Protegrity Analytics Guide 9.1.0.5

Created on: Nov 19, 2024

Notice

Copyright

Table of Contents

Copyright **2**

Chapter 1 Introduction to This Guide **6**
1.1 Sections contained in this Guide 6
1.2 Accessing the Protegrity documentation suite 6

Chapter 2 Introduction to Protegrity Analytics **7**

Chapter 3 Enabling Protegrity Analytics **8**
3.1 Using Protegrity Analytics 8
3.2 Connecting to the Audit Store 8
3.3 Verifying Services 9

Chapter 4 Viewing Dashboards **10**
4.1 Viewing the Overview Dashboard 10
4.1.1 Viewing the General Status 11
4.1.2 Viewing the Operation Trend 13
4.1.3 Viewing the Top 10 Active Users 13
4.1.4 Viewing the Top 10 Data Elements Used 14
4.2 Viewing Metering Information 14
4.3 Viewing the Surprise Dashboard 16
4.3.1 Viewing the Heat Map 17
4.3.2 Viewing the Surprise Line Graph 20

Chapter 5 Working with Alerts **22**
5.1 Viewing Alerts 22
5.2 Working with Destinations 24
5.2.1 Creating Destinations 24
5.2.2 Modifying Destinations 26
5.3 Working with Monitors 28
5.3.1 Creating a Monitor 28
5.3.1.1 Configuring the Monitor 28
5.3.1.2 Configuring the Trigger 30
5.3.1.3 Configuring the Action 31
5.3.2 Modifying Monitors 33

Chapter 6 Working with Forensics **36**
6.1 Viewing Logs 37
6.1.1 Fields Logged 38
6.1.2 Querying Logs 41
6.1.3 Working with Saved Queries 43
6.1.4 Exporting Logs to an External File 46

Chapter 7 Viewing Reports **49**
7.1 Viewing Policy Reports 49
7.2 Working with Reports 51
7.2.1 Viewing Reports 51
7.2.2 Creating a Report 53
7.2.3 Editing a Report 54
7.2.4 Saving the Report Results 55

7.2.5 Example Queries ... 56

Chapter 8 Information Lifecycle Management (ILM) ... 60
8.1 Exporting Logs ... 61
8.2 Importing Logs ... 62
8.3 Deleting Indexes ... 63

Chapter 9 Using the Scheduler ... 65
9.1 Creating a Scheduled Task ... 66
9.2 Working with Scheduled Tasks ... 71
9.3 Viewing the Scheduler Monitor ... 72
9.4 Viewing the Scheduler Logs ... 73

Chapter 10 Verifying Signatures ... 75
10.1 Working with Signatures ... 78
10.2 Creating a Signature Verification Job ... 79
10.3 Editing a Signature Verification Job ... 81
10.4 Viewing Signature Verification Logs ... 83

Appendix 11 Troubleshooting ... 84
11.1 Known Issues for Protegrity Analytics ... 84

Appendix 12 Basics of Audit Store Querying ... 88
12.1 Creating Extraction Query Scripts ... 88
12.1.1 Using Variables ... 89
12.2 Creating Triggers ... 90
12.2.1 Basics for Creating Trigger Condition Scripts ... 90
12.2.2 Building the Trigger ... 91
12.3 Creating Alert Messages ... 91
12.3.1 Coding Basics for Alert Messages ... 91
12.3.2 Building the Alert ... 92
12.4 Sample Queries for Building Monitors ... 93
12.4.1 Sample: Alert For Unauthorized Access ... 93
12.4.2 Sample: Alert When A Particular User Performs a Large Number of Operations ... 94
12.4.3 Sample: Alert When Requests are Raised at Odd Hours ... 95
12.4.4 Sample: Alert for User Access from Multiple Nodes ... 96
12.4.5 Sample: Alert for Signature Verification Failures ... 97

Appendix 13 Removing Analytics from the Appliance ... 99

Chapter 1

Introduction to This Guide

1.1 Sections contained in this Guide
1.2 Accessing the Protegrity documentation suite

This document provides information about Protegrity Analytics. It describes the features of Analytics for viewing log data from the Audit Store. It also explains the settings for working with the components of Analytics.

1.1 Sections contained in this Guide

The guide is broadly divided into the following sections

- Section *Introduction to This Guide* defines the purpose and scope for this guide. In addition, it explains how information is organized in this guide.
- Section *Introduction to Protegrity Analytics* describes an overview about Protegrity Analytics.
- Section *Enabling Protegrity Analytics* lists the steps for setting up Protegrity Analytics. It also details the steps for creating and connecting to an Audit Store cluster.
- Section *Viewing Dashboards* describes the various dashboards available for monitoring logs. It also describes the steps for viewing metering information and the surprise dashboard.
- Section *Working with Alerts* describes the steps for working with alerts, monitors, and destinations.
- Section *Working with Forensics* explains the Forensics screen. It also explains the features for querying the Audit Store logs.
- Section *Viewing Reports* explains the Reporting feature. It also lists the steps for creating reports by querying the Audit Store.
- Section *Information Lifecycle Management (ILM)* explains the steps for exporting, importing, and migrating logs and deleting log indexes.
- Section *Using the Scheduler* lists the steps for creating schedules to perform repetitive tasks for reporting, ILM, the Forensics Autocomplete Index, and the signature verification feature.
- Section *Verifying Signatures* explains the Signature Verification feature. It also lists the steps for working with signature verification jobs.
- Section *Troubleshooting* describes the steps for troubleshooting issues related to Protegrity Analytics.
- Section *Basics of Audit Store Querying* provides basic information about the code for creating monitors. It also contains sample code for building monitors.
- Section *Removing Analytics from the Appliance* provides the steps for removing Analytics from the Appliance.

1.2 Accessing the Protegrity documentation suite

This section describes the methods to access the *Protegrity Documentation Suite* using the *My.Protegrity* portal.

Chapter 2

Introduction to Protegrity Analytics

Protegrity Analytics installed on the ESA provides the interface for viewing data from the Audit Store. The dashboards in Protegrity Analytics display information using graphs and charts for a quick understanding of the protect, unprotect, and reportect transactions performed. The surprise dashboard performs analysis on the Audit Store data and helps compare actual transactions with estimated transactions.

Forensics, reports, and alerts that were earlier available in the ESA are now available in Protegrity Analytics. Forensics can be queried to view the required data from the Audit Store. Reports can be customized and data from the reports downloaded as per audit requirements. Monitors can be added using Protegrity Analytics to raise alerts to monitor the systems.

Protegrity Analytics also has a scheduler to automate and perform repetitive tasks related to Forensics, archiving logs, and reporting. The Information Lifecycle Management (ILM) feature allows exporting and importing of logs for maintenance.

The various features provided by Protegrity Analytics are described in this guide.

Chapter 3

Enabling Protegrity Analytics

3.1 Using Protegrity Analytics
3.2 Connecting to the Audit Store
3.3 Verifying Services

Protegrity Analytics provides a set of tools for analyzing the data in the Audit Store. The Protegrity Analytics component provides various reports, graphs, and tables for viewing the Audit Store data. Protegrity Analytics is available on your system when you set up the ESA. However, you must set up a new Audit Store cluster or connect to an existing Audit Store cluster before you can view the Analytics.

3.1 Using Protegrity Analytics

The Protegrity Analytics component is installed on your system when you install the ESA.

For more information about installing the ESA, refer to the section *Installing ESA On-Premise* in the *Protegrity Installation Guide 9.1.0.5*.

> **Note:** After installation, ensure that the system time on all the machines in the Audit Store clusters are in sync. This is required for the Scheduler to work accurately. If required, select **Use ESA's NTP** to synchronize the system time of the node with the ESA.
>
> For more information about NTP, refer to the section *Setting Date and Time* in the *Protegrity Appliances Overview Guide 9.1.0.5*.

3.2 Connecting to the Audit Store

Protegrity Analytics queries the Audit Store to obtain data for the various reports. After you install the ESA, you need to complete configurations for reading the audit data. The Audit Store might be on the local ESA, in this case you need to start an Audit Store cluster on the ESA. Other nodes can join this Audit Store cluster for accessing the Audit Store data. If the Audit Store is started on a remote node, then connect to the Audit Store cluster on the remote node. Complete one of the following tasks as per your Audit Store installation location.

> **Note:** The *viewer* user or a user with the *viewer* role permission can only view cluster information. You need to log in using the *admin* role to create or join the Audit Store cluster.

> **Caution:** For creating the Audit Store Cluster, refer to the section *Configuring the Audit Store Cluster* in the *Protegrity Installation Guide 9.1.0.5*.

3.3 Verifying Services

After you have completed the installation of the ESA and created a new cluster, navigate to the services section on the ESA and verify that your services are present.

Complete the following steps to check the services.

1. On the ESA Web UI, navigate to **System** > **Services**.
2. Verify that the following service is installed.
   - **Misc**
     - Analytics

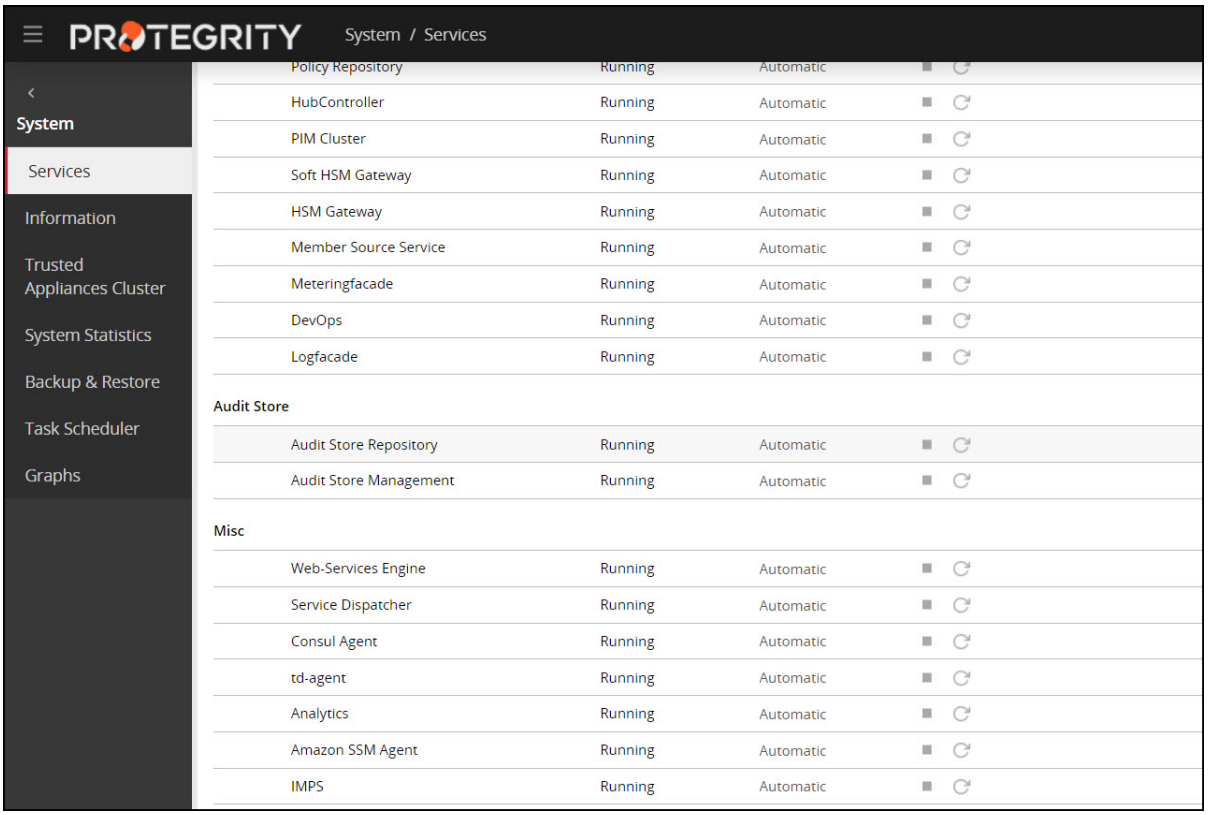

Figure 3-1: Analytics Service

Chapter 4

Viewing Dashboards

4.1 Viewing the Overview Dashboard
4.2 Viewing Metering Information
4.3 Viewing the Surprise Dashboard

The Dashboard displays different graphs and tables. You can use the data provided on the different dashboards to view and analyze the flow and working of your organizational data.

> **Note:** The data that appears in the various graphs in dashboard depends on the audit data and the protector configuration. Hence, if audits are turned off or aggregation is turned on at the protector level, then the dashboard might not show the correct data.
>
> Only logs for Protectors version 9.0.0.0 and above are considered for displaying the data on the various dashboards.

4.1 Viewing the Overview Dashboard

The Overview Dashboard is divided into four sections that show the top 10 data elements, top 10 users, operation graph, and general stats. To view the Overview, from Analytics, navigate to **Dashboard** > **Overview**.

> **Note:** Only logs for Protectors version 9.0.0.0 and above are considered for displaying the data on the dashboard.

The Overview page is shown in the following figure.

