Log Collection protocols and event sources.	Configure Collection Protocols and Event Sources
Administrator	Verify that Log Collection is working.	Verify That Log Collection Is Working

*You can perform this task here.

Related Topics

- Configure ODBC Event Sources in NetWitness
- Configure Data Source Names (DSNs)
- Troubleshoot ODBC Collection
- Create Custom Typespec for ODBC Collection

Data Source Name (DSN) Parameters

Use the Sources panel to review, add, modify, and delete Data Source Name (DSN) parameters.

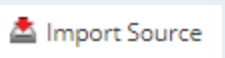
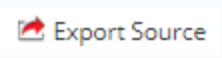
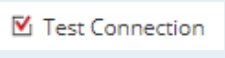
Sources Panel

An ODBC DSN tells the Log Collector how to reach an ODBC endpoint. You refer to an ODBC DSN when you configure a data source name with information such as which ODBC driver to use or the host name and port of the ODBC endpoint.

An ODBC DSN is a sequence of name-value pairs. For information about the valid names for a given ODBC data source type, such as Sybase, Microsoft SQL Server, or Oracle, please download the *DataDirect Connect Series for ODBC User's Guide and DataDirect Connect Series for ODBC User's Guide* in the Progress DataDirect Document Library.

Toolbar

The following table provides descriptions of the toolbar options.

Option	Description
	Opens the Add DSN dialog in which you add an event source for the event source type you selected in the Event Categories panel.
	Deletes the selected event sources.
	Opens the Edit DSN dialog in which you modify the configuration parameters for the selected event source. When you select multiple event sources, this option opens the Bulk Edit Source dialog in which you can edit the parameters values for the selected file directories.
	Opens the Bulk Add Option dialog in which you can import DSN parameters in bulk from a comma-separated values (CSV) file. The Bulk Add Option dialog has the following two options: • Import CSV File • Paste CSV Content
	Creates a **.csv** file that contains the parameters for the selected DSNs.
	Validates the configuration parameters for the selected ODBC database.

Add or Edit DSN Dialog

In this dialog, you add or modify an event source for the selected event source.

Basic Parameters

Note: Required parameters are marked with an asterisk. All other parameters are optional.

Name	Description
DSN*	The data source name (DSN) that defines the database from which to collect events. Select an existing DSN from the drop-down list. For details, see ODBC DSNs Event Source Configuration Parameters.
Username*	User name that the data source name uses to connect to the database. You must specify a user name when you create the event source.
Password	Password that the data source name uses to connect to the database. **Caution:** The password is encrypted internally and is displayed in its encrypted form.
Enabled	Select the checkbox to enable the event source configuration to start collection. The checkbox is selected by default.
Address*	For ODBC, this field is not used. The Log Collector uses the address in the **ODBC.ini** file.

Advanced Parameters

Name	Description
Max Cell Size	Maximum size in bytes of the data that the Log Collector can pull from one cell in the database. The default value is **2048**.
Nil Value	Character string that the Log Collector displays when NIL is returned for a cell in the database. Default value: “” (null).
Polling Interval	Interval (amount of time in seconds) between each poll. The default value is **180**. For example, if you specify 180, the collector schedules a polling of the event source every 180 seconds. If the previous polling cycle is still underway, the collector waits for that cycle to finish. If you have a large number of event sources that you are polling, it may take longer than 180 seconds for the polling to start because the threads are busy.
Max Events Poll	The maximum number of events per polling cycle (how many events collected per polling cycle).

Name	Description
Debug	**Caution:** Caution: Only enable debugging (set this parameter to On or Verbose) if you have a problem with an event source and you need to investigate this problem. Enabling debugging will adversely affect the performance of the Log Collector. Enables or disables debug logging for the event source. Valid values are: • **Off** = (default) disabled • **On** = enabled • **Verbose** = enabled in verbose mode - adds thread information and source context information to the messages. This parameter is designed to debug and monitor isolated event source collection issues. If you change this value, the change takes effect immediately (no restart required). The debug logging is verbose, so limit the number of event sources to minimize performance impact.
Initial Tracking Id	Initial identification code that the Log Collector assigns to this event source if collection is not started. If there is no value for this parameter, the Log Collector starts at the end of the table and only pulls rows after the end of the table as they are added. The default value is "" (null). For Example: `2020-11-11 05:05:11` **Note:** The Initial Tracking ID format varies based on the format of the tracking column used in the data query in the typespec. For Example: If the tracking column value is in epoch, the Initial Tracking ID must be in the epoch format.
Filename	For Microsoft SQL Server Event Sources only, the location of the trace files directory (for example, **C:\MyTraceFiles**). Refer to the NetWitness Microsoft SQL Server Event Source Configuration Guide, located on NetWitness Link.
Test Connection	Checks the configuration parameters specified in this dialog to make sure they are correct.
Cancel	Closes the dialog without adding or modifying DSN parameters.
OK	Adds or modifies the parameters for the DSN.

ODBC DSNs Event Source Configuration Parameters

Open Database Connectivity (ODBC) event sources require Data Source Names (DSNs) so you need to define DSNs with their associate value pairs for ODBC event source configuration.

Access ODBC Configuration Parameters

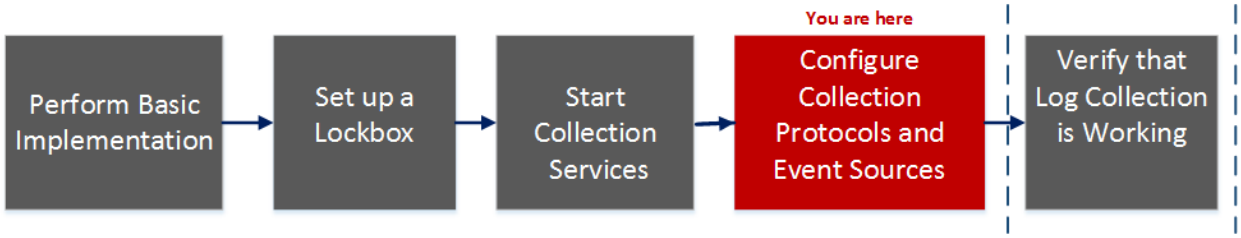
To access the ODBC Event Source Configuration Parameters:

1. Access the Services view by selecting  **(Admin) > Services** from the NetWitness menu.
2. Select a Log Collection service.
3. Under Actions, select  **View > Config** to display the Log Collection configuration parameter tabs.

   The **Service Config** view is displayed with the Log Collector **General** tab open.
4. Click the **Event Sources** tab, and select **ODBC/DSNs** from the drop-down menu.

