NetWitness® Platform

Custom JDBC Event Source Log Configuration Guide

NETWITNESS
Platform

Custom JDBC

Event Source Product Information:

Vendor: Custom JDBC

Event Source: Custom JDBC

NetWitness Product Information:

Supported On: NetWitness Platform XDR 12.3 and later (both the Admin Server and Log Collector Node)

Collection Method: Logstash

Event Source Class.Subclass: Storage.Database

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

Miscellaneous

October 2024

Contents

Configure JDBC Custom Event Source **5**
Database Auditing 5
JDBC Custom Logstash Pipeline 5
Configure Databases 5
Deploy Logstash JDBC Pipelines from NetWitness Live 5
Set Up Logstash JDBC Event Sources (Pipelines) in NetWitness Platform 6
Custom JDBC Collection Configuration Parameters 8
Basic Parameters 8
Advanced Parameters 9
Examples of SQL statement 10
Configure NetWitness Platform to Collect Events **13**
Getting Help with NetWitness Platform **14**
Self-Help Resources 14
Contact NetWitness Support 14
Feedback on Product Documentation 15

Configure JDBC Custom Event Source

Database Auditing

If you are using database auditing on an Oracle Windows or Unix platform, mysql, postgres and so on, you can collect messages through the NetWitness Platform Logstash JDBC Service.

Collecting messages through the NetWitness Platform Logstash JDBC Service has the following advantages:

- Database auditing collection is server specific.
- You can collect messages from a Windows platform.
- All messages are in a fixed format, making them easier to read.

JDBC Custom Logstash Pipeline

The JDBC Custom Logstash pipeline is a flexible data ingestion pipeline that enables users to pull data from any database using Logstash, a popular open-source data processing tool.

The JDBC Custom Logstash pipeline allows customers to configure and customize the pipeline based on their specific database and requirements. This flexibility allows users to leverage different JDBC drivers, classes, log device parsers based on their choice and the databases they are working with.

You must complete these tasks to configure JDBC Custom Event Source:

I. Configure Databases

II. Deploy Logstash JDBC Pipelines from NetWitness Live

III. Set Up Logstash JDBC Event Sources (Pipelines) in NetWitness Platform

IV. Custom JDBC Collection Configuration Parameters

V. Configure NetWitness Platform to Collect Events

Configure Databases

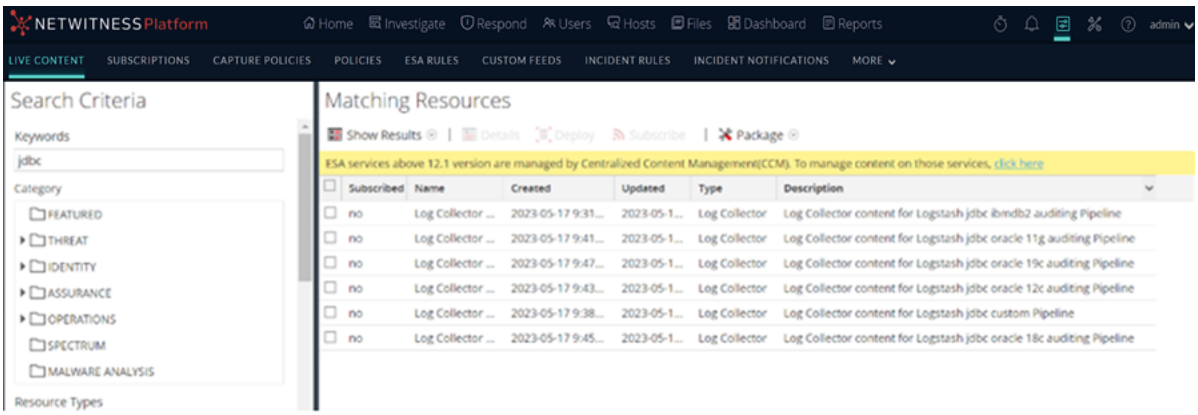
When configuring the event source specific to any supported database, it is recommended to visit the NetWitness community portal for official documentation or configuration guides provided by the respective vendors.

