How-To Library

# Generating User Activity Event Logs in Secure@Source using Dynamic Data Masking

# Abstract

You can configure Dynamic Data Masking to send user activity event logs in a Common Event Format (CEF) format to Secure@Source using a custom logger.

# Supported Versions

- Dynamic Data Masking 9.9.1
- Secure@Source 5.0 and later

# Table of Contents

# Overview

Dynamic Data Masking can send user activity event logs in a Common Event Format (CEF) format to Secure@Source using a custom logger. You configure the CEF format using Dynamic Data Masking symbols to send user activity information such as the user, the data store queried, and the query to Secure@Source for all user interaction with the data store. Dynamic Data Masking continuously sends user activity to Secure@Source.

# Prerequisites

Before you send user activity events to Secure@Source, you must perform the following steps:

1. Configure the CEF format.
2. Create a database node.
3. Create a custom logger and appender.
4. Create a rule set.
5. Create a connection rule.
6. Identify sensitive fields in the Secure@Source data store.

## *1. Configure the CEF Format*

To generate events in Secure@Source, configure the following CEF in Dynamic Data Masking:

```
CEF: 0|Informatica Inc.|DDM|\(DDM_VERSION)|Audit|Audit.DAM|Informative|dst=\(AUTH_DATABASE_IP)
duser=\(AUTH_USERNAME) src=\(CLIENT_IP) rt=\(AUTH_STATEMENT_RECEIVED_TIME) cs1=\
(AUTH_DATABASE_NAME) cs1Label=Database cs2=10 cs2Label=AffectedRows cs3=\(AUTH_ORIG_STATEMENT)
cs3Label=ParsedQuery cs4=\(AUTH_PROGRAM_NM) cs4Label=ApplicationName
```

The following table describes the Dynamic Data Masking symbols in the CEF format. The database user session populates each symbol value:

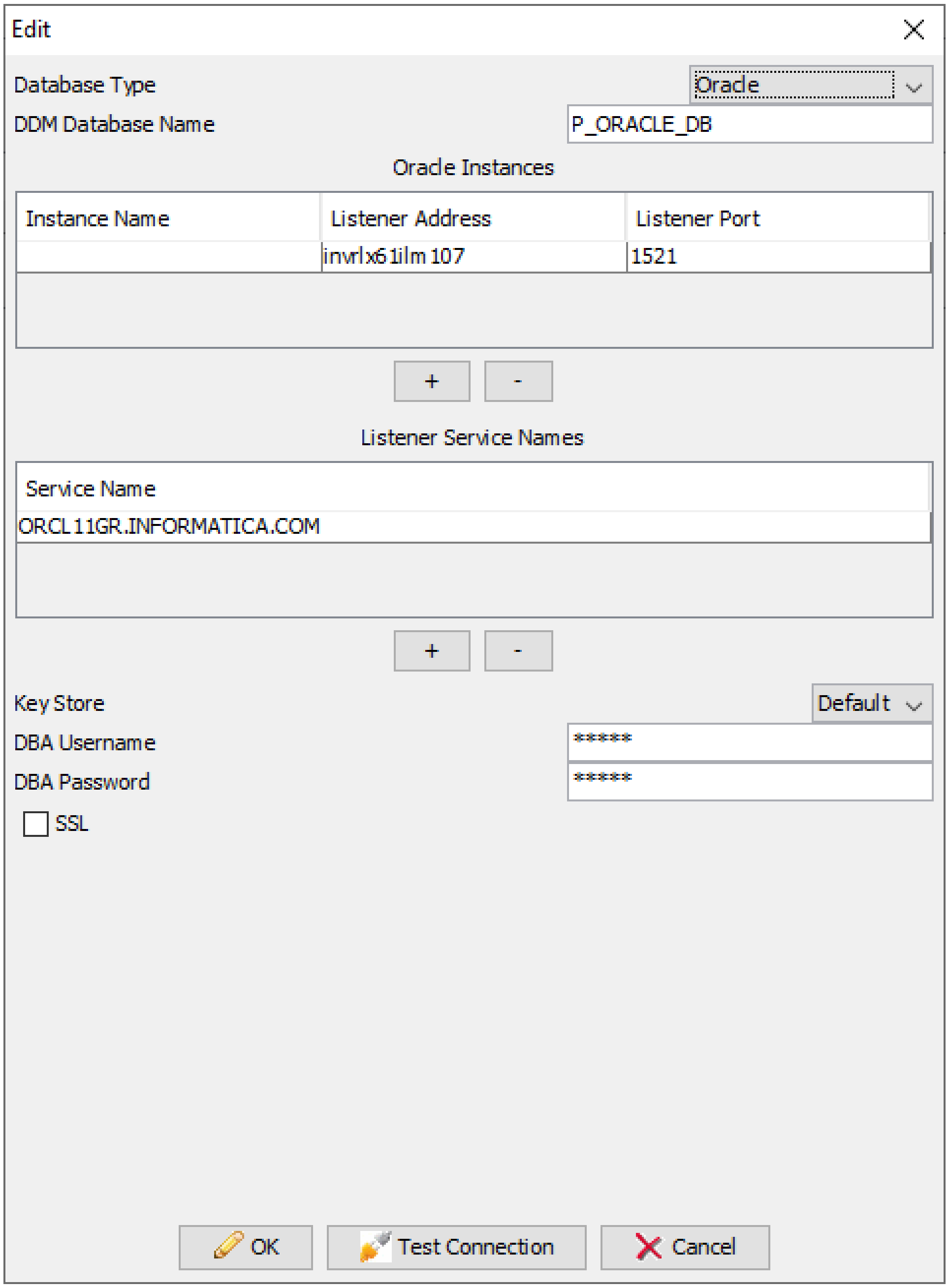
| Symbol | Description |
|---|---|
| DDM_VERSION | Dynamic Data Masking version. |
| AUTH_DATABASE_IP | Data store IP address. |
| AUTH_USERNAME | Database user name. |
| AUTH_STATEMENT_RECEIVED_TIME | Query received time to Dynamic Data Masking. |
| AUTH_DATABASE_NAME | Name of the data store. |
| AUTH_ORIG_STATEMENT | Query executed by the client. |
| AUTH_PROGRAM_NM | Client program name used to query the data store. |