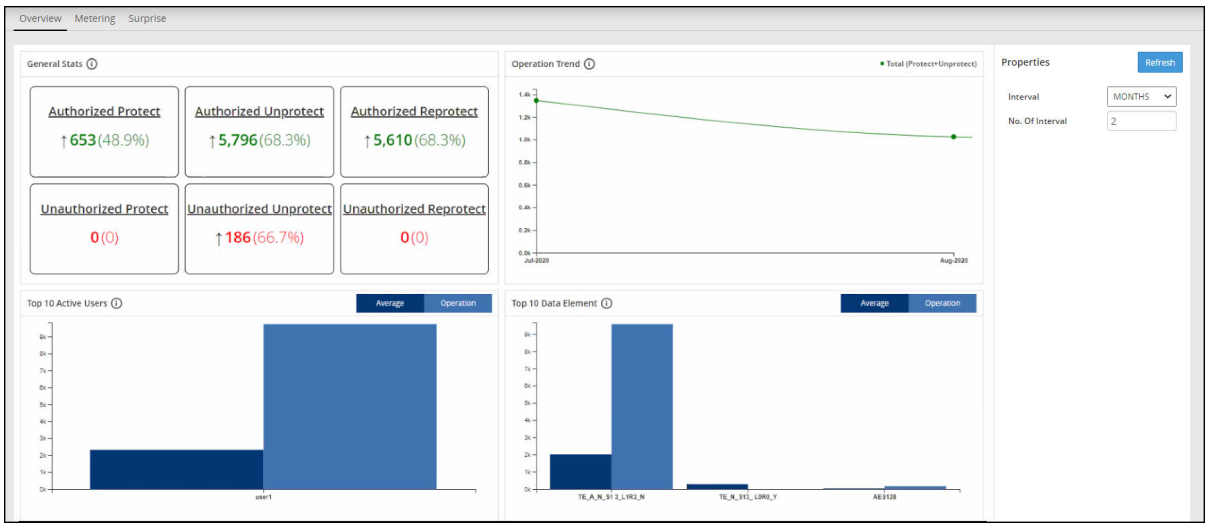

Figure 4-1: Overview Page

The following options are available on this page:

- **Refresh**: Click the button to refresh the data on this page.
- **Interval**: Select the aggregation for your analysis. The available intervals are:
  - Months
  - Weeks
  - Days
  - Hours
- **No. Of Interval**: Select the number of interval values to aggregate for displaying on the dashboard.

4.1.1 Viewing the General Status

The *General Status* section shows the number of successful and unsuccessful protect, unprotect, and reprotect operations for the specified interval. You can use this status to obtain a general overview about the operations performed over the time period selected. The *General Status* section is shown in the following figure.

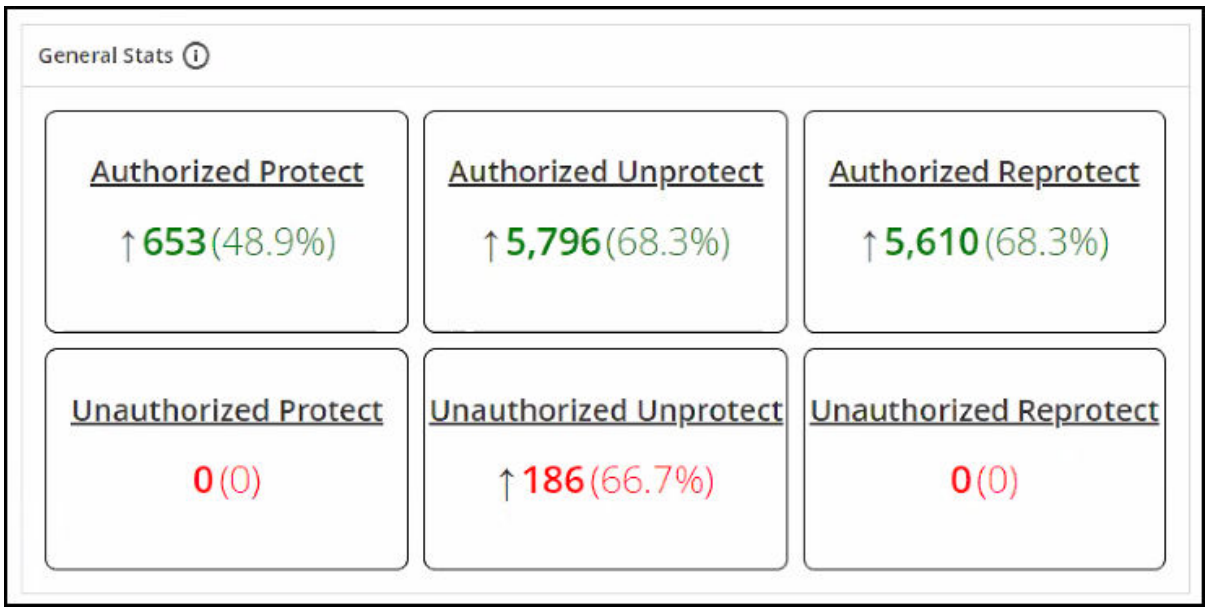

Figure 4-2: General Status

The following information appears in this section:

- Authorized and unauthorized transactions.
  - Green color is used for authorized operations.
  - Red color is used for unauthorized operations.
- Increase and decrease in operations:
  - Up arrow is used to show that operations have increased, where the current interval has more transactions than the earlier interval.
  - Down arrow is used to show that operations have decreased, where the current interval has fewer transactions than the earlier interval.
- Number of operations:
  - The number shows the number of operations as per the aggregation and the number of interval selected.
  - The number in brackets shows the percentage difference in the total number of operations compared with the earlier time interval.

    > **Note:** If the current interval count is 0, then the number of operations of the earlier interval is shown instead of the percentage.

The entries with the following audit codes and respective operation names are aggregated for obtaining the values displayed here:

- Authorized Protect: 6, 10, 34, and 36.
- Authorized Unprotect: 8, 10, and 11.
- Authorized Reprotect: 50.
- Unauthorized Protect: 1, 2, 3, 7, 25, 35, and 37.
- Unauthorized Unprotect: 1, 2, 3, 9, and 25.
- Unauthorized Reprotect: 1, 2, 3, 7, 9, 25, 35, and 37.

4.1.2 Viewing the Operation Trend

The *Operation Trend* section shows the sum of successful protect and unprotect operations for the specified interval by using a line graph. You can use this graph to understand the quantum of operations performed over the time period selected. The Operation Trend graph is shown in the following figure.

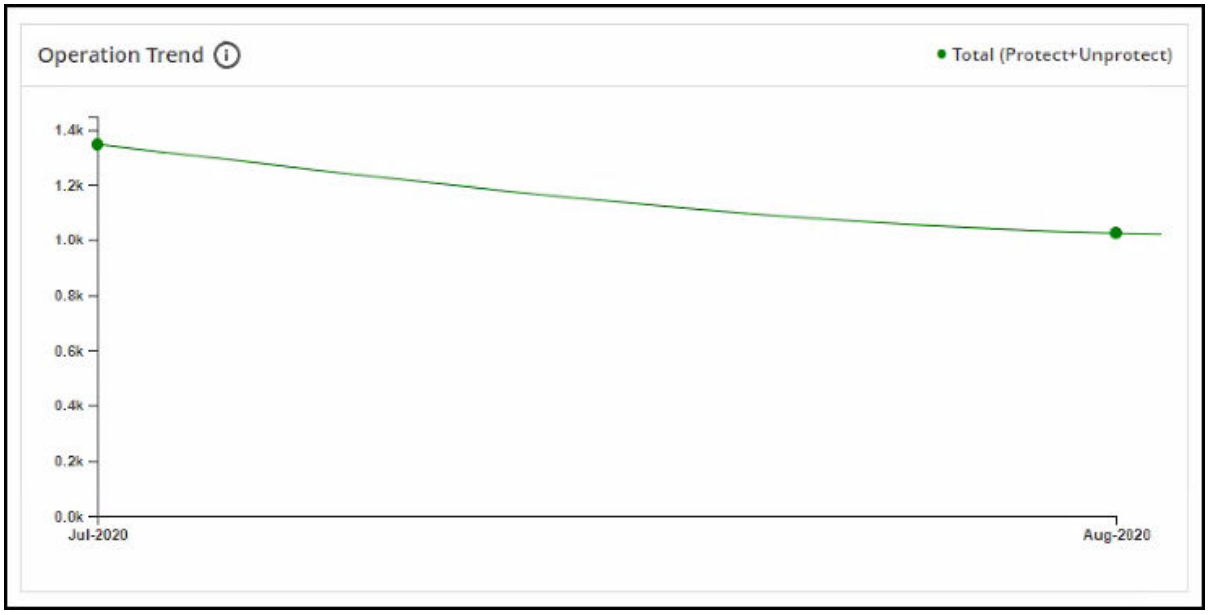

Figure 4-3: Operation Trend

4.1.3 Viewing the Top 10 Active Users

The *Top 10 Active Users* section shows a bar graph of the top 10 users on the basis of transactions performed during the interval selected. You can use this graph to compare and view the estimated transactions usage and the actual usage for the top 10 users. The graph is shown in the following figure.

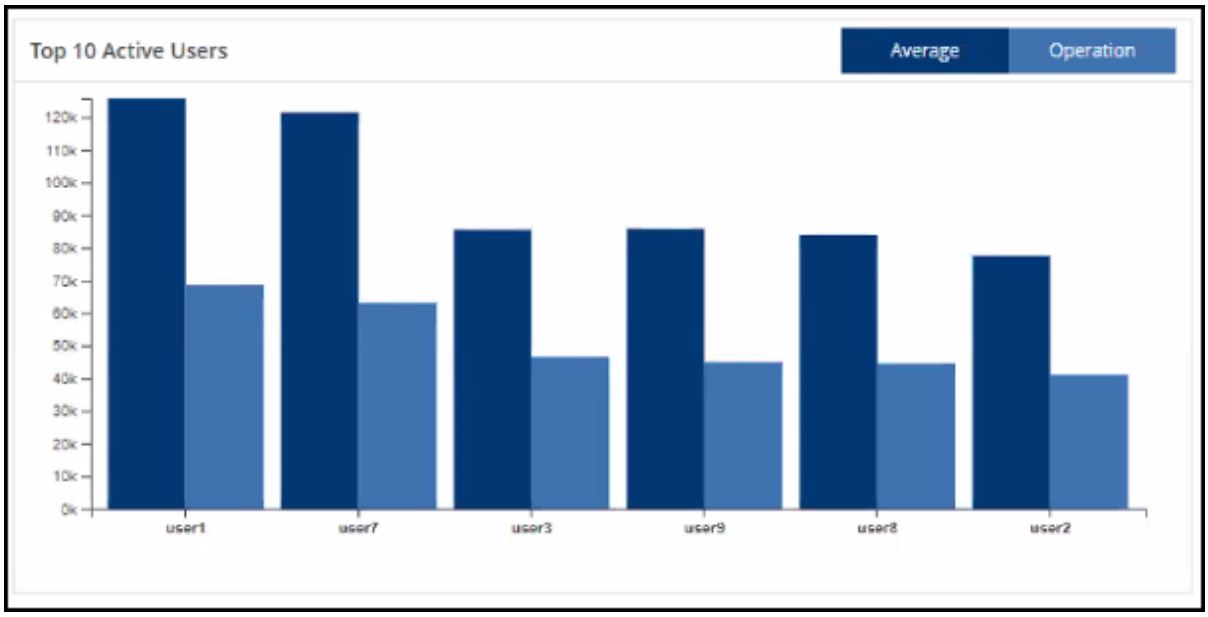

Figure 4-4: Top 10 Active Users

The following information about the top 10 active users is shown in the bar graph:

- The username is shown on the X-axis.
- The number of transactions is shown on the Y-axis.
- The light blue bar shows the number of transactions for the data element for the current selected interval.
- The dark blue bar shows the average number of transactions for the selected number of interval, except the current interval.

For example if the current month is June, and the number of interval selected is 6, then the average transactions for the months January to May would be averaged and shown in the dark blue bar. The light blue bar would show the transactions for June.

4.1.4 Viewing the Top 10 Data Elements Used

The *Top 10 Data Elements* section shows a bar graph of the top 10 data elements that have been used during the interval selected. You can use this graph to compare and view the estimated data element usage and the actual usage. The graph is shown in the following figure.

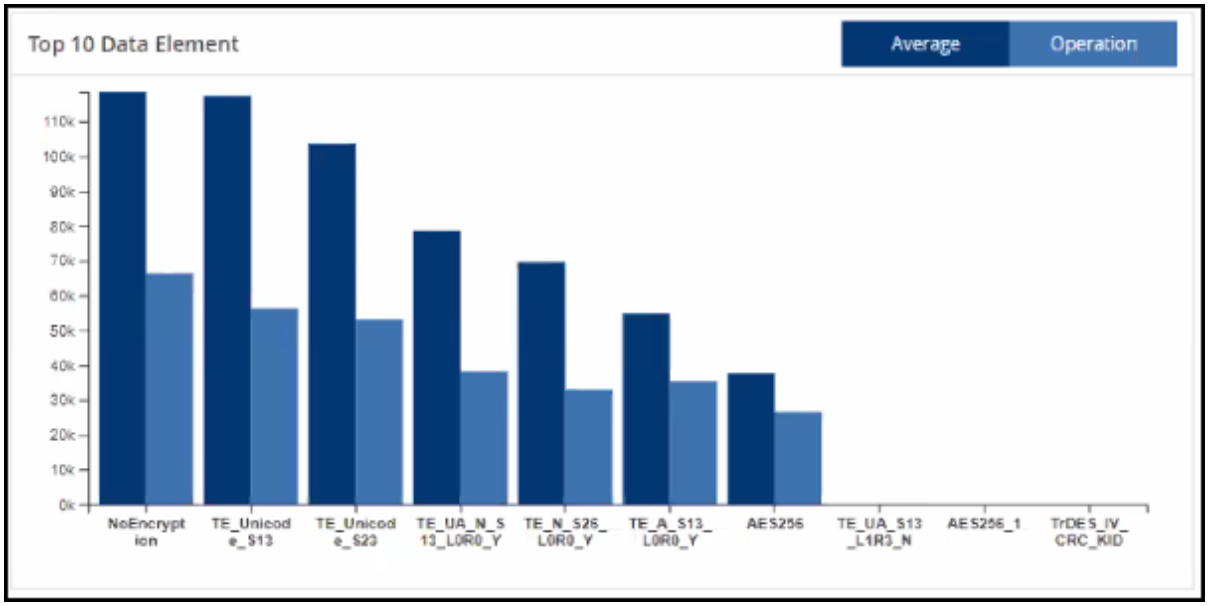

Figure 4-5: Top 10 Data Elements

The following information about the top 10 data elements is shown in the bar graph:

- The data element is shown on the X-axis.
- The number of transactions is shown on the Y-axis.
- The light blue bar shows the number of transactions for the data element for the current selected interval.
- The dark blue bar shows the average number of transactions for the selected number of interval, except the current interval.