Workflow

This workflow illustrates the basic tasks needed to start collecting events through Log Collection.



What do you want to do?

Role	I Want to...	Documentation
Administrator	Perform basic Log Collection implementation	Basic Implementation
Administrator	Set up a lockbox to maintain lockbox settings.	Set Up a Lockbox
Administrator	Start Log Collection services.	Start Collection Services
Administrator	*Configure Log Collection protocols and event sources.	Configure Collection Protocols and Event Sources
Administrator	Verify that Log Collection is working.	Verify That Log Collection Is Working

*You can perform this task here.

Related Topics

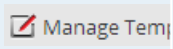
- Configure ODBC Event Sources in NetWitness
- Configure Data Source Names (DSNs)

ODBC DSN Configuration Parameters

This topic describes the Data Source Names DSNs configuration parameters.

DSN Panel

In the DSNs panel, you can add, delete, or edit DSNs and the DSN name-value pairs for ODBC Event sources.

Feature	Description
	Displays the Add DSN dialog in which you define a DSN and its parameters.
	Deletes the selected DSNs.
	Displays the Edit DSN dialog in which you edit the name-value pairs for the selected DSN.
	Displays the Manage DSN Templates dialog in which you can add or delete DSN name-value pair templates.
	Selects DSNs.
DSN	Name of the DSN that you added.
Parameters	**<name-value for="" p="" pairs="" the=""> </name-value>**

Add or Edit DSN Dialog

In this dialog, you add or modify a file directory for the selected event source.

Note: Required parameters are marked with an asterisk. All other parameters are optional.

Feature	Description
DSN Template	Select a predefined DSN value name-value pairs template for the DSN.

Feature	Description
DSN Name*	Add the name of the DSN. You cannot edit a DSN name after you add it. This value must correspond with a DSN entry in the ODBC.ini file. Valid value is a character string that is restricted to the following characters: `[_a-zA-Z][_a-zA-Z0-9]*` This means that the file directory must start with a letter followed by numbers, letters, and underscores (for example, **oracle_executive_compensation**).
Parameters	 Adds a row in which you can define a parameter name-value pair. Deletes the selected parameter name-value pair.  Selects parameter name-value pairs. Name - Enter or modify the parameter name. Value - Enter or modify the value associated with the parameter name.
Cancel	Closes the dialog without adding the DSN and its name-value pairs or saving modifications to the name-value pairs.
Save	Adds the DSN and its name-value pairs or saves modifications to the name-value pairs.

Manage DSN Templates Dialog

In this dialog, you can add or delete DSN name-value pair templates.

Feature	Description
	Template Selection Panel
	Opens the Add Template panel in which you can add a DSN name-value pair template.
	Deletes the selected template.
	Selects a template for deletion or modification.
	Add Template Panel
	Adds a value pair row.
	Deletes a value pair row.
	Selects a value pair row.

Feature	Description
Name	Enter the parameter name.
Value	Enter the value associated with the parameter name.
Cancel	Cancels any changes you made in the dialog.
Save	Adds the DSN and its name-value pairs or saves modifications to the name-value pairs.
Close	Closes the dialog without adding the DSN and its name-value pairs or saving modifications to the name-value pairs.

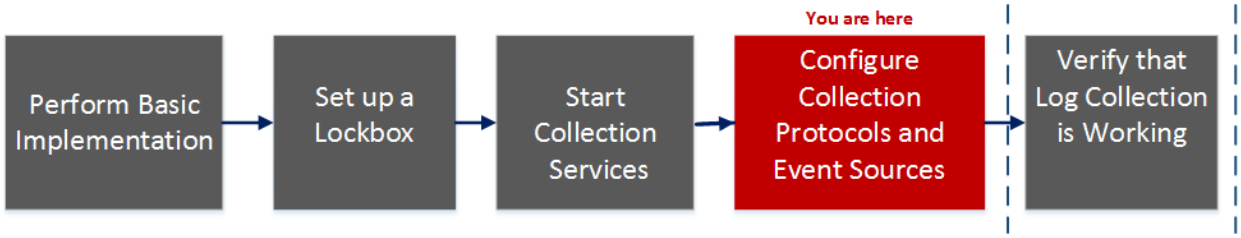
Remote/Local Collectors Configuration Parameters

When you deploy Log Collection, you must configure the Log Collectors to collect the log events from various event sources, and to deliver these events reliably and securely to the Log Decoder host, where the events are parsed and stored for subsequent analysis.

This topic introduces features of the Services Config view > Remote Collectors/Local Collectors tab.

Workflow

This workflow illustrates the basic tasks needed to start collecting events through Log Collection.



What do you want to do?

Role	I Want to...	Documentation
Administrator	Perform basic Log Collection implementation	Basic Implementation
Administrator	Set up a lockbox to maintain lockbox settings.	Set Up a Lockbox
Administrator	Start Log Collection services.	Start Collection Services
Administrator	*Configure Log Collection protocols and event sources.	Configure Collection Protocols and Event Sources
Administrator	Verify that Log Collection is working.	Verify That Log Collection Is Working

*You can perform this task here.

Related Topics

- Provision Local Collectors and Remote Collectors
- Configure Local and Remote Collectors

Services Config View

The Services Config view is the view on which you maintain all the Log Collection parameters. The tab in which you maintain the deployment parameters referred to in this guide is the **Remote/Local** Collectors tab:

- If you are configuring a Local Collector , NetWitness displays the **Remote Collectors** tab so that you can configure the Local Collector to pull events from Remote Collectors.

- If you are configuring a Remote Collector , NetWitness displays the **Local Collectors** tab so that you can configure the Remote Collector to push events to a Local Collector .

Remote Collectors Tab

On a Local Collector, the Remote Collectors panel provides a way to add or delete Remote Collectors from which the Local Collector pulls events.

Column	Description
	Displays the **Add Source** dialog in which you select the Remote Collectors from which you want the Local Collector to pull events.
	Deletes the Remote Collector from the Local Collector Remote Collectors panel.
	Displays the **Edit Source** dialog for the selected Remote Collector .
	Selects Remote Collectors.
Name	Names of the Remote Collectors from which the Local Collector currently pulls events.
Address	IP Addresses of the Remote Collectors from which the Local Collector currently pulls events.
Collections	Choose which collection protocols that the Remote Collector pushes to a Local Collector. You can select any combination of protocols. If you do not select a protocol, NetWitness selects all protocols.

Local Collector Tab

On a Remote Collector , the Local Collector panel provides a way to add or delete the Local Collectors to which you want to the Remote Collector to push events.

Select the **Destination** or **Source** in the **Select Configuration** drop-down menu.