For more information about symbols, see https://docs.informatica.com/data-security-group/dynamic-data-masking/9-9-1/user-guide/security-rules/security-rule-matchers/symbol-matcher.html.

## 2. Create a Database Node

In Dynamic Data Masking, create a database node. Ensure that the name and connection details match the Secure@Source data store for which you want to generate events. The DBA user name that you provide for the database node must have read access to all schema tables.

The following image shows an example database node:



Edit

Database Type: Oracle

DDM Database Name: P_ORACLE_DB

Oracle Instances

| Instance Name | Listener Address | Listener Port |
|---|---|---|
| | invrlx61ilm107 | 1521 |

Listener Service Names

| Service Name |
|---|
| ORCL11GR.INFORMATICA.COM |

Key Store: Default

DBA Username: ******

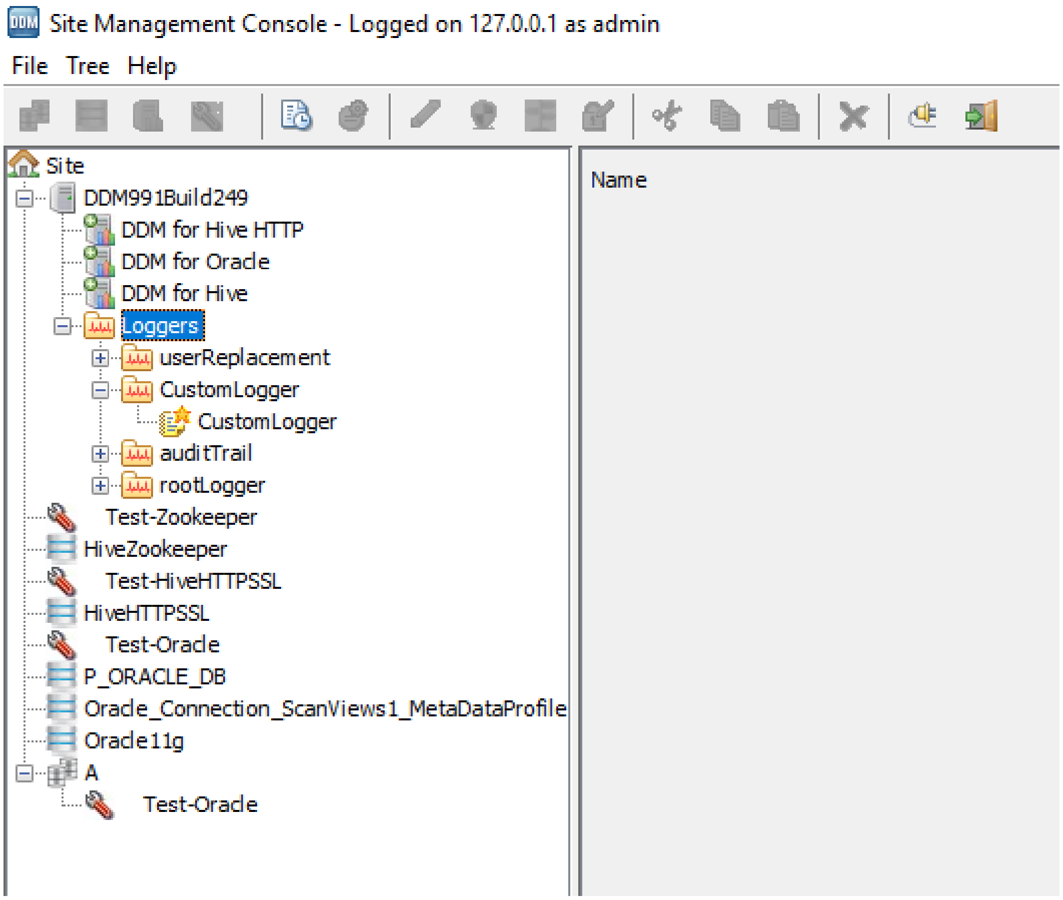
DBA Password: ******

☐ SSL

For more information about creating database nodes for different database types, see https://docs.informatica.com/data-security-group/dynamic-data-masking/9-9-1/administrator-guide/connection-management.html.