Deploy Logstash JDBC Pipelines from NetWitness Live

Logstash JDBC Pipeline files requires resources available in Live to collect logs.

To deploy Logstash JDBC Pipeline files from Live:

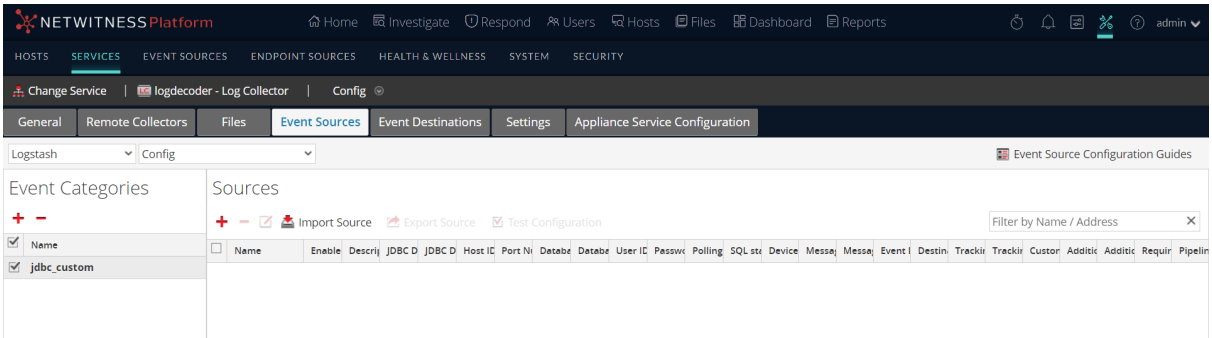
1. In the NetWitness Platform XDR menu, select **Configure > Live Content**.
2. Type **Jdbc** into the Keywords text box and click **Search** to browse Live for Logstash JDBC Custom Pipeline files.
3. Select the JDBC Custom Pipeline item returned from the search .
4. Click **Deploy** to deploy the Logstash JDBC Custom Pipeline files to the appropriate Log Collector in the **Deployment Wizard**.



Set Up Logstash JDBC Event Sources (Pipelines) in NetWitness Platform

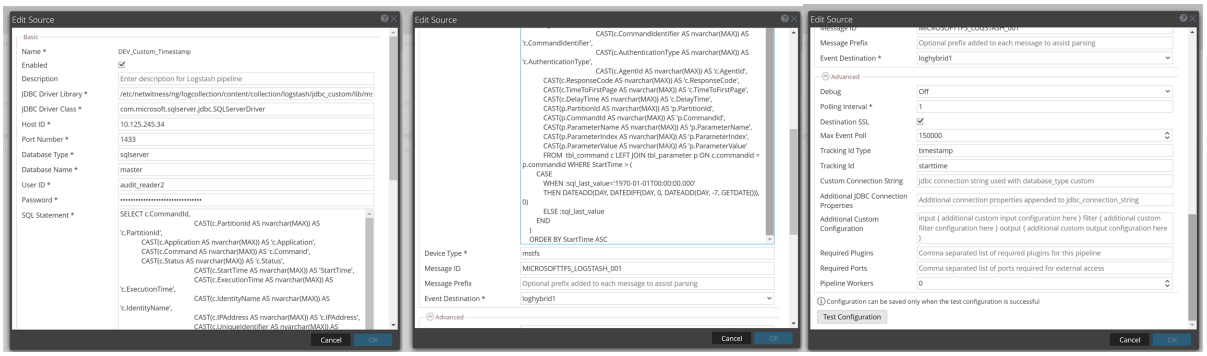
To set up the JDBC Custom Event Source:

1. In the NetWitness Platform menu, select  (Admin) **> Services**.
2. In the **Services** grid, select a **Log Collector** service, and from the **Actions (** **)** menu, choose **View > Config**.
3. In the **Event Sources** view, select **Logstash/Config** from the drop-down menu.
4. In the **Event Categories** panel toolbar, click +.