For example if the current month is June, and the number of interval selected is 6, then the average transactions for the months January to May would be averaged and shown in the dark blue bar. The light blue bar would show the transactions for June.

4.2 Viewing Metering Information

The Metering report shows the consolidated metering data according to the metering UID. It shows the protect, unprotect, and reprotect operations processed over different periods of time in a bar graph and a table. This gives you a visual representation of the frequency of operations carried out. This report displays the information that you can retrieve from the Audit Store using the filter *logtype = "Metering"*. The filter shows the cumulative values for the metering data. However, this report shows the actual number of operations, and not the cumulative value, for the time period specified.

You can view the Metering screen by logging in to the ESA Web UI and navigating to **Analytics** > **Dashboard** > **Metering**. The **Metering** screen appears.

> **Note:** Only logs for Protectors version 9.0.0.0 and above are considered for displaying the data on the dashboard.

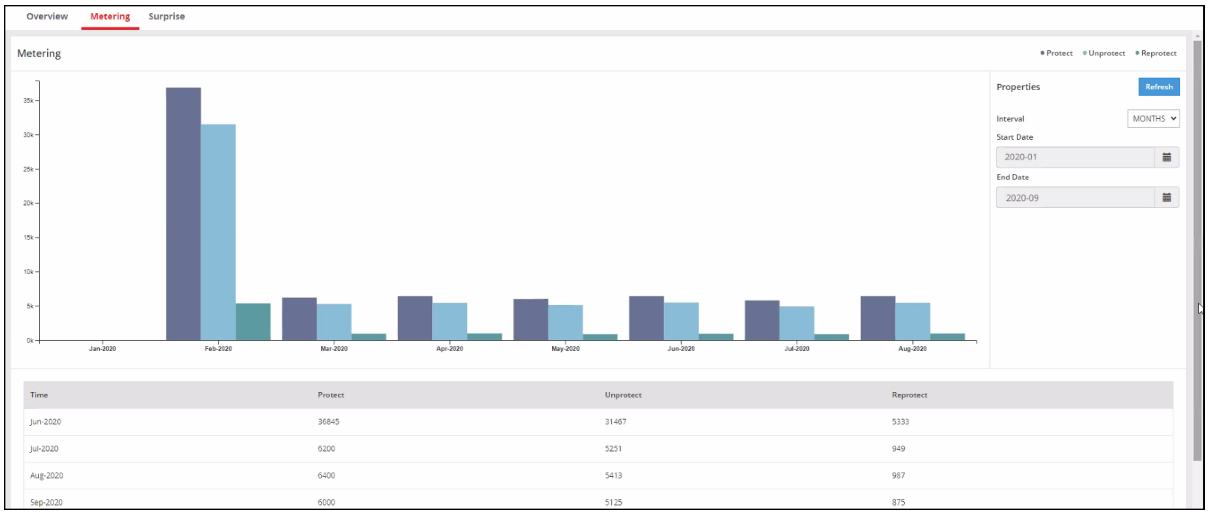

Figure 4-6: Metering Screen

You can use the following options for displaying data in the graph:

- **Refresh**: Click this button to obtain and refresh the graph with the latest data that is available.
- **Interval**: Specify the interval for aggregating the data. The following options are available:
  - **MONTHS**: The data is aggregated and displayed using the month name and year.
  - **WEEKS**: The data is aggregated on a weekly basis with Monday as the start of the week.
  - **DAYS**: The data is aggregated on a day-to-day basis.
  - **HOURS**: The data is aggregated on an hourly basis.
- **Start Date**: Specify the start date for displaying data in the graph.
- **End Date**: Specify the end date for displaying data in the graph.

> **Note:** The end date is non-inclusive.

The time period is shown along the X-axis and the number of transactions are displayed along the Y-axis. You can hover over a bar to view the exact number of transactions performed.

The following legend is used to display the information in the graph:

- Protect operations are displayed in grey.
- Unprotect operations are displayed in light blue.
- Reprotect operations are displayed in green.

The table following the graph displays the source data that is used to display the graph.

> **Note:** Only rows that have a protect, unprotect, or reprotect value are displayed in the table.

4.3 Viewing the Surprise Dashboard

The Surprise dashboard helps you analyze the operations (protect, unprotect, and reprotect) carried out. The number of operations are compared with different moving average models and shown on a heat map.

The heat map is built using technical analysis with the help of the following moving average models:

- Simple
- Linear
- Exponential weighted moving average (EWMA)
- Double exponential (HOLT)

> **Note:** The operations are grouped and used as a data point for calculating the results in this map. This map does not show individual operations. It also does not show whether the operation was valid or successful. It just considers all the operations processed.
>
> Only logs for Protectors version 9.0.0.0 and above are considered for displaying the data on the dashboard.

The following sample graph can be used to understand the heat map.

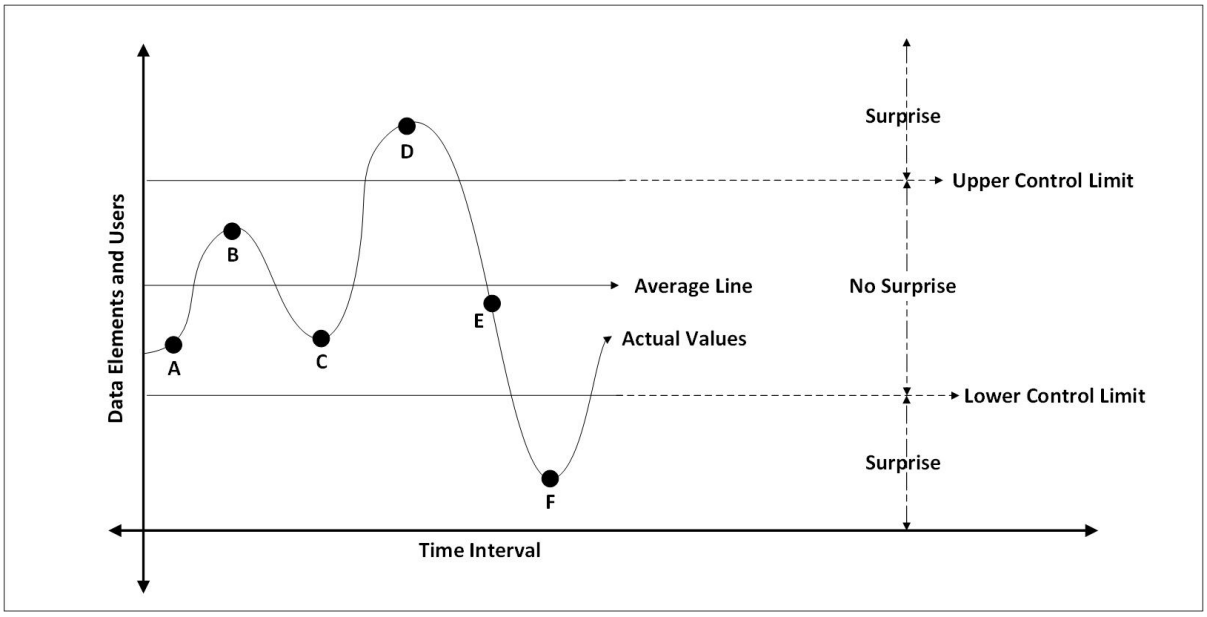

Figure 4-7: Heat Map

In the graph, the average line is calculated by using the weighted moving average. The surprise dashboard uses four different moving average models.

The actual values A, B, C, D, E, and F are plotted and displayed using a line on the graph.

The control limits are derived by performing technical analysis to obtain a value. That value is then added and subtracted to the moving average value to get the upper and lower control limits respectively.

The area above the upper control limit and below the lower control limit is the *Surprise* area. The area between the upper control limit and the lower control limit is the *No Surprise* area. In the graph, the points A, B, C, and E are within the control limits and are in the *No Surprise* area. The points D and F are in the *Surprise* area.

The operations for each unit of time are calculated using the moving average model selected and the data window value to derive the average line. The operations are mapped to this graph to create the heat map using various moving average models. The following models are supported in the Surprise dashboard:

- **Simple**: Calculate the average from the values. The following property is available for this model:
  - **Data Window**: Specify the number of previous data points to include for calculating the moving average.
- **Linear**: Calculate the weighted moving average from the values. The latest values have a higher weightage and importance in calculating the result. The following property is available for this model:
  - **Data Window**: Specify the number of previous data points to include for calculating the moving average.
- **Exponential weighted moving average (EWMA)**: Calculate the EWMA from the values. The latest values have an exponentially higher weightage and importance in calculating the result. It uses a variable named as alpha. The following properties are available for this model:
  - **Alpha**: Specify the value of alpha for calculating the moving average. Alpha is a fixed value between 0 and 1. The default value of alpha is 0.3.
  - **Data Window**: Specify the number of previous data points to include for calculating the moving average.
- **Double exponential (HOLT)**: Calculate the HOLT average from the values. The latest values have an exponentially higher weightage and importance in calculating the result. It uses two variables, alpha and beta. The following properties are available for this model:
  - **Alpha**: Specify the value of alpha for calculating the moving average. Alpha is a fixed value between 0 and 1. The default value of alpha is 0.3.
  - **Beta**: Specify the value of beta for calculating the moving average. Beta is a fixed value between 0 and 1. The default value of beta is 0.1.
  - **Data Window**: Specify the number of previous data points to include for calculating the moving average.

4.3.1 Viewing the Heat Map

The heat map appears using a grid of colored boxes that shows the surprises. You can view the heat map on the surprise dashboard by navigating to **Dashboard** > **Surprise**.

The heat map is shown in the following figure.

> **Note:** The earliest data, up to a maximum of two years, is displayed when the page is first opened.

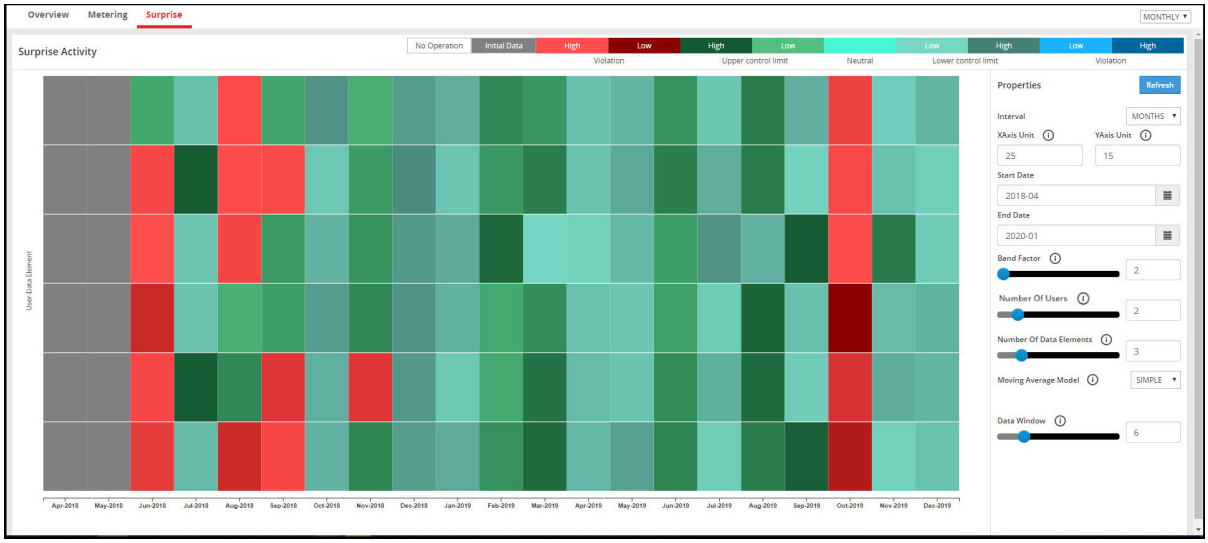

Figure 4-8: Heat Map

The surprise dashboard provides you with various options for customizing the heat map as properties. You can either specify the values you require in the text boxes or you can use the sliders provided. The properties are shown in the following figure.

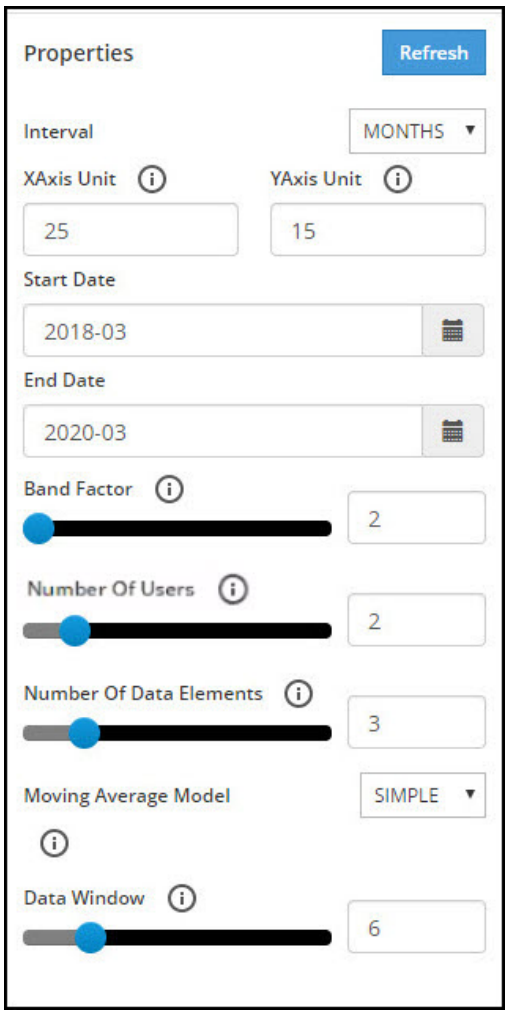

Figure 4-9: Properties

The following properties are available for working with the heat map:

- **Refresh**: Click this button to update the following data on the dashboard:
  - Number of users available in the slider
  - Number of data elements available in the slider
  - Data window
- **Interval**: Select the interval setting for your analysis. The available intervals are:
  - Months
  - Weeks
  - Days
  - Hours
- **X Axis Units**: Specify the number of blocks to display on the X-axis. You can increase the units to shrink the width of the blocks and show more blocks on the screen. Modify this interval to remove the horizontal scroll bar, if it appears.
- **Y Axis Units**: Specify the number of blocks to display on the Y-axis. You can increase the units to shrink the height of the blocks and show more blocks on the screen. Modify the number of users or data elements to remove the vertical scroll bar, if it appears.
- **Start Date**: Specify the start date for the graph data.