- **Destination** displays the **Add Remote Destination** dialog.
- Source displays the **Add Source** dialog.

The following table describes the Add Source dialog.

Column	Description
	Displays the **Add Source** dialog in which you select the Remote Collectors from which you want the Local Collector to pull events.
	Deletes the Remote Collector from theLocal Collector Remote Collectors panel.

Column	Description
	Displays the **Edit Source** dialog for the selected Remote Collector .
	Selects Remote Collectors.
Name	Names of the Remote Collectors from which the Local Collector currently pulls events.
Address	IP Addresses of the Remote Collectors from which the Local Collector currently pulls events.

The following table describes the Local Collectors Panel.

Column	Description
	Displays the **Add Remote Destination** dialog for the Group that you selected. You add destination Local Collectors for this group to which you want the Remote Collector to push events.
	Deletes the destination Log Collector from the group.
	Displays the **Edit Remote Destination** dialog for the selected destination Local Collector .
	Selects a destination Local Collector .
Destination Name	Displays the name of the destination Local Collector .
Address	Displays the IP address of the destination Local Collector .
Collections	Choose which collection protocols that the Local Collector pulls from a Remote Collector. You can select any combination of protocols. If you do not select a protocol, NetWitness selects all protocols.

Log Collection Tabs

This topic describes the tabs available in the Log Collection view.

Access Log Collection View

To access the log collection view:

1. Go to **(Admin) > Services** from the NetWitness menu.
2. Select a Log Collection service.

3. Under Actions, select **View > Config** to display the Log Collection configuration parameter tabs.

   The **Service Config** view is displayed with the Log Collector **General** tab open.

4. Select any of the available tabs to view or update the corresponding parameters.

Available Tabs

Use the  (Admin) > Services view to maintain Log Collection parameters. It has the following tabs:

- **General**: contains high-level parameters that govern the operation of the Log Collector service and each collection protocol. See Log Collection General Tab for details.
- **Remote Collectors**: use this tab to set up remote collectors. See Configure Local and Remote Collectors for details.
- **Files**: provides an interface for editing Log Collector configuration files.
- **Event Sources**: use this tab to configure collection for your event sources. See Log Collection Event Sources Tab for details.
- **Event Destinations**: Use the Event Destinations tab of the Log Collection service Config view to configure the destination of event data collected by the Log Collector. See Log Collection Event Destinations Tab for details.
- **Settings**: contains parameters for Lockbox security setup, and certificate management.
- **Appliance Service Configuration**: contains configuration parameters for the NetWitness Core Appliance service.

Please refer to the **Files** tab and the **Appliance Service Configuration** tab in the *Host and Services Configuration Guide* for information on the configuration parameters on these tabs.

Log Collection General Tab

This topic introduces features of the service Config view > General tab that relate specifically to Log Collector .

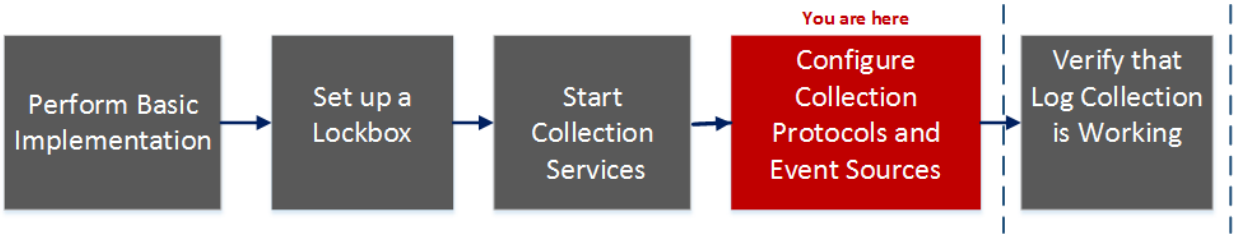
To access the Log Collection General tab:

1. Go to  **(Admin) > Services** from the NetWitness menu.
2. Select a Log Collection service.
3. Click  > **View > Config**.

   The **Service Config** view is displayed with the Log Collector **General** tab open.

Workflow

This workflow illustrates the basic tasks needed to start collecting events through Log Collection.



What do you want to do?

Role	I want to...	Documentation
Administrator	Perform basic Log Collection implementation.	Basic Implementation
Administrator	Set up a lockbox to maintain lockbox settings.	Set Up a Lockbox
Administrator	Start Log Collection services.	Start Collection Services
Administrator	Configure Log Collection protocols and event sources.	Configure Collection Protocols and Event Sources
Administrator	*Verify that Log Collection is working.	Verify That Log Collection Is Working

*You can perform this task here.

Related Topics

- Configure AWS (CloudTrail) Event Sources in NetWitness
- Configure Check Point Event Sources in NetWitness

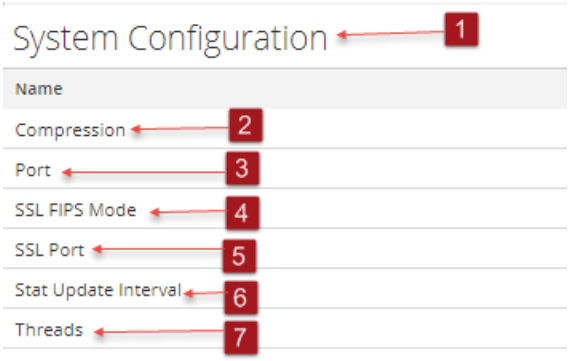
- Configure File Event Sources in NetWitness
- Configure Netflow Event Sources in NetWitness
- Configure ODBC Event Sources in NetWitness
- Configure SDEE Event Sources in NetWitness
- Configure SNMP Event Sources in NetWitness
- Configure Syslog Event Sources
- Configure VMware Event Sources in NetWitness
- Configure Windows Event Sources in NetWitness
- Windows Legacy and NetApp Collection Configuration

Quick Look

The NetWitness administrator must configure event sources to send logs to the collectors. When event sources are configured they poll event sources, retrieve logs, and send the event data to NetWitness).

System Configuration Panel

The System Configuration panel manages service configuration for a NetWitness service. When a service is first added, default values are in effect. You can edit these values to tune performance. Refer to the **General** tab for a description of these parameters.