5. Select **jdbc_custom** from the list and in the **Sources** panel, click + .

   The **Add Source** dialog is displayed.



6. Define the parameter value described in Custom JDBC Collection Configuration Parameters.

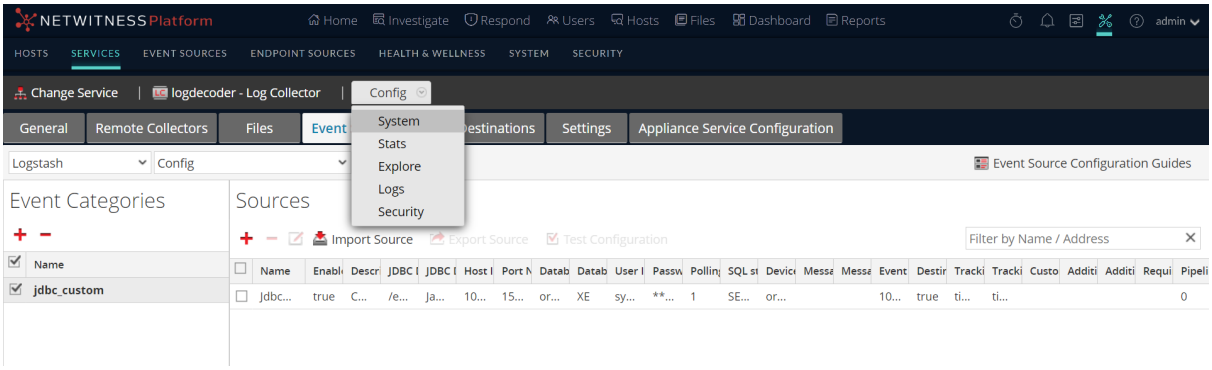
7. Click **Test Configuration**.

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

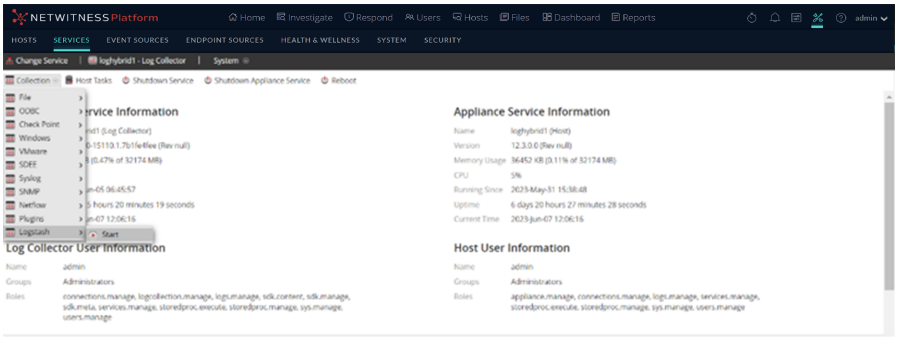
   Note: The log collector may take 1 to 3 minutes to return the test results. If it exceeds the time limit, the test times out and NetWitness Platform XDR displays a Request Timed Out error.

8. If the test is successful, click **OK**. The new event source is displayed in the **Sources** panel.

9. Save the configuration. From the **Actions** menu, choose **System**.



10. In the **Collection** drop-down menu, select **Logstash > Start** to start the log collection.



Custom JDBC Collection Configuration Parameters

The tables below list the configuration parameters required for integrating different database event source with NetWitness Platform through JDBC Custom logstash pipeline.