- **End Date**: Specify the end date for the graph data.

  > **Note:** The end date is non-inclusive.

- **Band Factor**: Select the multiplicative value for calculating the upper and lower control limits. The band factor value is used as a multiplicative value for deciding the upper control limit and the lower control limit. Increasing the band factor results in the widening of the no surprise area and decreasing the band factor results in the tightening of the no surprise area as shown in the figure in the section *Viewing the Surprise Dashboard*.
- **Number of Users**: Select the number of users to display in the graph.
- **Number of Data Elements**: Select the number of data elements to display in the graph.
- **Moving Average Model**: Select the moving average model for calculating the moving average. The available models are described in the section *Viewing the Surprise Dashboard*.

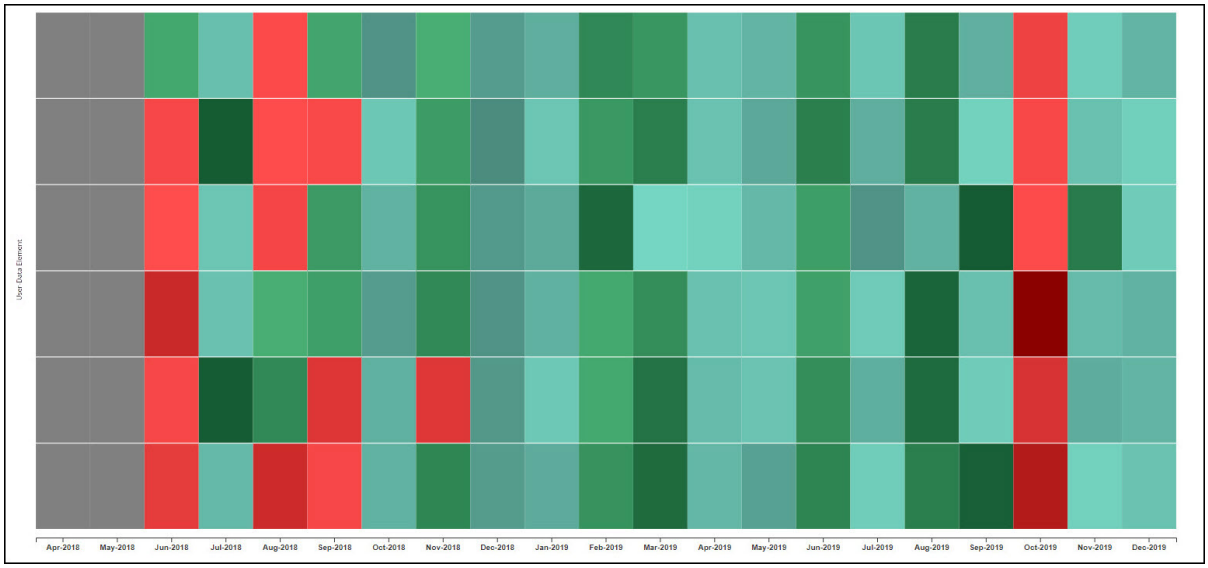

Figure 4-10: Heat Map

The legend is shown in the following figure:

Figure 4-11: Heat Map Legend

In the heat map, the time interval appears on the X-axis. A combination of the different users and data elements are shown on each row of the Y-axis. Hover the mouse over a block to view the block information. When you click a block, the surprise line graph for that row is displayed.

For more information about the surprise line graph, refer to the section *Viewing the Surprise Line Graph*.

The operations are analyzed using the various moving average models for the properties specified and the blocks are assigned the following colors:

- Bright green blocks: Operations that are on the average line. This is shown as *Neutral* in the legend.
- Red blocks: Operations that cross the upper control limit. Dark red blocks are used for operations near the upper control limit and the color becomes lighter and lighter as it moves away from the upper control limit. This is shown as high and low red *violations* in the legend.

- Blue blocks: Operations that cross the lower control limit. Light blue blocks are used for operations near the lower control limit and the color becomes darker and darker as it moves away from the lower control limit. This is shown as high and low blue *violations* in the legend.
- Dark green blocks: Operations that are between the average line and upper control limit. Dark green blocks are used for operations near the average line and the color becomes darker and darker as it moves towards the upper control limit. This is shown as high and low green *upper control limit* in the legend.
- Light green blocks: Operations that are between the average line and lower control limit. Light green blocks are used for operations near the average line and the color becomes and darker as it moves towards the lower control limit. This is shown as high and low green *lower control limit* in the legend.
- White blocks: No operations are present. This is shown as *No Operation* in the legend.
- Grey blocks: Starting operations that are used to calculate the moving average values. This is shown as *Initial Data* in the legend.

4.3.2 Viewing the Surprise Line Graph

The *Surprise Graph* is a line graph for each row in the heat map. Click a block on the heat map to view the surprise line graph for that row. You can view the data elements and user information in the legend. Click the **Back to HeatMap** button to return to the heat map. In this graph, you look for operations that are not in the grey area, these are surprises and need to be investigated.

The surprise graph is shown in the following figure.

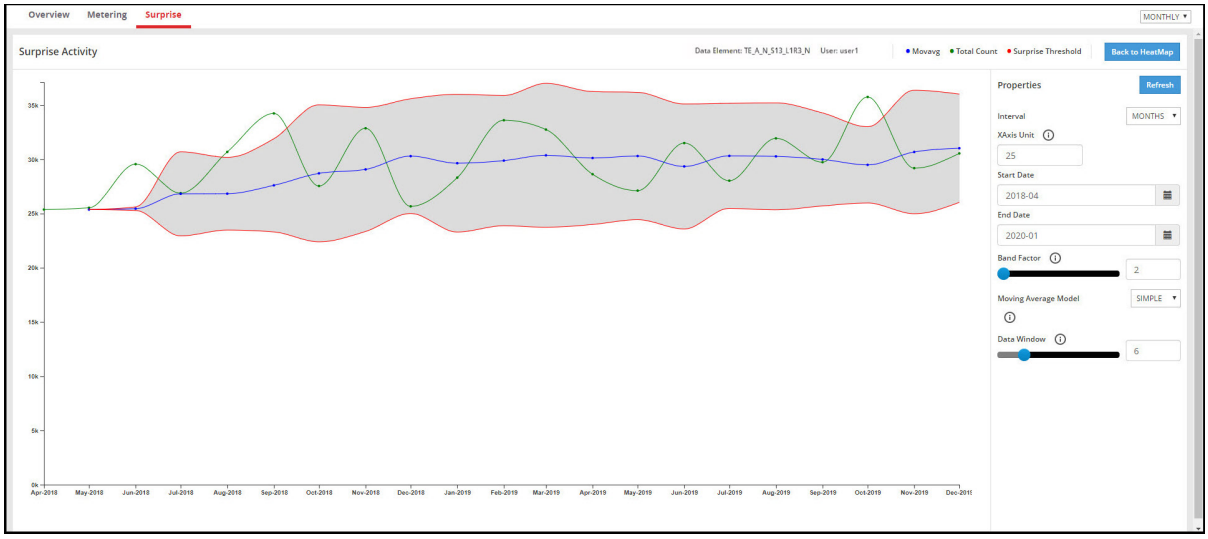

Figure 4-12: Surprise Graph

In the graph, the time interval appears on the X-axis. The number of operations for the user and data element are shown on the Y-axis. Hover the mouse over a point in the graph to view the values at that point. The following information appears in the graph:

- The moving average of the values are shown on the average line in blue.
- The upper control limit and the lower control limit are calculated and shown using red lines in the graph.
- The area between the upper control line and lower control line is colored in grey. This grey area is the *No Surprise* area.
- The number of operations is shown using a green line on the graph.
- Every time the green line goes over the upper control limit or below the lower control limit, it results in a surprise. These need to be investigated.

You can modify the various properties for widening or tightening the control limits and the averaging models used.

For more information about the moving average models and the properties, refer to the sections *Viewing the Surprise Dashboard* and *Viewing the Heat Map* respectively.

Chapter 5

Working with Alerts

5.1 Viewing Alerts
5.2 Working with Destinations
5.3 Working with Monitors

You can use alerting to keep a track of the different activities that take place on your system. The alerting ecosystem consists of the monitor, trigger, action, and destination. This system starts with the monitor. The monitor runs at regular intervals and keeps track of the activities taking place on a system. The information retrieved by the monitor is used as the input for a trigger. When the criteria defined in the trigger is fulfilled, the trigger is activated, which in turn calls an action. The action raises an alert using the destination that is defined, which can be a webhook or by informing users using email messages.

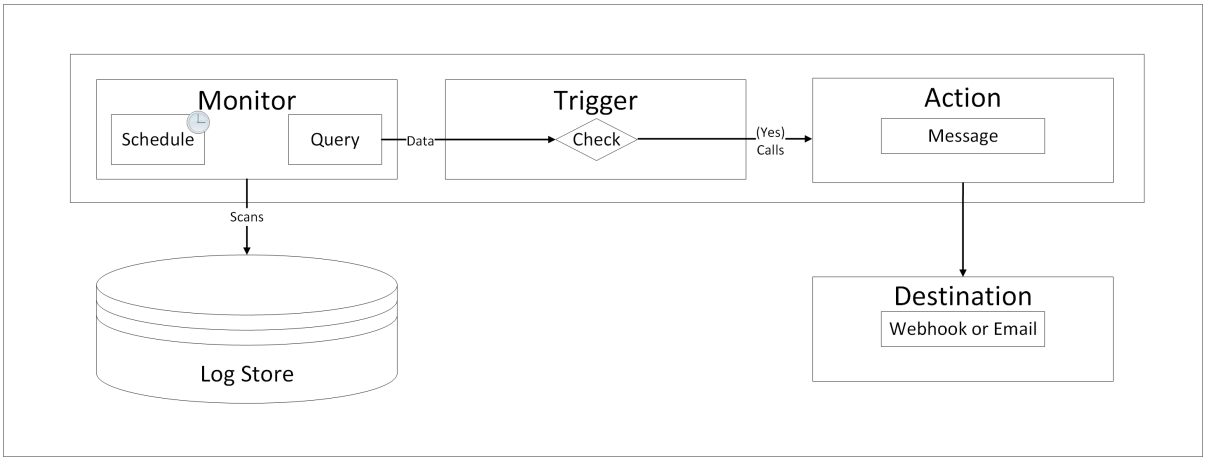

Figure 5-1: Alerting Ecosystem

5.1 Viewing Alerts

The alerts generated are displayed on the dashboard. You can access the dashboard from the Analytics screen by navigating to **Alerting** > **Alerts**. The dashboard is shown in the following figure.

> **Note:** The *viewer* role user or a user with the *viewer* role can only view alerts. You need admin rights to acknowledge alerts.

Figure 5-2: Viewing Alerts

The following fields are displayed in the alerts view:

- **Start Time**: The time when the alert was first raised.
- **Monitor Name**: The monitor that raised the alert.
- **Trigger Name**: The trigger that was activated for the alert.
- **Severity**: The severity level for the alert.
- **State**: The last state of the alert. The available states are:
  - Active: The alert is active and notifications are sent to the destination or the alert condition still persists.
  - Acknowledged: The alert is acknowledged, stop sending alert notifications to the destination.
  - Completed: The monitor completed processing without raising an alert condition or the alert was in an active state earlier and the alert condition does not exist anymore.
  - Error: An error occurred while processing an alert. This might be due to an error in the trigger condition or error in executing the action.
  - Deleted: A monitor or trigger is deleted.
- **Last Modified**: The time that the alert was received or the alert status was changed.
- **Action**: Click to update the state of an alert from **Active** to **Acknowledged**. You stop receiving notifications for active alerts that are acknowledged.

The lifecycle of an alert showing the different states is shown in the following figure.

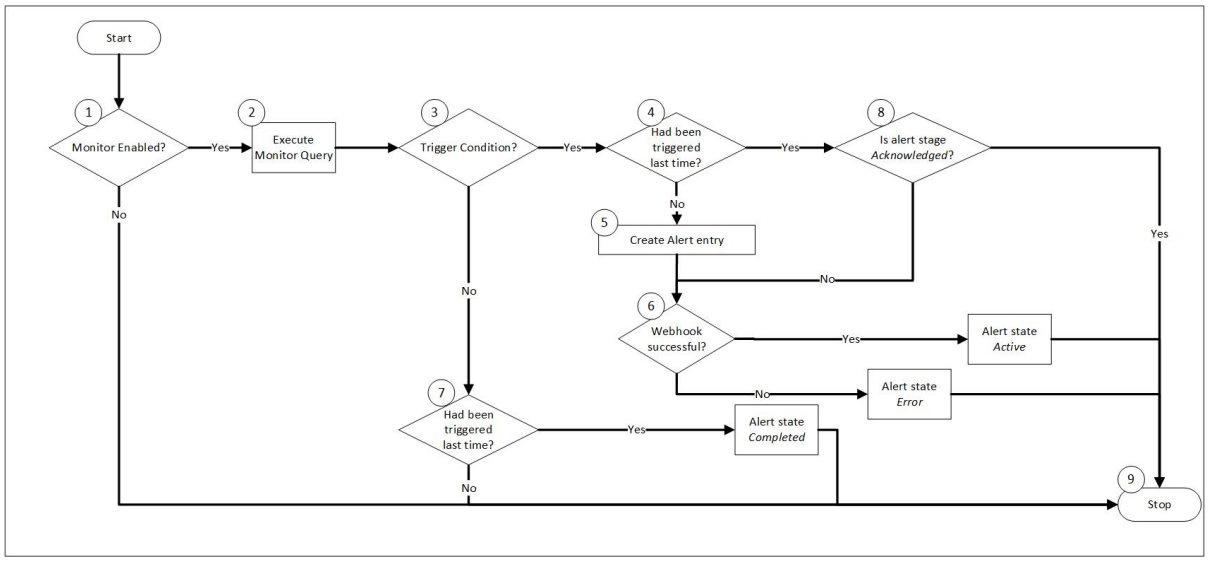

Figure 5-3: Alert Lifecycle

1. Is the monitor enabled?
   - If yes, then go to step 2.
   - If no, then go to step 9.
2. The monitor query is executed.