## 3. Create a Custom Logger and Appender

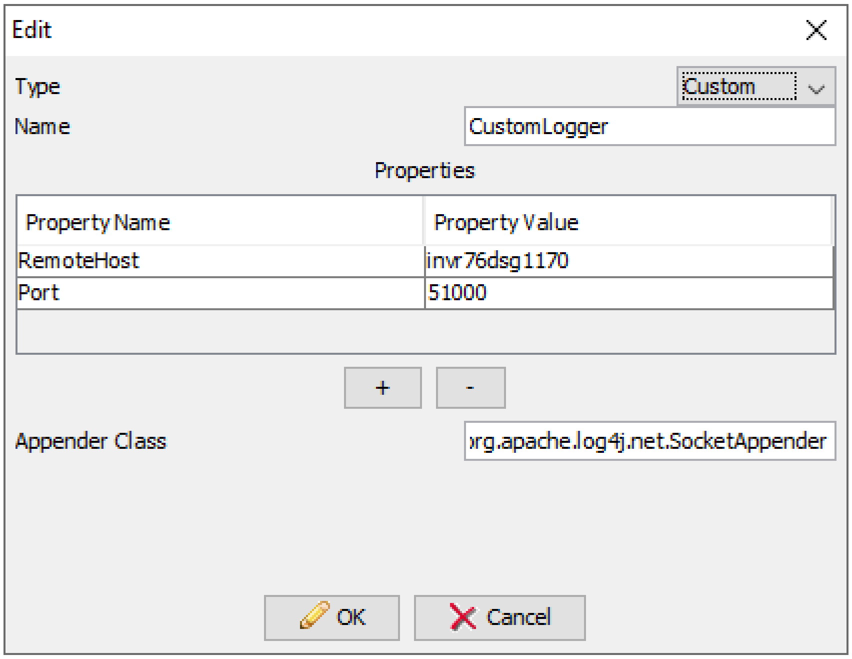
1. In Dynamic Data Masking, create a new logger under the **Loggers** folder.
   The following image shows the **CustomLogger** in the newly created **Loggers** folder:



2. Under the **CustomLogger** folder, create a new appender with a type of **Custom** and the following details:

| Property | Value |
|---|---|
| RemoteHost | Secure@Source host name where the TCP/IP listener runs. |
| Port | TCP/IP listener port. Default port is 51000. |
| Appender Class | org.apache.log4j.net.SocketAppender |

The following image shows an example of an appender created under the **CustomLogger** folder:



Edit

Type: Custom

Name: CustomLogger

Properties

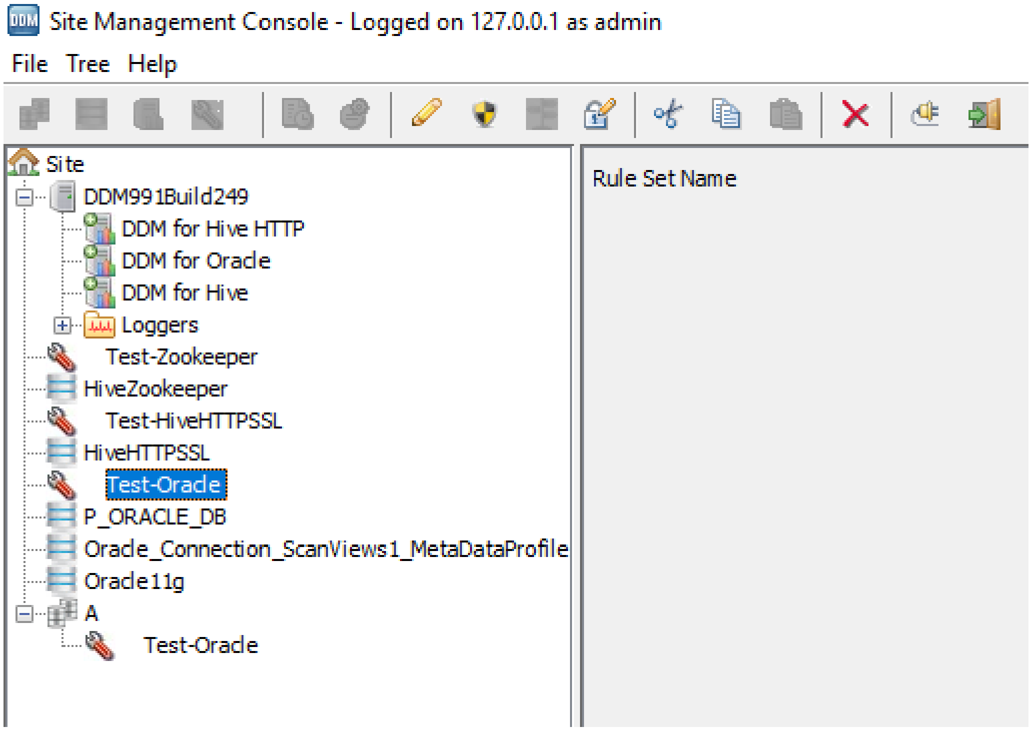
| Property Name | Property Value |
|---|---|
| RemoteHost | invr76dsg1170 |
| Port | 51000 |