Basic 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.
JDBC Driver Library*	Enter the JDBC driver library path value given below to interact with a database. **`/etc/netwitness/ng/logcollection/content/collection/logstash/jdbc_custom/lib/<driver-name.jar>`** For any additional or custom drivers, you can place the driver in this location and set the permissions using `chmod 644 <custom-driver-name.jar>`. Replace <driver-name.jar> or <custom-driver-name.jar> with the actual driver name depending on the database server.
JDBC Driver Class*	Enter the JDBC Driver class to interact with a database. For example, for sql server, driver is `mssql-jdbc-12.2.0.jre11.jar` and driver class would be `com.microsoft.sqlserver.jdbc.SQLServerDriver`.
Host ID*	Enter the IP address of the machine where the database server is installed.
Port Number*	Enter the port number that you configured for your event source.
Database Type*	Enter the database type used to define the jdbc_connection_string format. For example, db2, oracle, sqlserver and custom.
Database Name*	Enter the name of the database where the audit table exists.
User ID*	Enter the username of database.
Password*	Enter the password to log into the database.
Polling Interval*	Polling interval takes the input in minutes. Based on the minutes entered, the pipeline will pull the data from the database. For example, If the polling interval is 1, then the pipeline will pull the data from the database for every 1 minute. If the polling interval is 2, then the pipeline will pull the data from the database for every 2 minute. This filed takes the values between 1 to 60.

Name	Description
SQL Statement*	Enter the SQL statement based on the requirement. Here, the text area field also supports any new custom query. For more clarity and examples, refer the Examples of SQL statement section.
Device Type*	Enter the device type of the event source used for parsing.
Message ID	Enter the message group ID to bypass header parsing.
Message Prefix	Enter the prefix added to each message to assist parsing.
Event Destination*	Select the NetWitness Log Collector or Log Decoder to which event needs to be sent from the drop-down list.
Test Configuration	Checks the configuration parameters specified in this dialog to ensure 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.
Destination SSL	Select the checkbox to communicate using destination SSL.
Tracking ID Type	Enter the tracking ID type of the tracking ID field for bookmarking purposes. The tracking ID type value should be `numeric` or `timestamp`.

Name	Description
Tracking ID	Enter the name of the field from the database based on tracking ID type for bookmarking purposes. **Note:** The tracking ID must always be entered in lower case.
CustomConnection String	Enter the custom connection string if the database type value is custom in the basic section.
AdditionalJDBC ConnectionProperties	Enter the additional JDBC connection string properties to append to jdbc_connection_string.
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. - 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.
Required Ports	Enter the list of ports required for external access.
Pipeline Workers	Number of pipeline worker threads allocated for logstash pipeline.

Examples of SQL statement

An SQL statement is a command used to perform operations on data within a relational database. The SQL statement changes with respect to the tracking ID type, which can be either numeric or timestamp.

When tracking ID type is a numeric field:

```
SELECT column-1, column-2, column-3, ..., column-N
FROM Table-A
WHERE tracking_id > (
CASE
WHEN :sql_last_value=0
THEN X
ELSE :sql_last_value
END
```

```
) ORDER BY tracking_id ASC
```

Replace "X" with the ID from which the collection should start.

Example:

```
SELECT c.CommandId AS 'c.CommandId',
CAST(c.Command AS nvarchar(MAX)) AS 'c.Command',
CAST(c.Status AS nvarchar(MAX)) AS 'c.Status',
CAST(c.StartTime AS nvarchar(MAX)) AS 'StartTime',
CAST(p.ParameterName AS nvarchar(MAX)) AS 'p.ParameterName',
CAST(p.ParameterIndex AS nvarchar(MAX)) AS 'p.ParameterIndex',
CAST(p.ParameterValue AS nvarchar(MAX)) AS 'p.ParameterValue'
FROM tbl_command c LEFT JOIN tbl_parameter p ON c.CommandId =
p.commandid
WHERE c.CommandId > (
CASE
WHEN :sql_last_value=0
THEN 10
ELSE :sql_last_value
END
) ORDER BY c.CommandId ASC
```

In the above example, collection starts from the ID "10."