3. Is the trigger condition fulfilled?
   - If yes, then go to step 4.
   - If no, then go to step 7.
4. Was the trigger condition fulfilled earlier?
   - If yes, then go to step 8.
   - If no, then go to step 5.
5. Raise an alert.
6. Is the webhook successful?
   - If yes, then set the state as *Active* and go to step 9.
   - If no, then set the state as *Error* and go to step 9.
7. Was the trigger condition fulfilled earlier?
   - If yes, then set the state as *Completed* and go to step 9.
   - If no, then go to step 9.
8. Is the alert state *Acknowledged*?
   - If yes, then go to step 9.
   - If no, then go to step 6.
9. End of error flow.

5.2 Working with Destinations

A destination is a location that receives the alert message for an action. The destination can either be a webhook or an email. Destinations inform the users that the system meets certain requirements that need to be analyzed.

For example, a user can receive alerts over email when more than 10,000 unprotect operations have been carried out in 20 minutes.

> **Note:** The *viewer* role user or a user with the *viewer* role can only view destination information. You need admin rights to create or modify destinations.

5.2.1 Creating Destinations

A destination is a location for receiving the alert message. The destination can be a web location or an email address. In case the destination is set to email address then the system sends an email to the address specified. It does not monitor or provide an alert if the email address does not exist. You can define the destination from the **Destinations** screen.

1. From the Analytics screen, navigate to **Alerting** > **Destinations**.

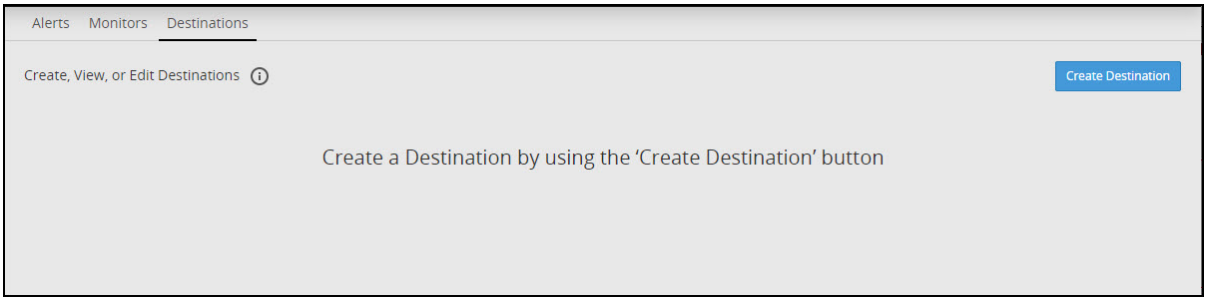

Figure 5-4: Destinations Screen

2. Click **Create Destination**.
   The **Create Destination** screen appears.

Figure 5-5: Create Destination Screen

3. Enter a unique destination name in the **Name** field.
4. Select one of the following options from the **Type** list and enter the required information:
   - Select **Webhook** to define a web location for receiving the alert.
     - **URL**: Specify a URL for receiving the alert. The URL must include the *http://* or *https://* protocol.
   - Select **Email** to define email addresses to receive a web location for receiving the alert.
     - **To**: Specify the email address for the **To** field.
     - **Cc**: Specify the email address for the **Cc** field.
     - **Bcc**: Specify the email address for the **Bcc** field.

> **Note:** You can specify multiple email addresses in the **To**, **Cc**, and **Bcc** fields. Separate the email addresses with a semi colon (;). At least one email address must be specified for creating the destination.

5. If you select the destination type as email, then click **Test** to test the SMTP connection.
   A test email message is sent to the email addresses specified in the fields on this page.

> **Note:** For the destination set as Email, ensure that SMTP is configured on all the nodes that are present in the Audit Store cluster by navigating to **Settings** > **Network** > **SMTP Settings** on each node.
>
> For more information about configuring SMTP, refer to the section *Setting Up the Email Server* in the *Protegrity Appliances Overview Guide 9.1.0.5*.

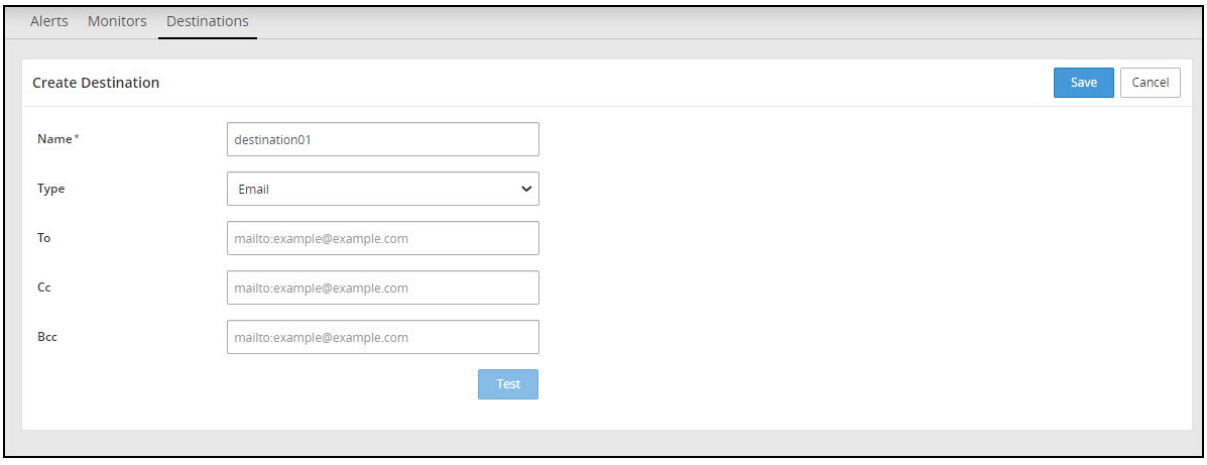

Figure 5-6: Test Destination

6. Click **Save** to save the destination.

The destination is created. You can now link the destination to a monitor.

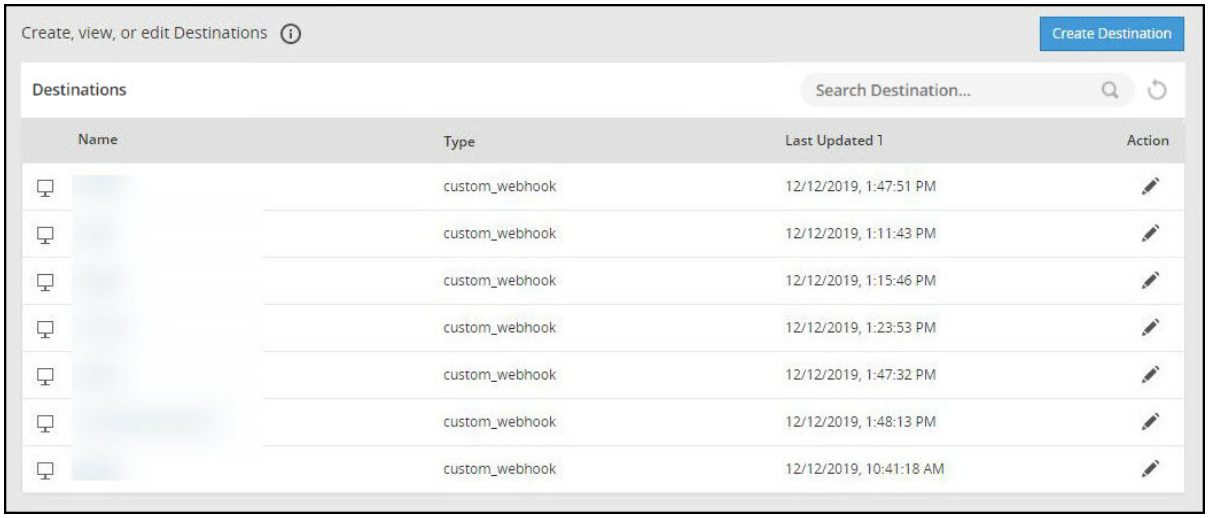

Figure 5-7: Destination Created

5.2.2 Modifying Destinations

You can view and manage the different destinations available from the **Destinations** screen. You can also delete destinations that are not linked to any monitor from this screen.

1. From the Analytics screen, navigate to **Alerting** > **Destinations**.

   The list of destinations created appears. You can use the search field to search for a specific destination by name from the list. Additionally, you can also use the wildcard character *while searching.

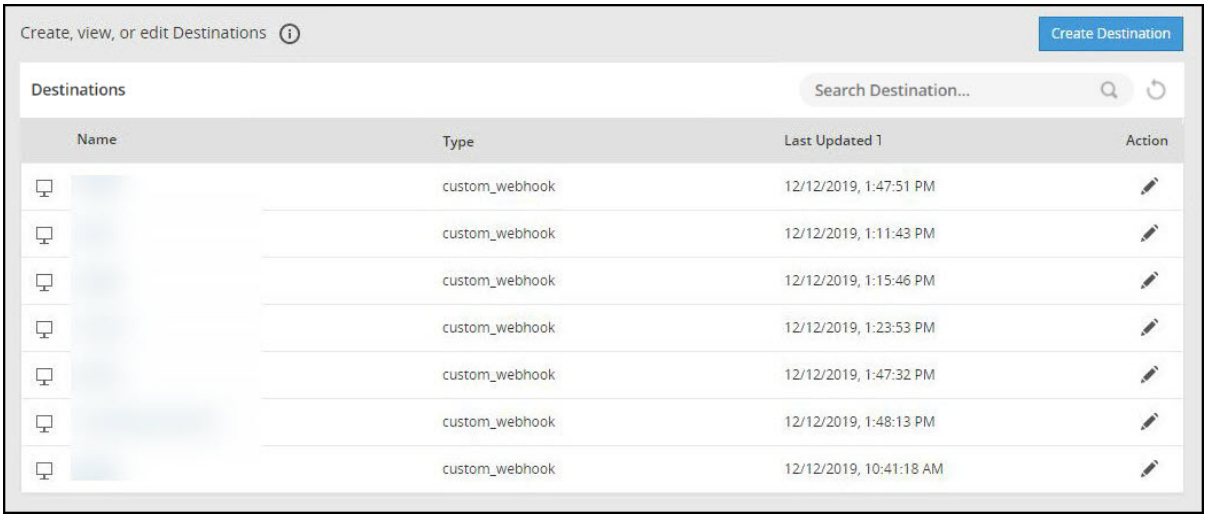

Figure 5-8: Destination List

Clear the search results to view all destinations with the **Reset Search** button.

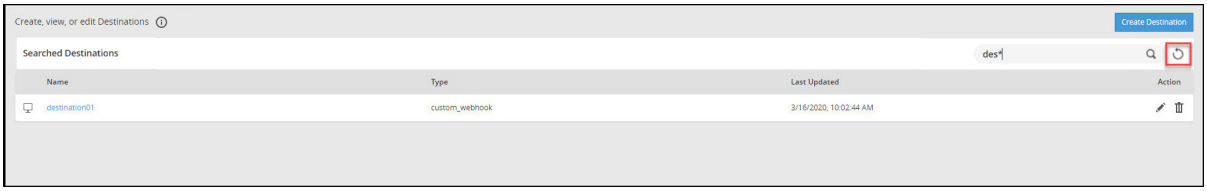

Figure 5-9: Reset Search Button

2. Click the destination name to view the information about the destination.
   A screen with the description about the destination appears.

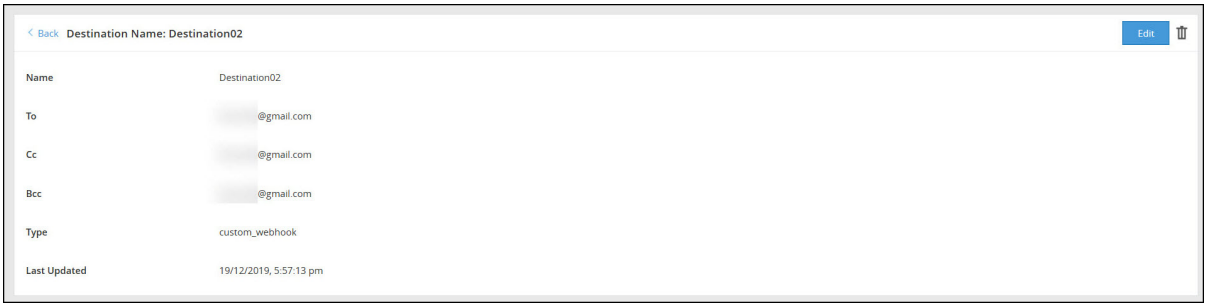

Figure 5-10: Destination Description

3. Select one of the following options. In this case, click **Edit**.
   - **Edit**: Open the edit view to update the definition for the destination.
   - **Delete**: Delete the destination.

   > **Note:** You can also delete destinations that are not required, however, you cannot delete a destination that is linked to the action of a monitor. In this case, you need to first delete or relink the monitor action to a different destination before deleting the destination.