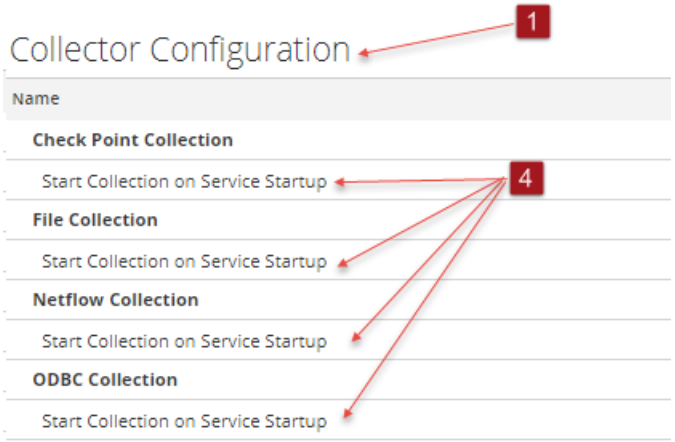


1	System Configuration Panel manages service configuration for a NetWitness service.
2	Compression: The minimum number of bytes that must be transmitted per response before compression. A setting of 0 disables compression. The default value is **0**. A change in value is effective immediately for all subsequent connections.
3	Port: The port on which the service listens. The ports are: • 50001 for Log Collectors • 50002 for Log Decoders • 50003 for Brokers • 50004 for Decoders • 50005 for Concentrators

	• 50007 for other services
4	SSL FIPS Mode: When enabled (**on**), the security of data transmission is managed by encrypting information and providing authentication with SSL certificates. The default value is **off**.
5	SSL Port: The NetWitness Core SSL port on which the service listens. The ports are: • 56001 for Log Collectors • 56002 for Log Decoders • 56003 for Brokers • 56004 for Decoders • 56005 for Concentrators • 56007 for other services
6	Stat Update Interval: The number of milliseconds between statistic updates on the system. Lower numbers cause more frequent updates and can slow down other processes. The default value is **1000**. A change in value is effective immediately.
7	Threads: The number of threads in the thread pool to handle incoming requests. A setting of 0 lets the system decide. The default value is 15. A change takes effect on service restart.

Collector Configuration Panel

The Collector Configuration panel provides a way to enable automatic start of log collection by event source type.



1	Collector Configuration Panel provides a way to enable automatic start of log collection by event source type.
2	Enable All enables the automatic collection for all event types. **Enable All** = start receiving events and collecting logs for all event types when the Log Collector service starts.
3	Disable all disables the automatic collection for all event types. **Disable All** = (default) do not receive event data for all event types until you explicitly start collection.

4 Start Collection on Service Startup enables automatic start, per event source type, of log collection when the Log Collector service starts. Valid values are:

- Selected = start collecting logs when the Log Collector service starts.
- Not selected = (default) do not collect event data until you explicitly start collection.

5 **Apply**: Click **Apply** to save the changes to the parameter values.

Log Collection Event Destinations Tab

Use the Event Destinations tab of the Log Collection service Config view to configure the destination of event data collected by the Log Collector :

- Log Decoders
- Identity Feed

Prerequisites

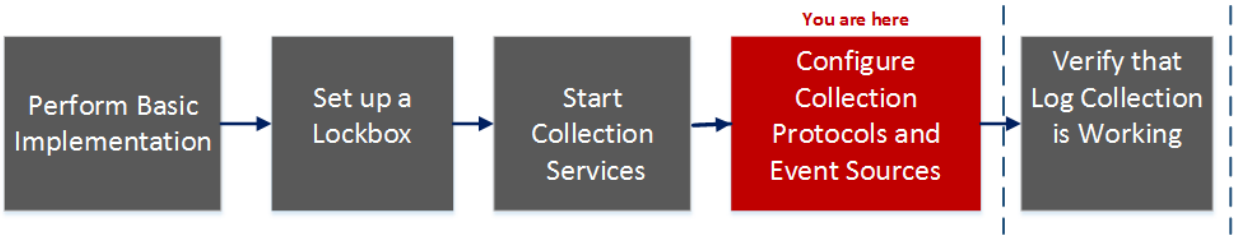
You must implement the following configuration to create an identity feed.

- A Log Collector service with an Identity Feed Event Processor
- A Log Collector service with Windows Collection configured and enabled

Note: See the "Create an Identity Feed" topic in the *Live Resource Management Guide* for more information on how to create and investigate on an identity feed.

Workflow

This workflow illustrates the basic tasks needed to start collecting events through Log Collection.



What do you want to do?

Role	I want to...	Documentation
Administrator	Perform basic Log Collection implementation.	Basic Implementation
Administrator	Set up a to maintain lockbox settings.	Set Up a Lockbox
Administrator	Start Log Collection services.	Start Collection Services
Administrator	*Configure Log Collection protocols and event sources.	Configure Collection Protocols and Event Sources
Administrator	Verify that Log Collection is working.	Verify That Log Collection Is Working

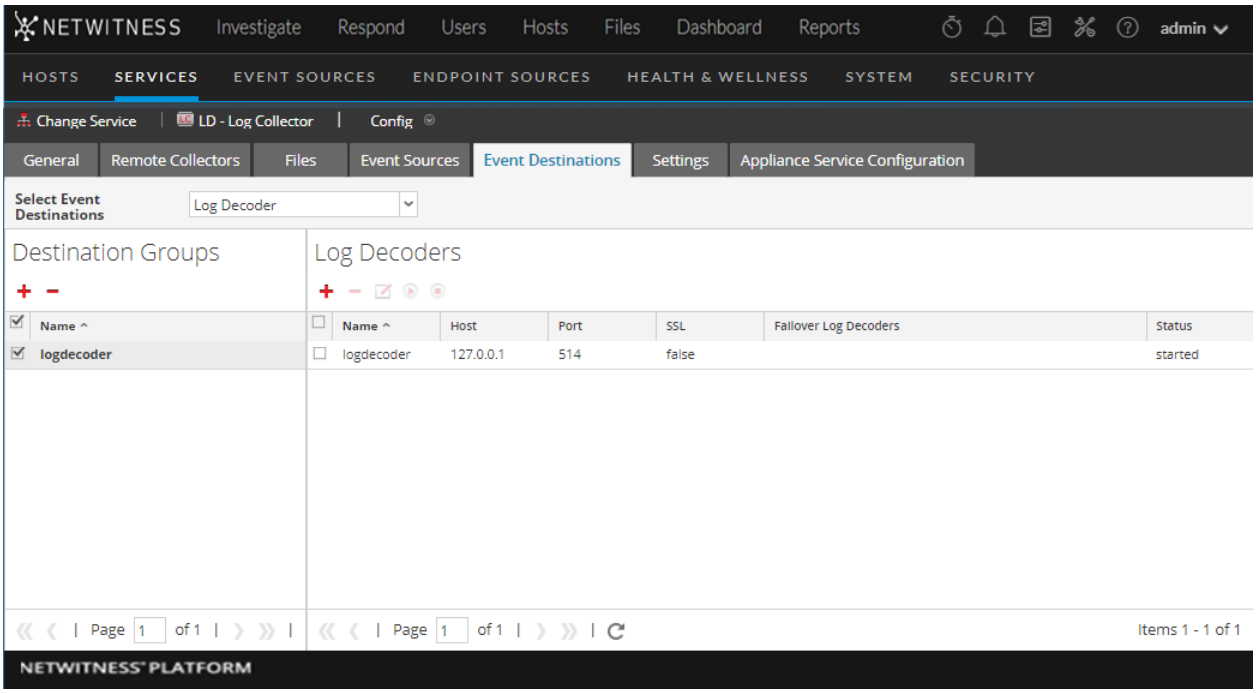
*You can perform this task here.