Appender Class: ›rg.apache.log4j.net.SocketAppender

## 4. Create a Rule Set

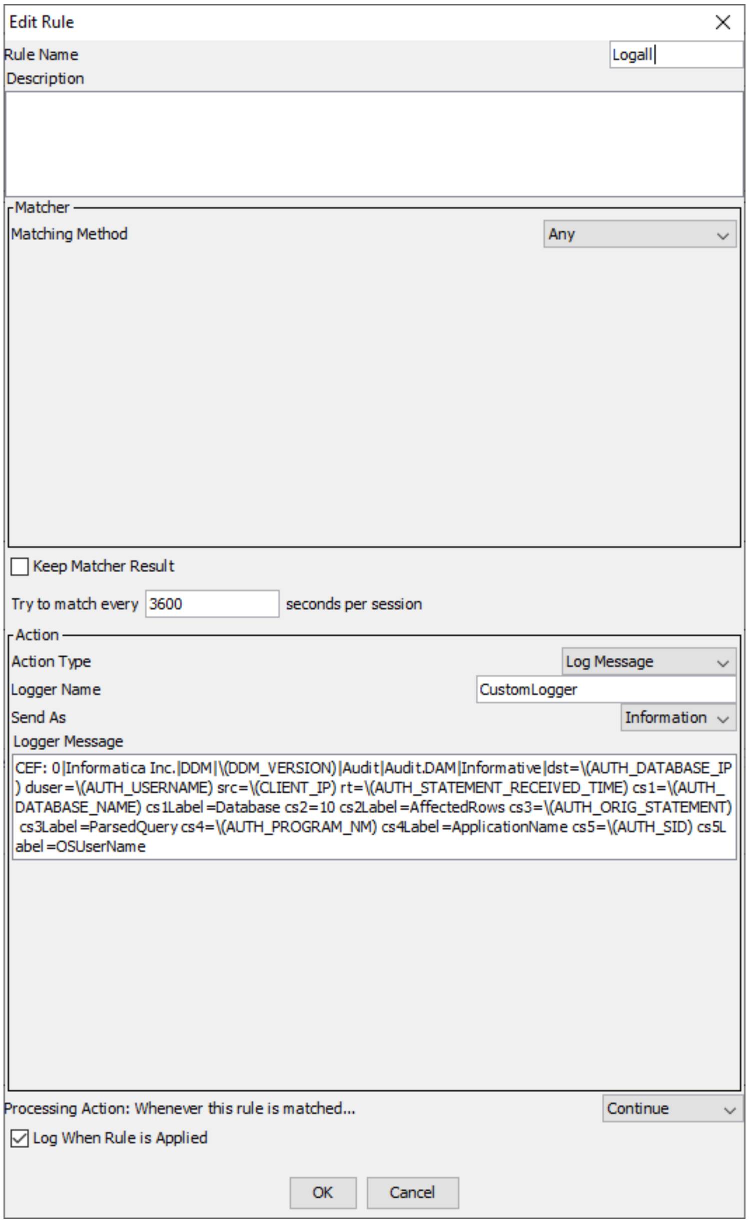
Create a rule set in Dynamic Data Masking to contain the rule with the CEF format.

1. In Dynamic Data Masking, create a rule set under the **Site** node.
   The following image shows an example of the **Test-Oracle** rule set:



Site Management Console - Logged on 127.0.0.1 as admin

2. Create a rule under the rule set to capture and define the CEF format. The following image shows an example of a rule with the CEF format:



Rule Name: Logall

Description:

Matching Method: Any

☐ Keep Matcher Result

Try to match every 3600 seconds per session

Action Type: Log Message

Logger Name: CustomLogger

Send As: Information

Logger Message:

```
CEF: 0|Informatica Inc.|DDM|\(DDM_VERSION)|Audit|Audit.DAM|Informative|dst=\(AUTH_DATABASE_IP
) duser=\(AUTH_USERNAME) src=\(CLIENT_IP) rt=\(AUTH_STATEMENT_RECEIVED_TIME) cs1=\(AUTH_
DATABASE_NAME) cs1Label=Database cs2=10 cs2Label=AffectedRows cs3=\(AUTH_ORIG_STATEMENT)
 cs3Label=ParsedQuery cs4=\(AUTH_PROGRAM_NM) cs4Label=ApplicationName cs5=\(AUTH_SID) cs5L
abel=OSUserName
```