The tracking ID and tracking ID type for this SQL statement are as follows:

tracking id: c.commandid

tracking id type: numeric

When tracking ID type is a timestamp field:

```
SELECT column-1, column-2, column-3, ..., column-N
FROM Table-A
WHERE tracking_id > (
CASE
WHEN :sql_last_value='1970-01-01T00:00:00.000'
THEN 'YYYY-MM-DDThh:mm:ss.sss'
ELSE :sql_last_value
END
) ORDER BY tracking_id ASC
```

Example:

```
SELECT c.CommandId AS 'CommandId',
```

```
CAST(c.Command AS nvarchar(MAX)) AS 'c.Command',
CAST(c.Status AS nvarchar(MAX)) AS 'c.Status',
CAST(c.StartTime AS nvarchar(MAX)) AS 'StartTime',
CAST(p.ParameterName AS nvarchar(MAX)) AS 'p.ParameterName',
CAST(p.ParameterIndex AS nvarchar(MAX)) AS 'p.ParameterIndex',
CAST(p.ParameterValue AS nvarchar(MAX)) AS 'p.ParameterValue'
FROM tbl_command c LEFT JOIN tbl_parameter p ON c.commandid =
p.commandid
WHERE StartTime > (
CASE
WHEN :sql_last_value='1970-01-01T00:00:00.000'
THEN '2024-10-15T00:00:00.000'
ELSE :sql_last_value
END
)
ORDER BY StartTime ASC
```

In the above example, collection starts from 15th October 2024.
The tracking ID and tracking ID type for this SQL statement are as follows:

tracking ID: starttime

tracking ID type: timestamp

Configure NetWitness Platform to Collect Events

To configure NetWitness platform to collect events:

1. You must start capture on the Log Decoder to which you are sending the Logstash data. To start or restart network capture on a Log Decoder:

   i. In the NetWitness Platform menu, select **(Admin) > Services**. The Services view is displayed.

   ii. Select a **Log Decoder** service.

   iii. Under **Actions**, select **View** > **System**.

   iv. In the toolbar, click **Start Capture**.

   Note: If the toolbar is displaying the **Stop Capture ()** icon, then capture has already started.

   By default, Log Decoders support events that are up to 32 KB in size. If the events are getting truncated on the Log Decoder, use the following procedure to change the event size:

   1. Change `LogDecoder REST config at http://LogDecoder_IP:50102/decoder/config`, where LogDecoder_IP is the IP address of your Log Decoder.
   2. Set `pool.packet.page.size` to 64 KB.
   3. Restart the Log Decoder. This is required after you change the `pool.packet.page` value.

   Note: If you are collecting events larger than 64 KB in size, follow instructions above in the Filter out unwanted logs section. You can drop unwanted logs or fields for a specific event source to reduce the size of the incoming data.

Getting Help with NetWitness Platform

Self-Help Resources

There are several options that provide you with help as you need it for installing and using NetWitness:

- See the documentation for all aspects of NetWitness here: https://community.netwitness.com/t5/netwitness-platform/ct-p/netwitness-documentation.
- Use the **Search** and **Create a Post** fields in NetWitness Community portal to find specific information here: https://community.netwitness.com/t5/netwitness-discussions/bd-p/netwitness-discussions.
- See the NetWitness Knowledge Base: https://community.netwitness.com/t5/netwitness-knowledge-base/tkb-p/netwitness-knowledge-base.
- See the documentation for Logstash JDBC input plugin here: https://www.elastic.co/guide/en/logstash/current/plugins-inputs-jdbc.html.
- See Troubleshooting section in the guides.
- See also NetWitness® Platform Blog Posts.
- If you need further assistance, Contact NetWitness Support.

Contact NetWitness Support

When you contact NetWitness Support, please provide the following information:

- The version number of the NetWitness Platform or application you are using.
- Logs information, even source version, and collection method.
- If you have problem with an event source, enable **Debug** parameter (set this parameter to **On** or **Verbose**) and collect the debug logs to share with the NetWitness Support team.

Use the following contact information if you have any questions or need assistance.

NetWitness Community Portal	https://community.netwitness.com In the main menu, click **Support > Case Portal > View My Cases**.
International Contacts (How to Contact NetWitness Support)	https://community.netwitness.com/t5/support/ct-p/support
Community	https://community.netwitness.com/t5/netwitness-discussions/bd-p/netwitness-discussions

Feedback on Product Documentation

You can send an email to feedbacknwdocs@netwitness.com to provide feedback on NetWitness Platform documentation.