Related Topics

- See the **Create an Identity Feed** topic in the *Live Resource Management Guide.*

Quick Look

The Event Destinations tab of the Log Collection service Config view allows you to configure the destination of event data collected by the Log Collector.



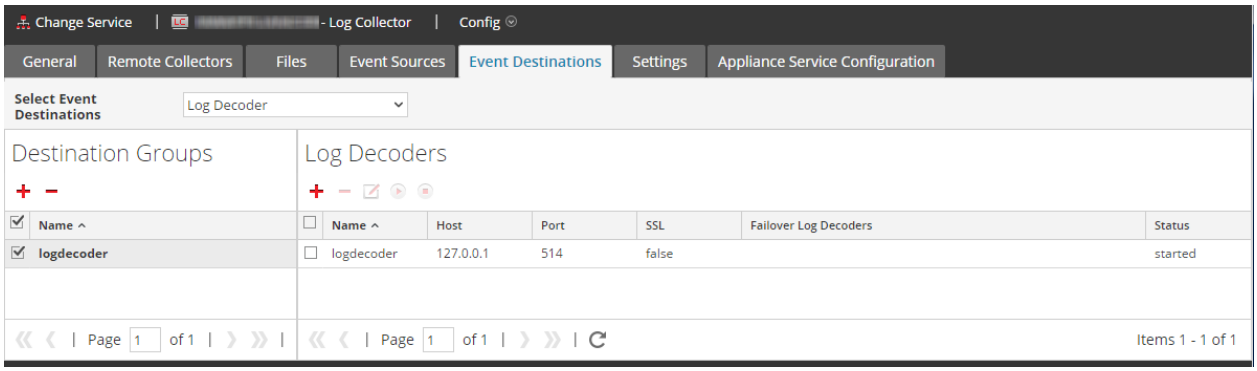
The required permission to access this view is Manage Services.

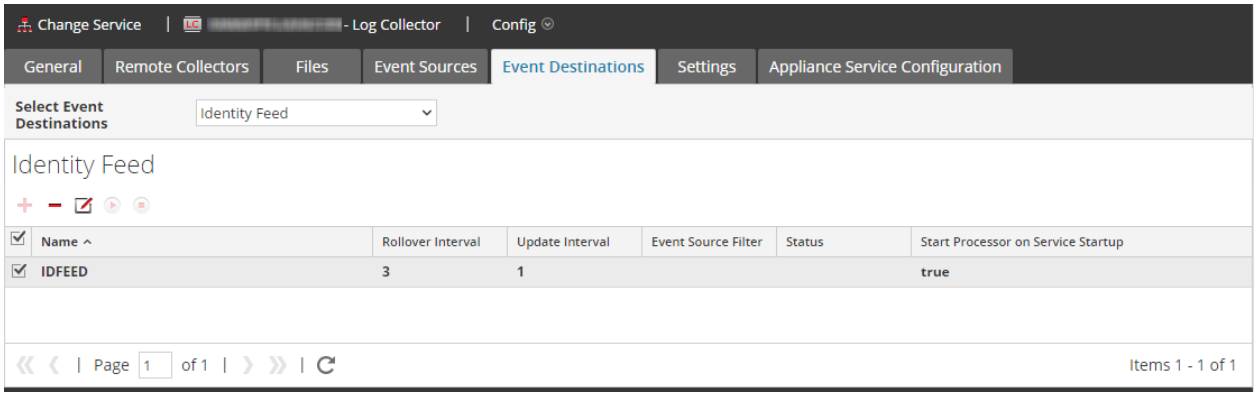
To access the Event Destinations tab:

1. Go to **(Admin) > Services** .
2. Select a Log Collection service.
3. Select **> View > Config** to display the Log Collection configuration parameter tabs.
4. Click the **Event Destinations** tab.
5. In the **Select Event Destinations** drop-down menu:
   - Select **Log Decoder** to configure Log Decoder destinations for event data collected by the Log Collector.

     Note: You must select a Log Decoder service from the Add Log Decoder Destination dialog, but the remainder of the configuration is done automatically.

   - Select **Identity Feed** to configure an identity feed destination for event data collected by the Log Collector.


Change Service
- Log Collector
Config
General
Remote Collectors
Files
Event Sources
Event Destinations
Settings
Appliance Service Configuration
Select Event Destinations
Log Decoder
Destination Groups
Name
logdecoder
Log Decoders
Name
Host
Port
SSL
Failover Log Decoders
Status
logdecoder
127.0.0.1
514
false
started
Page 1 of 1
Items 1 - 1 of 1


Change Service
- Log Collector
Config
General
Remote Collectors
Files
Event Sources
Event Destinations
Settings
Appliance Service Configuration
Select Event Destinations
Identity Feed
Identity Feed
Name
Rollover Interval
Update Interval
Event Source Filter
Status
Start Processor on Service Startup
IDFEED
3
1
true
Page 1 of 1
Items 1 - 1 of 1

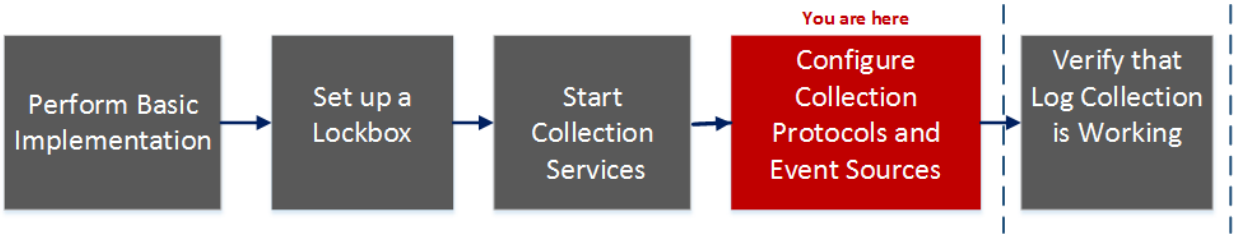
Log Collection Event Sources Tab

Use the Event Sources tab to configure the AWS (CloudTrail), Check Point, File, ODBC, SDEE, Logstash, SNMP, Syslog, SNMP, VMware, Windows, and Windows Legacy event sources.