Processing Action: Whenever this rule is matched... Continue

☑ Log When Rule is Applied

3. Define the **Action Type** as **Log Message**.

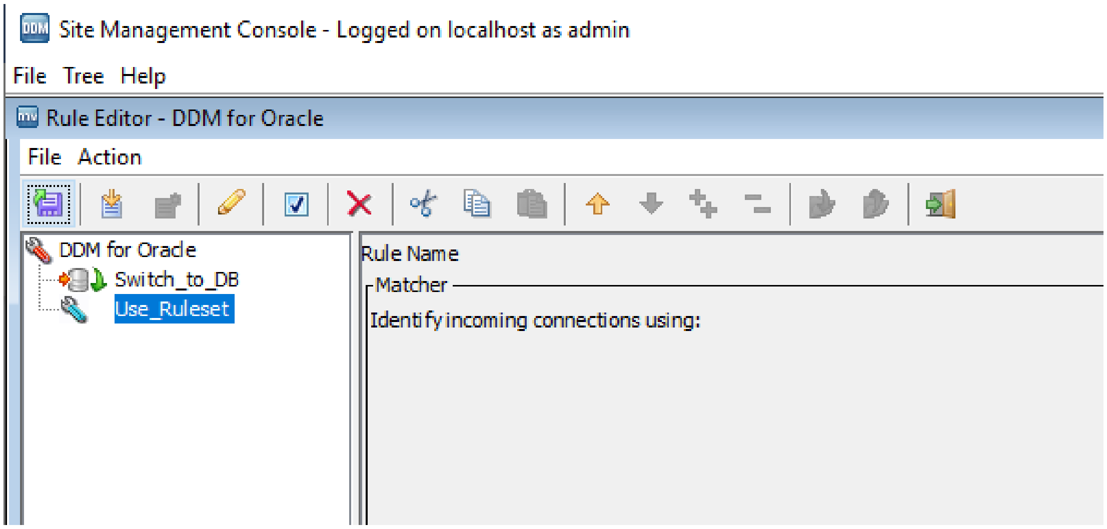
4. Define the **Logger name** as the newly defined folder name. In this example, the **Logger name** specifies the **CustomLogger** folder.

For more information, see https://docs.informatica.com/data-security-group/dynamic-data-masking/9-9-1/user-guide/preface.html.

## 5. Create a Connection Rule

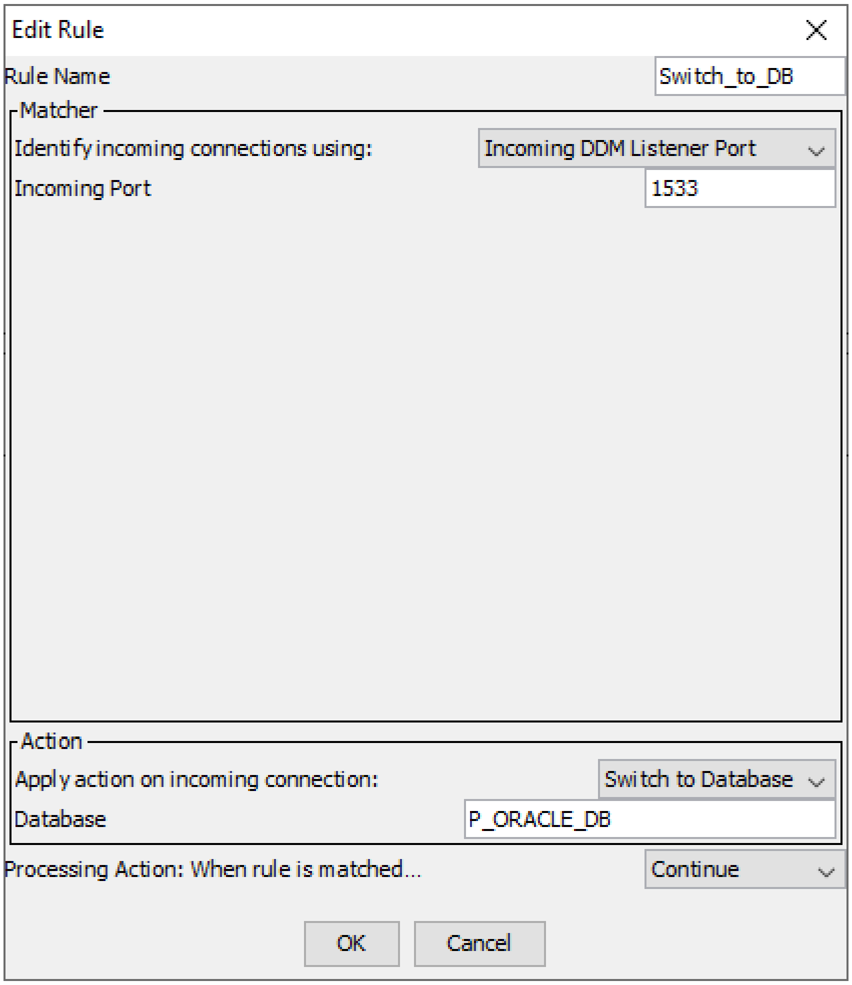
After you create the database node and rule set, you must create a connection rule in Dynamic Data Masking so that user activity event logs are generated and sent to the Secure@Source TCP/IP listener.

Under the database service, create two connection rules in Dynamic Data Masking as shown in the following images:


Site Management Console - Logged on localhost as admin
File Tree Help
Rule Editor - DDM for Oracle
File Action
DDM for Oracle
Switch_to_DB
Use_Ruleset
Rule Name
Matcher
Identify incoming connections using:

The rules enable database clients to connect to the database node through Dynamic Data Masking and capture log events in the CEF format that you defined.