   The destination screen appears in the edit view.

Figure 5-11: Editing the Destination

4. Edit the destination information, as required.
5. Click **Update Destination** to save the updated information for the destination.
   The updated destination appears.

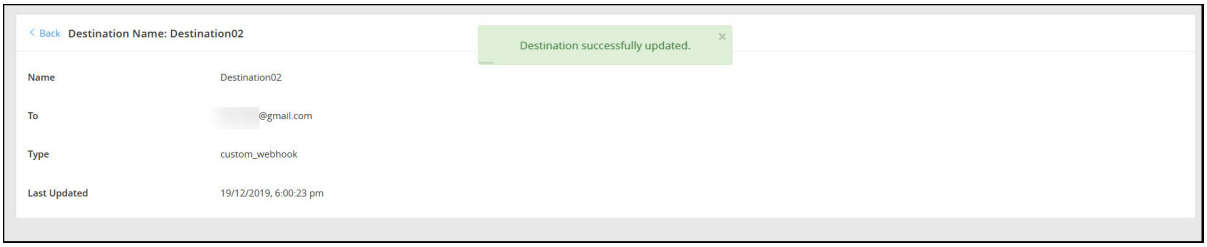

Figure 5-12: Updated Destination

5.3 Working with Monitors

A monitor is a job that is defined to run at regular intervals. It queries the entries in the data store and aggregates the information retrieved. The aggregated data is then sent to a trigger that compares the results against a set of values. If the set criteria is fulfilled, then the trigger initiates an action and sends alerts. You can use monitors to keep a check on the system.

For example, you can define monitors that scan and alert the user if the CPU usage crosses a certain threshold or if restricted tasks are executed. The monitor can be executed as per the schedule or can be executed manually. A monitor runs only when it is enabled. You can disable monitors that are not required or even disable monitors to conserve CPU utilization when you need to run memory intensive tasks.

> **Note:** The *viewer* role user or a user with the *viewer* role can only view monitors. You need admin rights to create or modify monitors.

The Signature Verification monitor is available as default. This monitor raises an alert when the signature verification on a log being processed fails. You can modify the monitor as per your requirements by adding a destination.

5.3.1 Creating a Monitor

You need to create a monitor for analyzing the data in the Audit Store to activate alerts. The **Create Monitor** screen allows you to define the schedule and the query for the monitor. To create the monitor, you need to configure the monitor, specify the trigger that activates the monitor, and specify a destination for the action when the trigger is activated.

5.3.1.1 Configuring the Monitor

You configure the monitor by specifying the schedule when the monitor must run. You need to also specify the extraction query for analyzing the logs. Based on the information retrieved by the query, a trigger and action can be configured for raising alerts.

1. From the Analytics screen, click **Alerting** > **Monitors**.

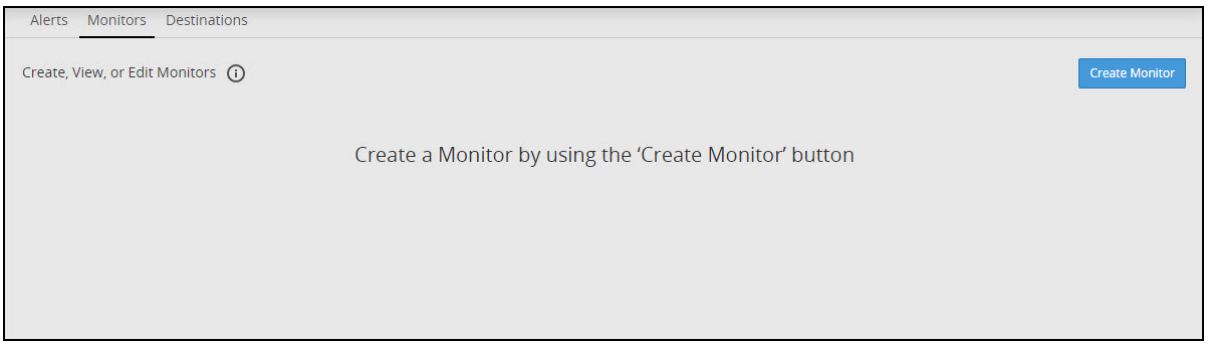

Figure 5-13: Monitors Screen

2. Click **Create Monitor**.

   The **Create Monitor** screen appears.

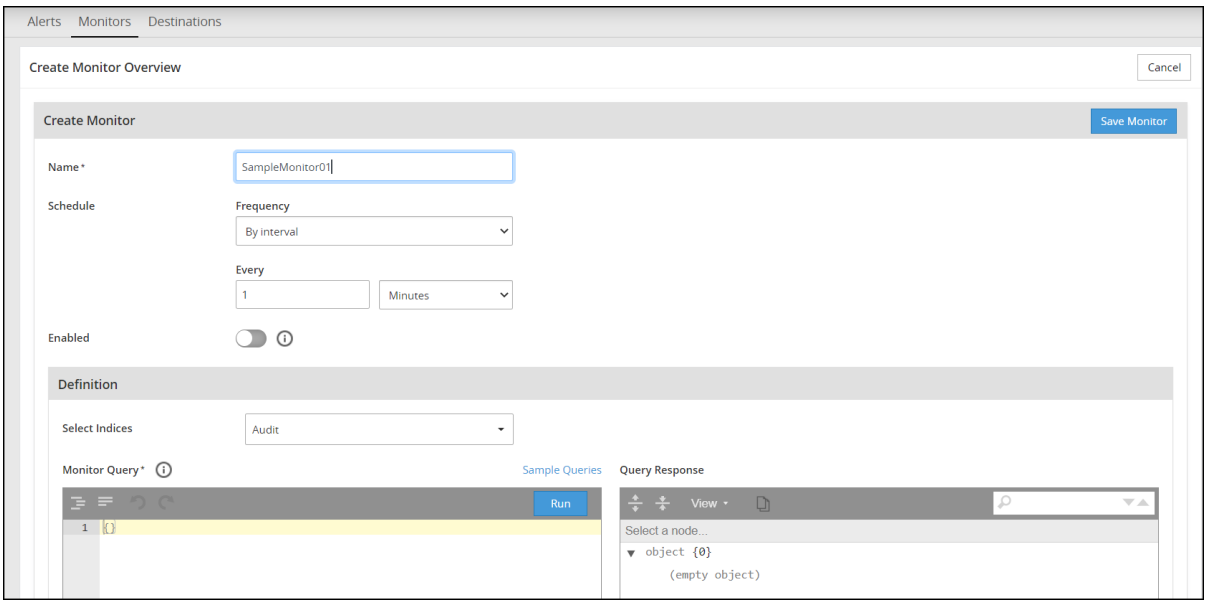

Figure 5-14: Create Monitor Screen

3. Enter the following information:
   - **Name**: Enter a unique name for the monitor.
   - **Schedule**: Select a schedule for running the monitor. The following frequencies are available:
     - **By interval**: This option allows you to run the monitor after the specified time interval.
     - **Daily**: This option allows you to run the monitor everyday at a fixed time.
     - **Weekly**: This option allows you to run the monitor on particular days of the week at a fixed time.
     - **Monthly**: This option allows you to run the monitor on a fixed day of the month.
   - **Enabled**: Select to enable the monitor.
   - **Search Indexes**: Select the indexes or alias to monitor. An alias is a reference, in this indexing feature an **Audit** alias is available. This *Audit* alias is a reference to use all the indexes specified in the **Indices** list.
   - **Monitor Query**: Define a query that the monitor must execute. The query defined should be well formed. If there is an error in the query, then a message appears on the screen for resolving the error.

     > **Note:** Click the **Sample Query** link to view sample queries that you can modify and use.

     The following options are available for working with the query:
     - **Indent code** (): Click to format the code using tab spaces.
     - **Remove white space from code** (): Click to format the code by removing the white spaces and displaying the query in a continuous line.
     - **Undo** (): Click to undo the last change made.
     - **Redo** (): Click to redo the last change made.

4. Click **Save** to save the monitor.

   Click **Run** to run the monitor query instantly and view the result in the **Query Response**. Use this to run the query and check for logical errors. If there are logical errors in the query, then a blank response is received.

   The following options are available to work with the output:
   - **Expand all fields** (): Click to expand all fields in the result.

- **Collapse all fields** (): Click to collapse all fields in the result.
- **Switch Editor Mode** (

): Click to select the editor mode. The following options are available:
  - **View**: Switch to the tree view.
  - **Preview**: Switch to the preview mode.
- **Copy** (): Click to copy the contents of the output to the clipboard.
- **Search fields and values** (): Search for the required text in the output.

5. Configure the trigger using the steps from the section *Configuring the Trigger*.

5.3.1.2 Configuring the Trigger

After you create the monitor, you can define the trigger and the action for the monitor. A trigger that is part of a monitor consists of a name, a severity level, and a condition. The trigger analyzes the input values received from the monitor and checks if the condition is fulfilled. If the condition is satisfied, then the trigger calls the action associated with the monitor. The action defines the alert message and the location, which is the destination, that must receive the alert when a trigger is activated. Use the action to direct the flow of the alerts for the monitor to an email or a website.

Before you begin
Ensure that you have configured the monitor using the steps from the section *Configuring the Monitor*.

1. Specify a unique name for the trigger in the **Name** field.

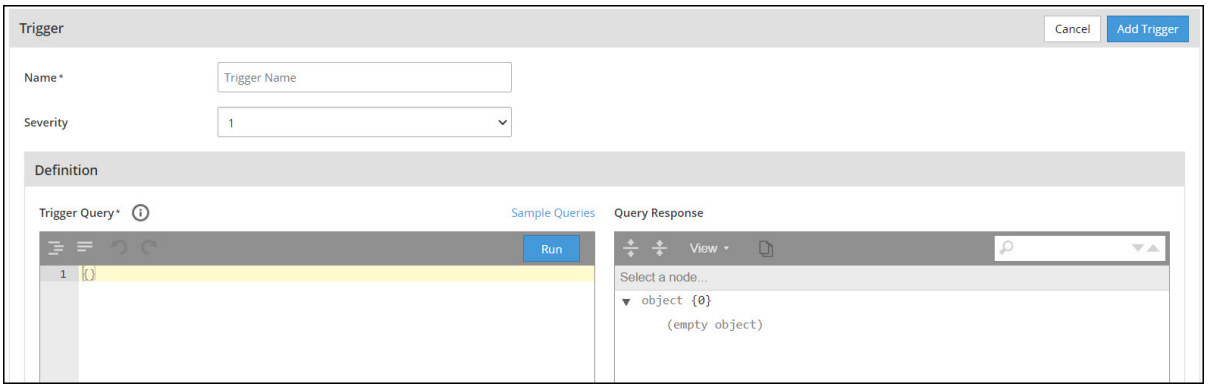

Figure 5-15: Defining a Trigger

2. Select a level from the **Severity** list. The severity level helps you define the severity for the alert. A high severity message can be assigned the level as 1 and a low severity message can be assigned the level as 5.
3. Specify the condition in the **Trigger Query** field using an Audit Store query.
   Click the **Sample Query** link to view sample queries that you can modify and use.
   For more information about the coding standards, refer to the section *Basics of Audit Store Querying*.

   The following options are available for working with the query:

   - **Indent code** (): Click to format the code using tab spaces.
   - **Remove white space from code** (): Click to format the code by removing the white spaces and displaying the query in a continuous line.

- **Undo** (): Click to undo the last change made.
- **Redo** (): Click to redo the last change made.

4. Click **Add Trigger** to add the trigger to the monitor.

> **Note:** Click **Cancel** to go back to the **Monitor** tab.

Click **Run** to run the trigger query instantly and view the result in the **Query Response**. Use this to run the query and check for logical errors. If there are logical errors in the query, then a blank response is received.

The following options are available to work with the output:

- **Expand all fields** (): Click to expand all fields in the result.
- **Collapse all fields** (): Click to collapse all fields in the result.
- **Switch Editor Mode** (

): Click to select the editor mode. The following options are available:
  - **View**: Switch to the tree view.
  - **Preview**: Switch to the preview mode.
- **Copy** (): Click to copy the contents of the output to the clipboard.
- **Search fields and values** (): Search for the required text in the output.

5. Configure the action using the steps from the section *Configuring the Action*.

5.3.1.3 Configuring the Action

The action defines the alert message and the location, which is the destination, that must receive the alert when a trigger is activated. Use the action to direct the flow of the alerts for the monitor to an email or a website.

Before you begin
Ensure that you have configured the monitor and trigger using the steps from the sections *Configuring the Monitor* and *Configuring the Trigger* respectively. In addition, ensure you created a destination using the steps from the section *Creating Destinations*.

1. Specify a name for the action in the **Action Name** field.

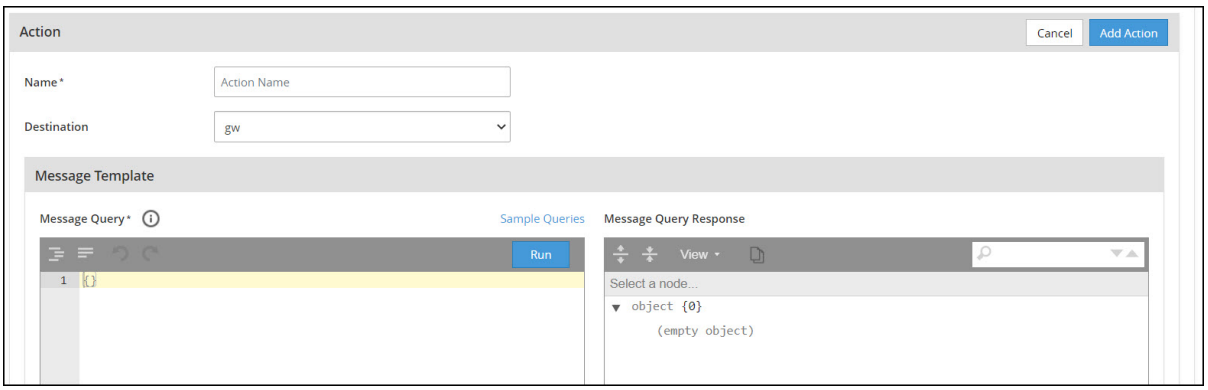

Figure 5-16: Defining an Action

2. Select a destination from the **Destination** list.
3. Create the alert message using the **Message Template** field.

   You can use Mustache template variables, the ctx.action.name variable, the current action, or valid trigger variables to define the alert message. If you are using a webhook, then you can use JSON or XML to define the message body.