To access the Event Sources tab, go to (Admin) > **Services** > select **Log Collection service** > **View** > **Config** > **Event Sources**) .

Workflow

This workflow illustrates the basic tasks needed to start collecting events through Log Collection.



What do you want to do?

Role	I want to...	Documentation
Administrator	Perform basic Log Collection implementation.	Basic Implementation
Administrator	Set up a lockbox to maintain lockbox settings.	Set Up a Lockbox
Administrator	Start Log Collection services.	Start Collection Services
Administrator	*Configure Log Collection protocols and event sources.	Configure Collection Protocols and Event Sources
Administrator	Verify that Log Collection is working.	Verify That Log Collection Is Working

*You can perform this task here.

Related Topics

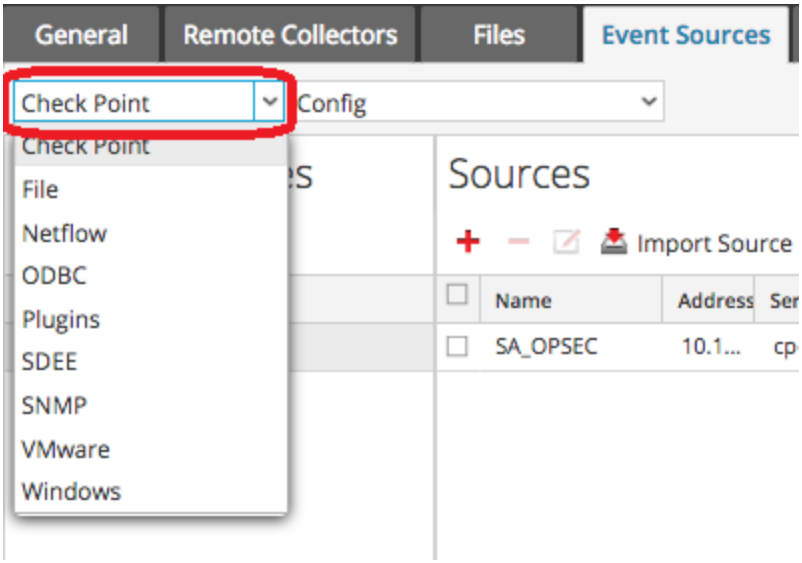
- Configure AWS (CloudTrail) Event Sources in NetWitness
- Configure Check Point Event Sources in NetWitness
- Configure File Event Sources in NetWitness
- Configure ODBC Event Sources in NetWitness
- Configure SDEE Event Sources in NetWitness
- Configure SNMP Event Sources in NetWitness

- Configure Syslog Event Sources
- Configure VMware Event Sources in NetWitness
- Configure Windows Event Sources in NetWitness
- Configure Logstash Event Sources in NetWitness
- Windows Legacy and NetApp Collection Configuration

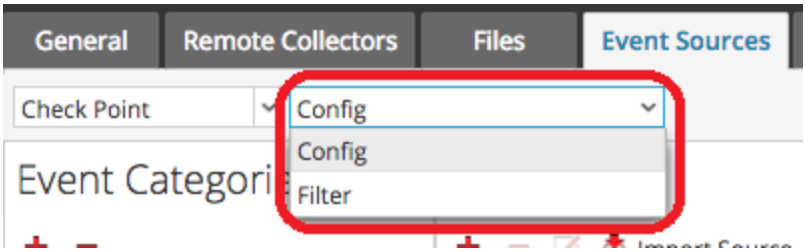
Quick Look

The Config view has two drop-down menus:

- The left-most menu lists all of the available collection protocols.



- The right-most menu has two choices: **Config** and **Filter**.



The Config view in the Event sources tab has two panels: Event Categories and Sources.

Note: For details on the Filter menu item, see Configure Event Filters for a Collector.

Event Source Types Menu

The Log Collector Event Sources tab has a two-box, drop-down menu in which you select the collection protocol and any supporting parameters for that protocol.

In the left box, you select one of the following protocols: Check Point, File, ODBC, Plugins, SDEE, SNMP, SNMP, VMware, Windows, and Windows Legacy.

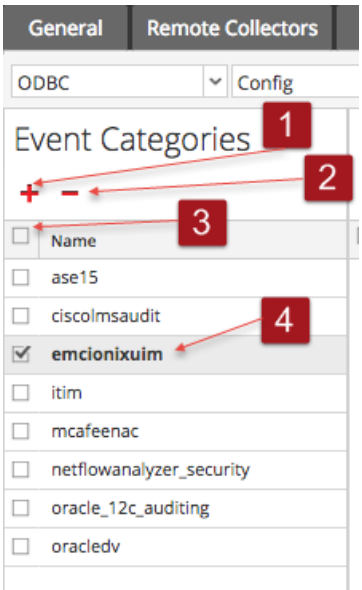
In the right box, you select:

- Config to configure the generic event source parameters for the type you selected in the left drop-down. All generic Config panels have a toolbar with these options:
  - Add, Edit, and Delete
  - Import (also Import Source, Import DSN)
  - Export (also Export Source, Export DSN)
- For ODBC, SNMP, and Windows only:
  - For ODBC, DSNs to configure
  - For SNMP, SNMP v3 User Manager
  - For Windows, Kerberos Realm Configuration

Selecting an option displays a configuration panel where you configure the collection parameters for the event source. The configuration panels are slightly different for different event sources and are described separately.

Event Categories Panel

Once you select a collection protocol, the Event Categories panel is populated with all of the event sources that you have configured for that collection protocol. For example, the following image shows ODBC event sources that have been configured:



The Event Categories panel provides a way to add or delete event source types.