In the following example, the first rule is Switch_to_DB:



Edit Rule

Rule Name: Switch_to_DB

Matcher

Identify incoming connections using: Incoming DDM Listener Port

Incoming Port: 1533

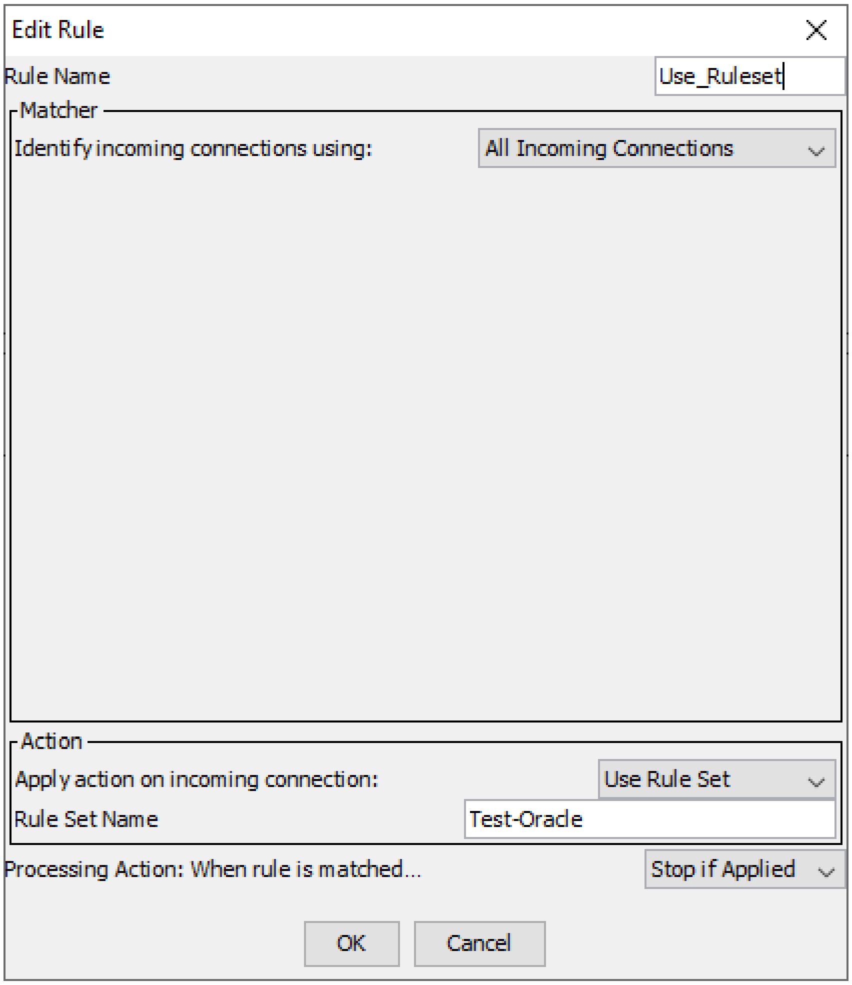
Action

Apply action on incoming connection: Switch to Database

Database: P_ORACLE_DB

Processing Action: When rule is matched... Continue

In the following example, the second rule is Use_Ruleset:



Rule Name: Use_Ruleset

Identify incoming connections using: All Incoming Connections

Apply action on incoming connection: Use Rule Set

Rule Set Name: Test-Oracle

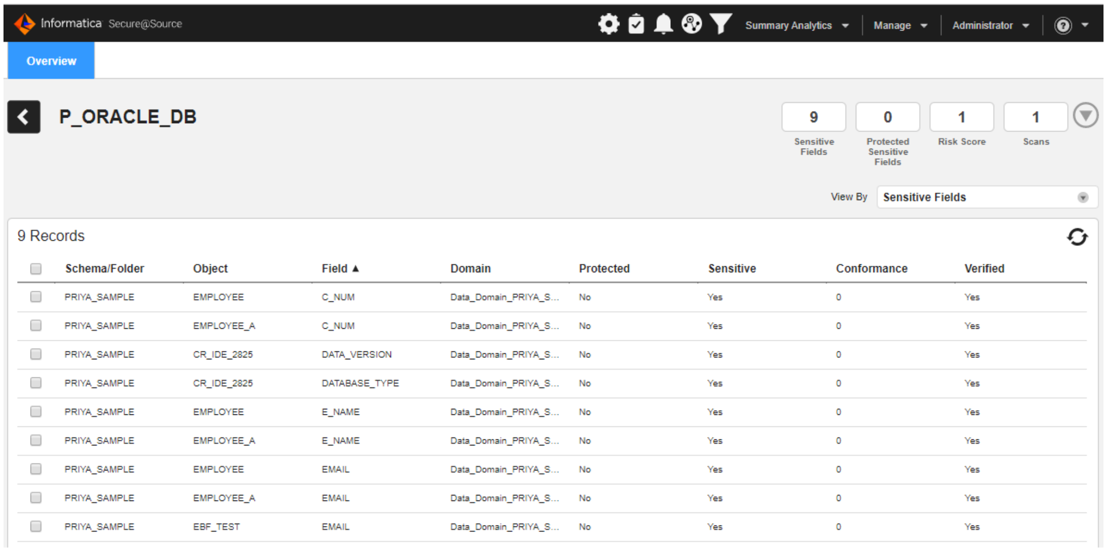
Processing Action: When rule is matched... Stop if Applied

For more information about Matchers and Actions, see https://docs.informatica.com/data-security-group/dynamic-data-masking/9-9-1/user-guide/connection-rules.html.