   Additionally, you can add HTML formatting to the message by adding *<html></html>* in the *message* tag. An example is displayed here:

```
{
"source": {
"message": "<html><body><b>Monitor Name:</b>{{ctx.monitor.name}}</br><b>start:</
b>{{ctx.periodStart}}</body></html>",
"subject": "Logs alert"
},
"lang": "mustache"
}
```

   For more information about coding, refer to the section *Basics of Audit Store Querying*.

   > **Note:** Click the **Sample Query** link to view sample queries that you can modify and use.

   The following options are available for working with the query:

   - **Indent code** (): Click to format the code using tab spaces.
   - **Remove white space from code** (): Click to format the code by removing the white spaces and displaying the query in a continuous line.
   - **Undo** (): Click to undo the last change made.
   - **Redo** (): Click to redo the last change made.

   Click **Run** to run the action query instantly and view the result in the **Message Query Response**. Use this to run the query and check for errors.

   The following options are available to work with the output:

   - **Expand all fields** (): Click to expand all fields in the result.
   - **Collapse all fields** (): Click to collapse all fields in the result.
   - **Switch Editor Mode** (): Click to select the editor mode. The following options are available:

- **View**: Switch to the tree view.
- **Preview**: Switch to the preview mode.

- **Copy** (): Click to copy the contents of the output to the clipboard.
- **Search fields and values** (): Search for the required text in the output.

4. Click **Add Action**.

> **Note:** Click **Cancel** to go back to the **Monitor** tab.

The monitor is created with the trigger and action attached to it.

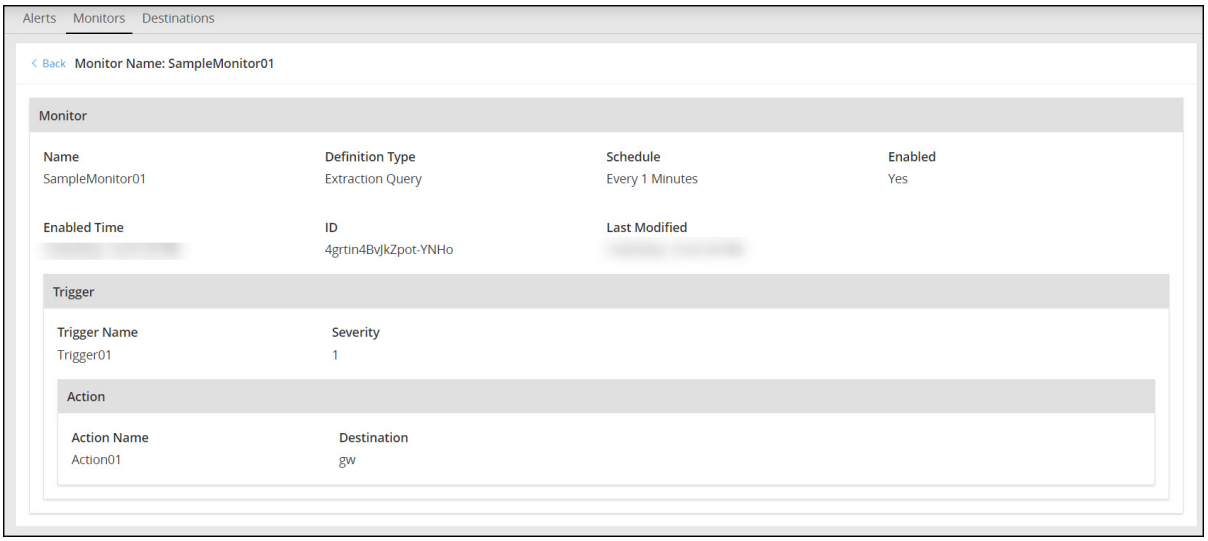

Figure 5-17: Monitor Created

5.3.2 Modifying Monitors

You can view the different monitors available and the information for each monitor. In addition to viewing monitors, you can manage monitors to run as per schedule or disable them when required. If necessary, you can also edit the definition for the monitor.

1. From the Analytics screen, click **Alerting** > **Monitors**.

   The list of monitors created appears. You can use the search field to search for a specific monitor by name from the list. Additionally, you can also use the wildcard character *while searching.

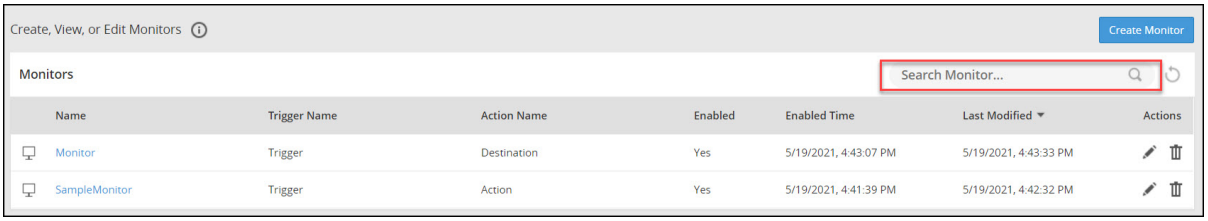

Figure 5-18: Monitor List

Clear the search results to view all monitors with the **Reset Search** button.

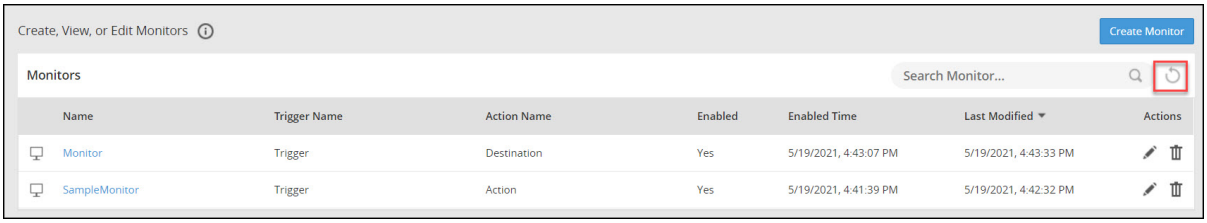

Figure 5-19: Reset Search Button

2. Sort the monitors by clicking the **Name**, **Enabled**, or **Last Modified** column names.

   When the page loads, the items are sorted in the descending order according to the **Last Modified** column. Click the **Name**, **Enabled**, or **Last Modified** column names to re-sort the data. Clicking the column name sorts the items in the ascending order. Clicking the column name again sorts the items in the descending order.

3. Click the monitor name to view the monitor information.

   A screen with the description about the monitor appears.

> **Note:** Click the **Edit** ( ) icon to enter the edit mode for updating the monitor.

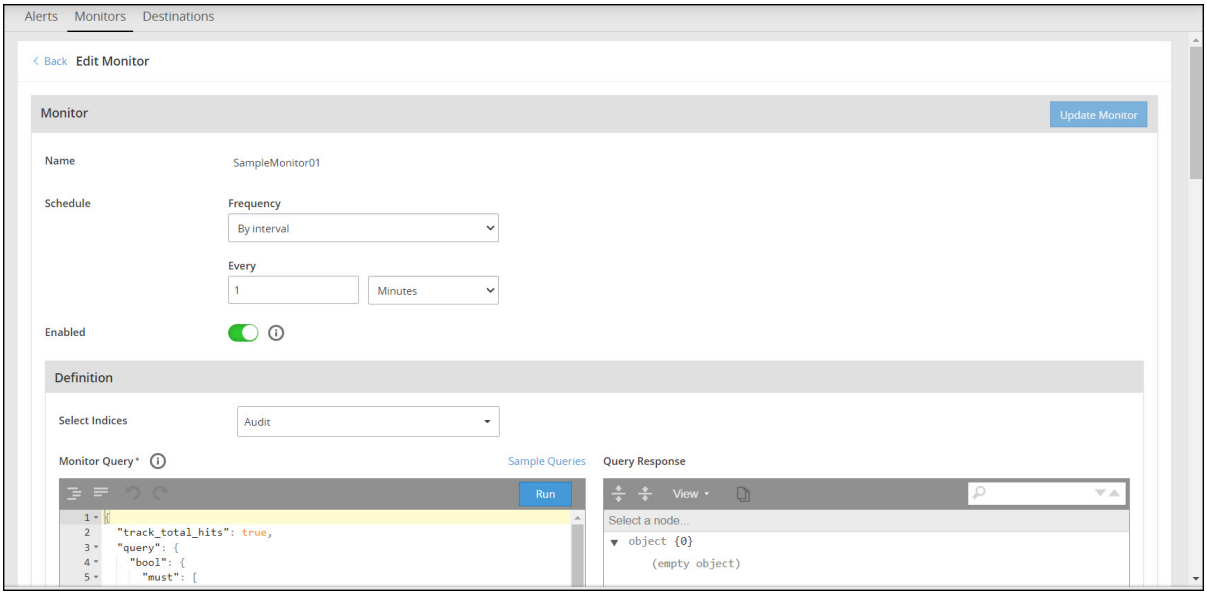

Figure 5-20: Monitor Description

4. Click the required button to perform the following actions for the monitor:
   - **Edit**: Open the edit view to update the definition for the monitor. This option allows you to modify the trigger and action associated with the monitor. You can also delete the trigger and action when you enter the Edit mode.
   - **Disable**: Disable the monitor so that it does not run.
   - **Enable**: Enable the monitor to run at the scheduled time.
   - **Delete**: Delete the monitor with the attached trigger and action.

   > **Note:** Click the **Delete** icon on the **Trigger** or **Action** bar to delete the trigger or action respectively. Deleting the trigger also deletes the action linked with the trigger.

5. Click **Edit** to edit the monitor.
6. Update the monitor, as required.
   a. Update the monitor schedule, state, and query, if required, and click **Update Monitor** to save the changes.

   > **Note:** Click the **Sample Query** link to view sample queries that you can modify and use.

> Clicking **Run Monitor Query** runs the monitor query and displays the result in the **Query Response** field for the monitor.

b. Update the trigger severity level and the query, if required, and click **Update Trigger** to save the changes.

> **Note:** Click the **Sample Query** link to view sample queries that you can modify and use.
>
> Click the **Expand** () icon to add a trigger if the trigger was not added earlier.
>
> Clicking **Run Trigger Query** updates the query results in the **Query Response** field for the trigger. It also runs the monitor query and updates the results in the **Query Response** field for the monitor.

c. Update the destination and message for the action, if required, and click **Update Action** to save the changes.

> **Note:** Click the **Sample Query** link to view sample queries that you can modify and use.
>
> Click the **Expand** () icon to add the action if the action was not added earlier.
>
> Clicking **Run Action Query** updates the query results in the **Message Query Response** field for the action. It also runs the monitor query and the trigger query and updates the results in the **Query Response** fields of the monitor and trigger.

7. Click **Back** to go back to the monitor list.

> **Note:** You can also click **Back** to discard any unnecessary edits you make to the monitor.

Chapter 6

Working with Forensics

6.1 Viewing Logs

Data plays an important part in analysis. You can view the data collected over time to estimate future trends. You can also study existing data to find issues or abnormalities in the current trend. These variations can help you to be better prepared to deal with irregularities in transactions. At times, such irregularities might also be due to unauthorized usage or actions that might need to be studied better to protect your business.

Forensics collates the data from the Audit Store and presents them in tables, lists, and graphs. This displays the data in a format that can be read easily and also provides a better analysis by providing different perspectives for viewing your data. The data in Forensics shows all the data that is stored in the Audit Store. This data is in the raw form and can be used by different tools for analysis.

The Forensics data can be used in the following manner:

- **Auditing**:

  Auditors check the data stored by the organization. This check allows to validate that the required data management practices are followed to maintain the quality of the data. Auditors would check that no data is modified, deleted, or missing.

- **Fixing errors**:

  The logs from all the connected systems are stored in the Audit Store. Viewing this data gives you a broader outlook of the data. You can view and understand the interoperability of the various systems connected. You can view the various errors reported in the Audit Store to fix any configuration errors or problems in the whole ecosystem.

  For example, a client, such as a protector, might send information and receive confirmation that a transaction is successful. However, due to some configuration compatibility issues the server a system might have trouble working with the request. This would show up as an error on the appliance and can be corrected using the information from Forensics.

- **Verifying for issues**:

  You can filter the data from the the Audit Store to find errors and warnings. These can provide further insights about system failures or issues with the system configurations. You can also study the logs to identify breaches in the system or unauthorized access attempts.

- **Reporting**:

  It might be a part of the organizations functioning to prepare and store reports. You can use the Audit Store data to prepare these reports. The data for all your systems is stored in a single Audit Store. Thus, you can create your reports faster and also modify the report formats as per your requirements to better analyze the data.

6.1 Viewing Logs

The logs aggregated and collected are sent to the Audit Store. The logs from the Audit Store are displayed on the **Forensics** screen. Here, you can view the different fields logged and the data logged. In addition to viewing the data, these logs serve as input for Analytics to analyze the health of the system and to monitor the system for providing security.

You can use the default index to view the log data. Alternatively, you can select an index or alias for the entries that you want to view if the data that you require is in a different index. To switch the index, select the required index or alias from the index list. An alias is a reference, in this indexing feature an **Audit** alias is available. This *Audit* alias is a reference to use all the indexes specified in the **Indices** list.