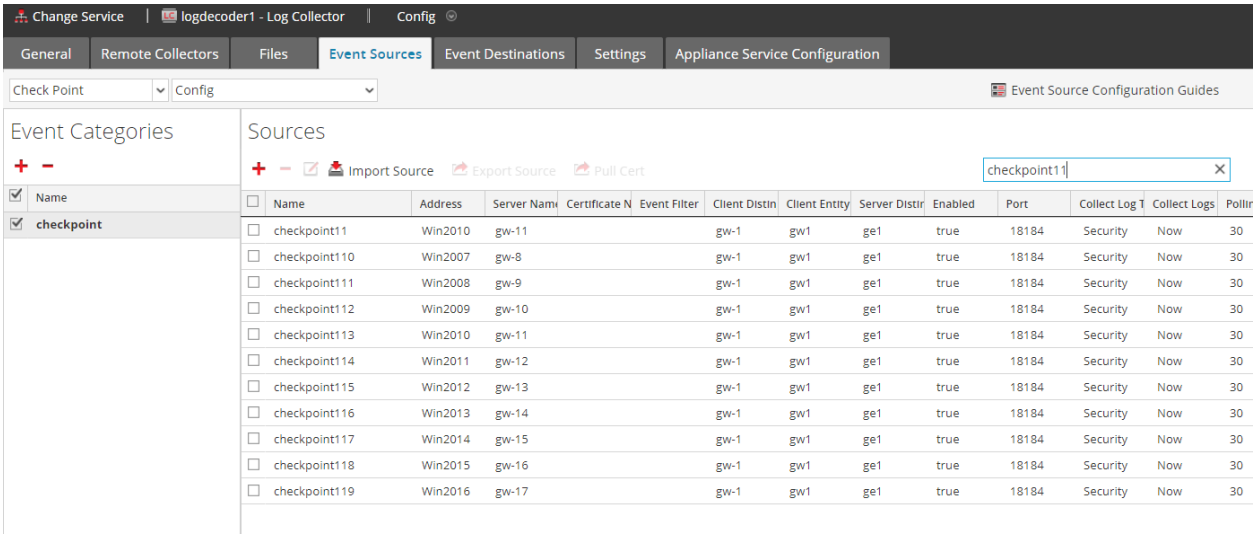
1	Displays the Available Event Source Types dialog from which you select the event source type for which you want to define parameters.
2	Deletes the selected event source types from the Event Categories panel.

3 Selects event source types.
4 Displays the name of the event source types that you have added.

Sources Panel

The Sources panel lists the values of the parameters for the selected event source type. For details, see the individual collection protocol topics.

Below is an example of a list of Check Point event sources. Note that the result set has been limited to sources whose names contain the string **checkpoint11**.



Log Collection Settings Tab

Use the Settings tab to:

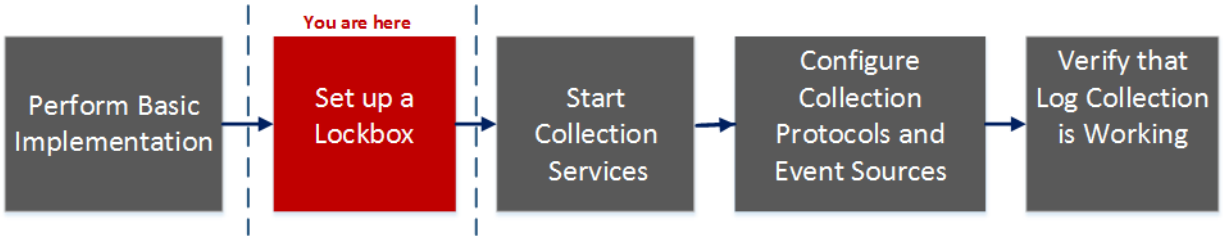
- Set up a lockbox
- Reset Stable System value
- Manage certificates

Caution: If the host name on which the Log Collector is installed is changed after installation, the Log Collector will fail to collect events from event sources. You must reset stable system values if the hostname changes.

To access the Log Collection Settings Tab, go to (Admin) > Services. In the Services grid, select a Log Collector Service. Click Actions menu cropped under Actions and select View > Config.

Workflow

This workflow illustrates the basic tasks needed to start collecting events through Log Collection.



What do you want to do?

Role	I want to...	Documentation
Administrator	Perform basic Log Collection implementation.	Basic Implementation
Administrator	*Set up a lockbox to maintain lockbox settings.	Set Up a Lockbox
Administrator	Start Log Collection services.	Start Collection Services
Administrator	Configure Log Collection protocols and event sources.	Configure Collection Protocols and Event Sources
Administrator	Verify that Log Collection is working.	Verify That Log Collection Is Working

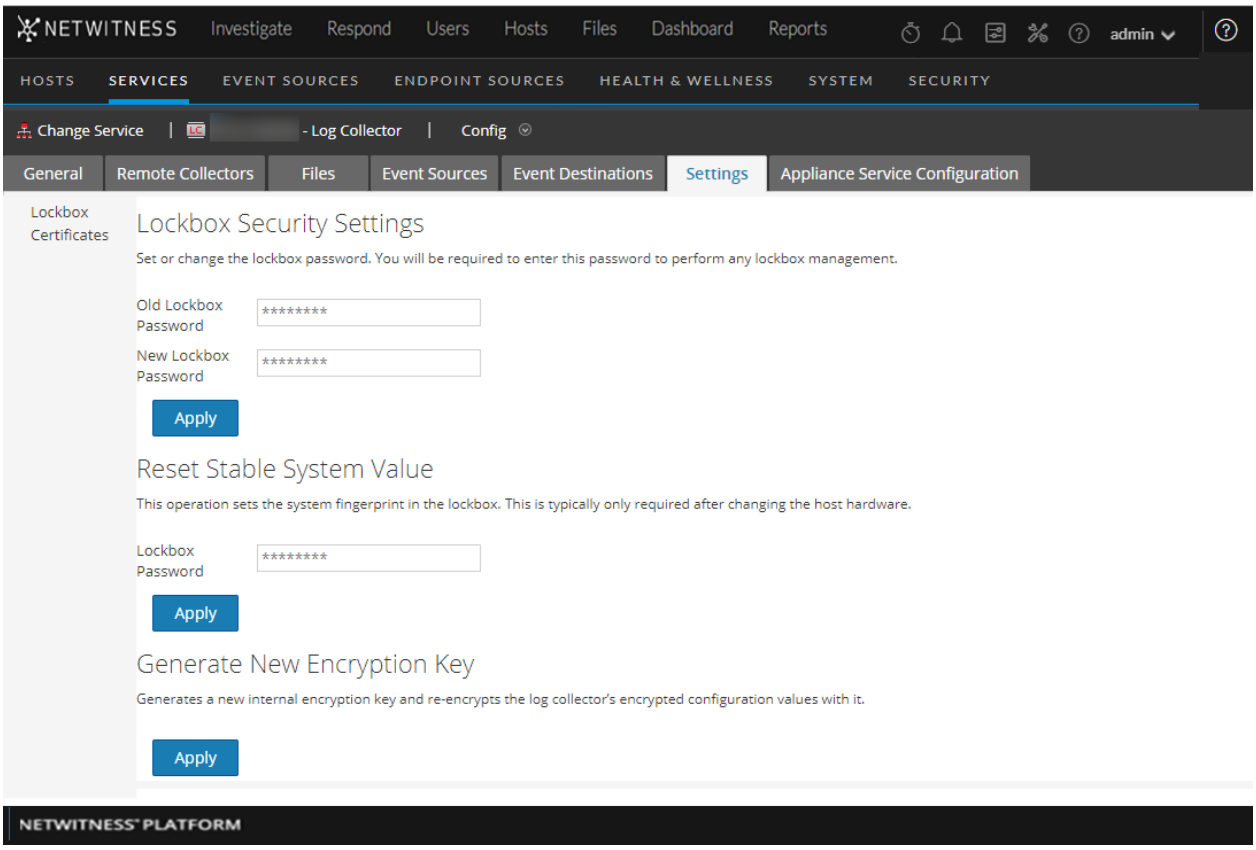
*You can perform this task here.