## *6. Identify Sensitive Fields in the Secure@Source Data Store*

Because Secure@Source generates user activity event logs only for sensitive columns, you must run a scan job and identify sensitive fields against the corresponding data store that you configured in Dynamic Data Masking.

The following image shows a data store with sensitive fields identified:



P_ORACLE_DB

| 9 | 0 | 1 | 1 |
|---|---|---|---|
| Sensitive Fields | Protected Sensitive Fields | Risk Score | Scans |

View By: Sensitive Fields

9 Records

| □ | Schema/Folder | Object | Field ▲ | Domain | Protected | Sensitive | Conformance | Verified |
|---|---|---|---|---|---|---|---|---|
| □ | PRIYA_SAMPLE | EMPLOYEE | C_NUM | Data_Domain_PRIYA_S... | No | Yes | 0 | Yes |
| □ | PRIYA_SAMPLE | EMPLOYEE_A | C_NUM | Data_Domain_PRIYA_S... | No | Yes | 0 | Yes |
| □ | PRIYA_SAMPLE | CR_IDE_2825 | DATA_VERSION | Data_Domain_PRIYA_S... | No | Yes | 0 | Yes |
| □ | PRIYA_SAMPLE | CR_IDE_2825 | DATABASE_TYPE | Data_Domain_PRIYA_S... | No | Yes | 0 | Yes |
| □ | PRIYA_SAMPLE | EMPLOYEE | E_NAME | Data_Domain_PRIYA_S... | No | Yes | 0 | Yes |
| □ | PRIYA_SAMPLE | EMPLOYEE_A | E_NAME | Data_Domain_PRIYA_S... | No | Yes | 0 | Yes |
| □ | PRIYA_SAMPLE | EMPLOYEE | EMAIL | Data_Domain_PRIYA_S... | No | Yes | 0 | Yes |
| □ | PRIYA_SAMPLE | EMPLOYEE_A | EMAIL | Data_Domain_PRIYA_S... | No | Yes | 0 | Yes |
| □ | PRIYA_SAMPLE | EBF_TEST | EMAIL | Data_Domain_PRIYA_S... | No | Yes | 0 | Yes |

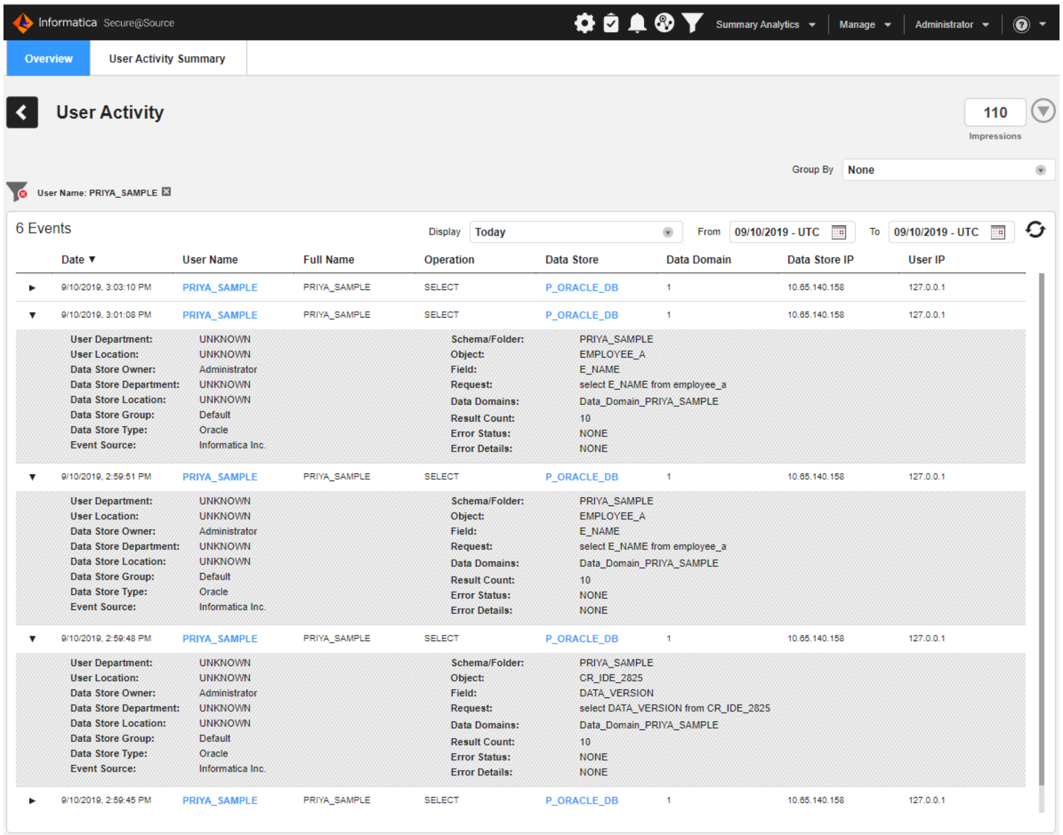
# Generating User Activity

To generate user activity events in Secure@Source, connect to the database node that you configured in Dynamic Data Masking using any client through the Dynamic Data Masking connection. Run database queries on the tables with sensitive fields.