Related Topics

- See the "Create an Identity Feed topic" in the *Live Resource Management Guide.*

Quick Look

This is an example of the Settings tab.



Troubleshoot Log Collection

This topic describes the format and content of Log Collection Troubleshooting. NetWitness informs you of Log Collector problems or potential problems in the following two ways.

- Log files.
- Health and Wellness Monitoring views.

Junk Syslog Messages

The remote log collector has been made looser in regards to how it handles syslog messages. This was done to reduce the number of dropped messages due to missing parts of the header, or for other minor formatting errors. However, this might also allow syslog event messages that contain junk to get through the parser. If you see such messages in the system, you can add a syslog collection filter to remove events that are sending these messages.

Log Files

If you have an issue with a particular event source collection protocol, you can review debug logs to investigate this issue. Each event source has a Debug parameter that you can enable (set parameter to On or Verbose) to capture these logs.

Caution: Only enable debugging if you have a problem with this event source and you need to investigate this problem. If you have Debug enabled all the time it will adversely affect the performance of the Log Collector.

Health and Wellness Monitoring

Health and Wellness monitoring makes you aware of potential hardware and software problems in a timely manner so that you can avoid to outages. NetWitness recommends that you monitor the Log Collector statistical fields to make sure that the service is operating efficiently and is not at or near the maximum values you have configured. You can monitor the following statistics (Stats) described in the  (Admin) > Health & Wellness view.

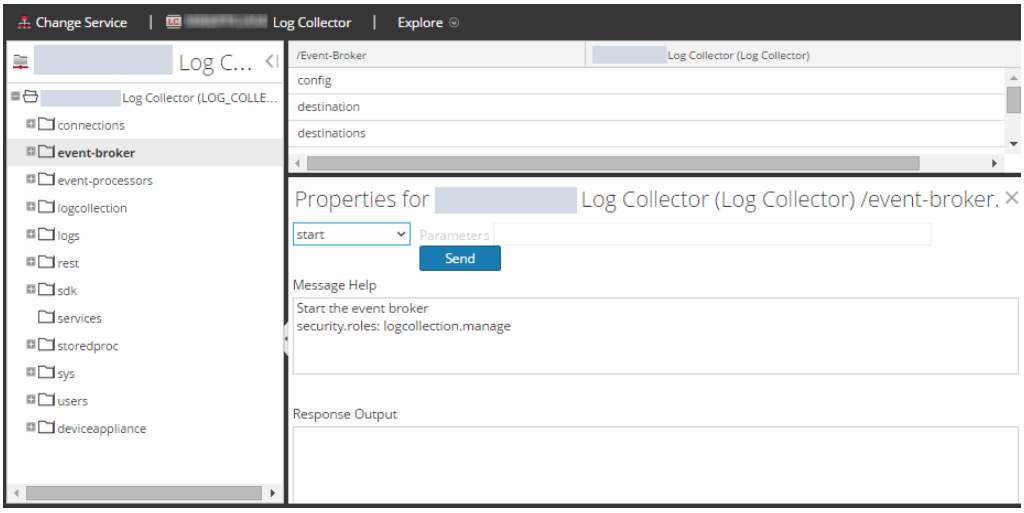
Sample Troubleshooting Format

NetWitness returns the following types of error messages in the log files for.

Log Messages	`timestamp failure (LogCollection) Message-Broker Statistics:...`
	`timestamp failure (AMQPClientBaseLogCollection):...`
	`timestamp failure (MessageBrokerLogReceiver):...`

Possible	The Log Collector cannot reach the Message Broker because the Message Broker:

Cause	• has stopped running • has erroneous connection settings

Solutions	1. `<use the="the" systemctl="systemctl" command="command" on="on" console="console" to="to" check="check" status="status" of="of" message="message" broker="broker" shell="shell" console.="console.">returns the following if the message broker is not running:</use>`
	`prompt$ systemctl status rabbitmq-server`
	`rabbitmq start/running, process 10916`
	2. Start the RabbitMQ Message Broker on event-broker node in the Explore view:
	 

Troubleshooting Logstash

Issue	Unable to re-configure or edit the custom configuration in Logstash.
Possible Cause	While re-configuring the events source configuration, if you update the Event source host's IP and save it, Logstash still connects to the old IP. For example, suppose the Elasticsearch IP is not reachable due to some reason, and you updated the correct IP, but Logstash connects to the old IP (which is not reachable).
Solution	You must restart the Logstash service: `systemctl restart logstash`

Issue	NetWitness Export connector UI does not validate user aggregation permissions.
Possible Cause	Users without aggregation permission are also allowed to configure NetWitness Export Connector in the UI.
Solution	Make sure you configure the NetWitness Export Connector Event source with the right

	aggregation permissions.

Issue	NetWitness Export Connector UI does not validate user credentials.
Possible Cause	NetWitness Export Connector UI displays test configuration successful even if you have configured Export Connector without appropriate aggregation permissions.
Solution	Make sure you configure the NetWitness Export Connector Event source using the correct credentials.

Issue	Logstash event source test configuration fails with memory allocation an error "There is insufficient memory for JVM".
Possible Cause	Sometimes test configuration fails due to insufficient memory available for JVM.
Solution	You must change the JVM -xms to a lower value at `/etc/logstash/jvm.options`. Perform the following: 1. Go to `/etc/logstash/jvm.options` and change `-Xms8g` to a lower value as per the memory configuration of the appliance. 2. Restart the Logstash service using the following command: `service logstash restart`

Issue	Issue with Logstash Performance.
Possible Cause	If EPS is more than 25K, you must increase JVM to get better performance.
Solution	You must change the JVM -xmx to a higher value at `/etc/logstash/jvm.options`. Perform the following: 1. Go to `/etc/logstash/jvm.options` and update JVM to `-Xmx32g`. 2. Restart the Logstash service using the following command: `service logstash restart`