1. Log in to Secure@Source.
2. From the **Overview** workspace, click the title of the User Activity indicator.

   The **User Activity** page appears.
3. View the user event details.

The following image shows example user event details:



User Activity

110
Impressions

Group By: None

User Name: PRIYA_SAMPLE

6 Events

Display: Today
From: 09/10/2019 - UTC
To: 09/10/2019 - UTC

| | Date ▼ | User Name | Full Name | Operation | Data Store | Data Domain | Data Store IP | User IP |
|---|---|---|---|---|---|---|---|---|
| ▶ | 9/10/2019, 3:03:10 PM | PRIYA_SAMPLE | PRIYA_SAMPLE | SELECT | P_ORACLE_DB | 1 | 10.65.140.158 | 127.0.0.1 |
| ▼ | 9/10/2019, 3:01:08 PM | PRIYA_SAMPLE | PRIYA_SAMPLE | SELECT | P_ORACLE_DB | 1 | 10.65.140.158 | 127.0.0.1 |

| | | | |
|---|---|---|---|
| **User Department:** | UNKNOWN | **Schema/Folder:** | PRIYA_SAMPLE |
| **User Location:** | UNKNOWN | **Object:** | EMPLOYEE_A |
| **Data Store Owner:** | Administrator | **Field:** | E_NAME |
| **Data Store Department:** | UNKNOWN | **Request:** | select E_NAME from employee_a |
| **Data Store Location:** | UNKNOWN | **Data Domains:** | Data_Domain_PRIYA_SAMPLE |
| **Data Store Group:** | Default | **Result Count:** | 10 |
| **Data Store Type:** | Oracle | **Error Status:** | NONE |
| **Event Source:** | Informatica Inc. | **Error Details:** | NONE |

| | Date ▼ | User Name | Full Name | Operation | Data Store | Data Domain | Data Store IP | User IP |
|---|---|---|---|---|---|---|---|---|
| ▼ | 9/10/2019, 2:59:51 PM | PRIYA_SAMPLE | PRIYA_SAMPLE | SELECT | P_ORACLE_DB | 1 | 10.65.140.158 | 127.0.0.1 |

| | | | |
|---|---|---|---|
| **User Department:** | UNKNOWN | **Schema/Folder:** | PRIYA_SAMPLE |
| **User Location:** | UNKNOWN | **Object:** | EMPLOYEE_A |
| **Data Store Owner:** | Administrator | **Field:** | E_NAME |
| **Data Store Department:** | UNKNOWN | **Request:** | select E_NAME from employee_a |
| **Data Store Location:** | UNKNOWN | **Data Domains:** | Data_Domain_PRIYA_SAMPLE |
| **Data Store Group:** | Default | **Result Count:** | 10 |
| **Data Store Type:** | Oracle | **Error Status:** | NONE |
| **Event Source:** | Informatica Inc. | **Error Details:** | NONE |

| | Date ▼ | User Name | Full Name | Operation | Data Store | Data Domain | Data Store IP | User IP |
|---|---|---|---|---|---|---|---|---|
| ▼ | 9/10/2019, 2:59:48 PM | PRIYA_SAMPLE | PRIYA_SAMPLE | SELECT | P_ORACLE_DB | 1 | 10.65.140.158 | 127.0.0.1 |

| | | | |
|---|---|---|---|
| **User Department:** | UNKNOWN | **Schema/Folder:** | PRIYA_SAMPLE |
| **User Location:** | UNKNOWN | **Object:** | CR_IDE_2825 |
| **Data Store Owner:** | Administrator | **Field:** | DATA_VERSION |
| **Data Store Department:** | UNKNOWN | **Request:** | select DATA_VERSION from CR_IDE_2825 |
| **Data Store Location:** | UNKNOWN | **Data Domains:** | Data_Domain_PRIYA_SAMPLE |
| **Data Store Group:** | Default | **Result Count:** | 10 |
| **Data Store Type:** | Oracle | **Error Status:** | NONE |
| **Event Source:** | Informatica Inc. | **Error Details:** | NONE |

| | Date ▼ | User Name | Full Name | Operation | Data Store | Data Domain | Data Store IP | User IP |
|---|---|---|---|---|---|---|---|---|
| ▶ | 9/10/2019, 2:59:45 PM | PRIYA_SAMPLE | PRIYA_SAMPLE | SELECT | P_ORACLE_DB | 1 | 10.65.140.158 | 127.0.0.1 |

The user activity events shown in the example are generated through Dynamic Data Masking as defined by the CEF format.

4. You can configure and generate user activity event logs with any attribute presented in the Secure@Source supported CEF format.

   **Note:**

   - The database user name captured in the AUTH_USERNAME symbol must match one of the Secure@Source user names. If the database user name does not match, import database users into the Secure@Source user group.
   - Dynamic Data Masking version 9.9.1 does not support the **AffectedRows** attribute in the CEF format and does not populate this value. The **AffectedRows** value is specified in the CEF format as a fixed value.
   - User activity events are not generated for queries against tables that do not have sensitive fields identified.
   - Symbols used in CEF format are case-sensitive and must be in uppercase.

# Authors

**Siva Krapa**

**Julie Henry**

## Acknowledgements

**Thanks to the Dynamic Data Masking development team for their help in completing this article.**