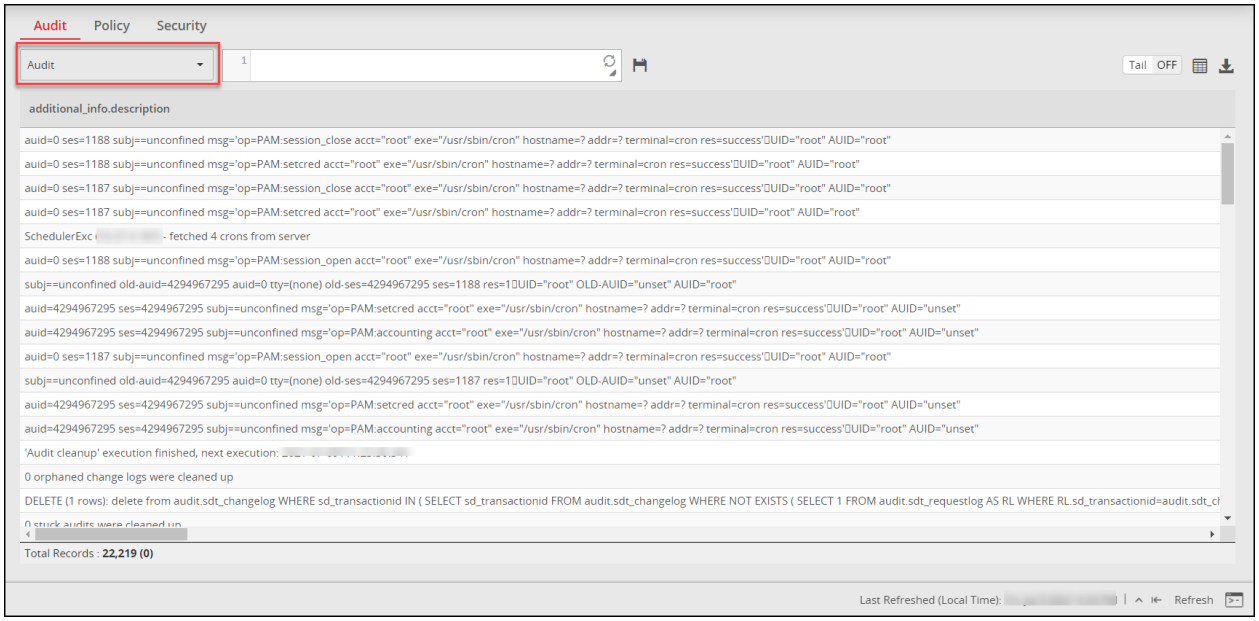

Figure 6-1: Selecting Indexes

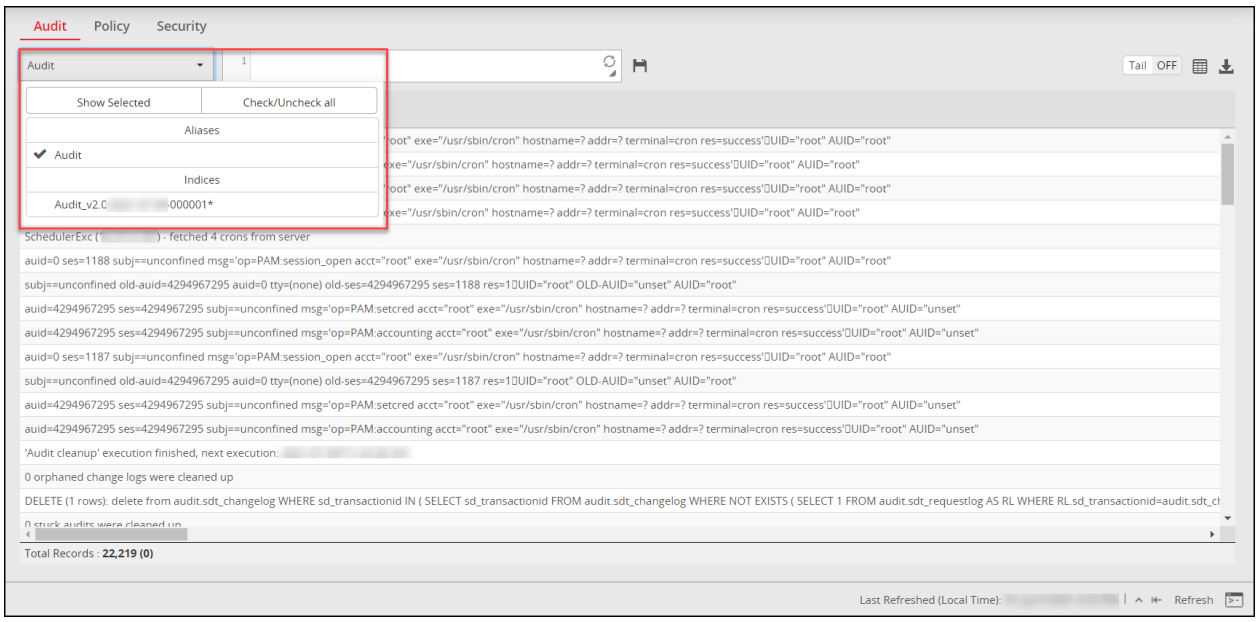

Figure 6-2: Index Entries Displayed

You can set the **Tail** option to retrieve logs when they are generated as follows:

- **Off**: Does not retrieve fresh logs from the ESA.

- **On**: Retrieve logs from the ESA when they are generated. Also displays a graph showing a frequency of the number of logs generated.

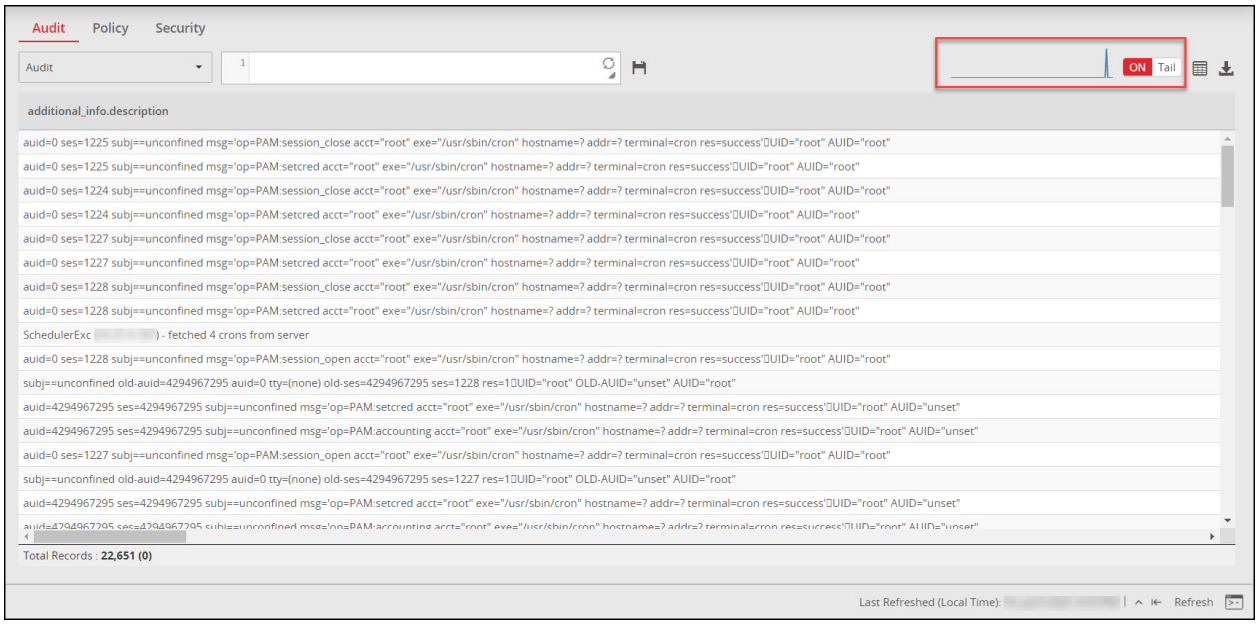

Figure 6-3: Selecting the Tail Option

6.1.1 Fields Logged

The Forensics section displays the various fields logged. These log entries provide information about the machine and events that raised the log message.

You can select the fields that you want to view in Analytics by clicking **Table Settings**. You can then select the number of entries to display from the **Table Size** list. Finally, select the fields that you require by selecting the check boxes from the **Column Display** list. The following fields in the **Column Display** list the entries that have been logged.

> **Note:** The items in the table are standard fields that are tracked for logging. The items available in your system might be different based on your configuration and usage.
>
> The system nano time was logged for records generated in earlier versions of the ESA. The nano time is not logged for records from version 9.0.0.0.

Field	Description
additional_info.description	This is the log message.
additional_info.module	This is the module name that initiated the log.
additional_info.procedure	This is the list of methods that were executed within the module.
additional_info.title	This is a dynamic field which can hold custom values of different protectors.
client.ip	This is the IP address of the machine where the log is generated.
client.username	This is the user who triggered an event which generated the log.
cnt	This is the number of log messages aggregated.
correlationid	This is the client handle ID to uniquely identify the transaction request as per the transaction metric.

Field	Description
endtime	This the end time of the process request as per the transaction metric.
filetype	This is the file type, such as, regular file, directory, or device, when operations are performed on the file.
index_node	The index node that ingested the log.
index_time_utc	This is the UTC timestamp when the log entry was indexed on the Audit Store.
ingest_time_utc	This is the UTC timestamp when the log entry was received by td-agent.
level	This is the log message type.
tiebreaker	This is an internal field that is used with the index time to make a record unique across nodes for sorting.
logtype	This is the log type.
metering.meteringmode	This is the mode for metering logs, such as, delta or total.
metering.origin	This is the IP from where metering data originated.
metering.protection_count	This is the number of protect operations metered.
metering.reprotection_count	This is the number of reprotect operations metered.
metering.timestamp	This is the UTC timestamp when the metering log entry was generated.
metering.uid	This is the unique ID of the metering source that generated the log.
metering.unprotection_count	This is the number of unprotect operations metered.
operation	This is the operation type (Protect, unprotect, or reprotect).
origin.hostname	This is the hostname of the system that generated the log entry.
origin.ip	This is the IP address of the system that generated the log entry.
origin.time_utc	This is the UTC timestamp when the log entry was generated. **Note:** If you are changing the timezone, then a restart of the Policy Management services is required for the log entries to be generated as per the new timezone. If the older logs are not yet processed in scenarios, such as, the *td-agent* service is in *stopped* state, then the origin time differs. In such a scenario, the origin time after UTC conversion for these log entries are generated as per the new timezone in the Forensics.
path	This field is provided for protector-related data and will be available in a future release.
policy.audit_code	This is the policy audit code for the policy log.
policy.policy_name	This is the policy name for the policy log.
policy.severity	This is the severity level for the policy log entry.
policy.username	This is the user who modified the policy.
process.id	This is the process ID of the event that generated the log.
process.name	This is the name of the process that generated the log.
process.pcc_version	This is the core pcc version.
process.platform	This is the OS name for which the log entry was generated.
process.thread_id	This is the process thread ID.

Field	Description
process.user	This is the ID of the user logged in when the log message was generated.
process.version	This is the version of the installed product.
protection.audit_code	This is the audit code of the protect operation.
protection.dataelement	This is the data element used for the protect operation.
protection.datastore	This is the data store, which corresponds to the policy deployed.
protection.devicepath	This is the device path in product, such as, FPVE, FPWIN, and so on.
protection.filetype	This is the file type, such as, regular file, directory, or device, when an operations are performed on the file.
protection.mask_setting	This is the mask setting from policy management.
protection.old_dataelement	This is the data element used to unprotect data during a reprotect operation.
protection.operation	This is the operation type (Protect, unprotect, or reprotect).
protection.path	This represents the absolute path of the protected file or volume.
protection.policy	This field is provided for protector-related data and will be available in a future release.
protection.policy_user	This is the user that performed the protect, unprotect, or reprotect operation.
protection.request_id	This is the ID for the request of a statement run on the protector (for 7.x and older protectors).
protection.role	This is the role of the user who raised the protect, unprotect, or reprotect operation.
protection.session_id	This is the ID for the session for a statement run on the protector (for 7.x and older protectors).
protector.core_version	This is the PEP server version.
protector.family	This is the protector family name for which the log was generated.
protector.vendor	This is the protector vendor name for which the log was generated.
protector.version	This is the protector version number.
returncode	This is the return code of the application log.
signature.checksum	This is the hash value of the signature key ID used to sign the log message when the log is generated.
signature.counter	This is the chain of custody value. It helps maintain the integrity of the log data.
signature.key_id	This is the key used to sign the log message when the log is generated.
starttime	This the start time of the process request as per the transaction metric.
system_nano_time	This is the Epoch time in milliseconds when the log is indexed in the Audit Store.
uri	This is the end point of the HTTP request as per the transaction metric.
verification.doc_id	This is the document ID for the audit log where the signature verification failed.
verification.index_name	This is the index name where the log signature verification failed.
verification.job_id	This is the job ID of the signature verification job.
verification.job_name	This is the job name of the signature verification job.
verification.reason	This is the audit log specifying the reason of the signature verification failure.

Field	Description
_id	This is the entry id for the record stored in the Audit Store.
_index	This is the index name of the Audit Store where the log is stored.

6.1.2 Querying Logs

The logs generated are stored in the Audit Store. Navigate to **Forensics** > **Audit** to view the logs from the Audit Store. If many logs are displayed, then you can filter the logs to view only the logs that you require using basic queries.

> **Note:** The *viewer* role user or a user with the *viewer* role can only view logs using query filter. You need admin rights to create or modify query filters.

Use the **filter** field to type your query for filtering logs. The different field names available are shown as suggestions, as you type the query. The number of results are displayed at the bottom of the screen. A spinner appears in the field when the system is processing the request. Delete the query that you typed to remove the filter and display all the logs. If you specify an incorrect query or your query cannot be processed, then an error message appears on the screen.

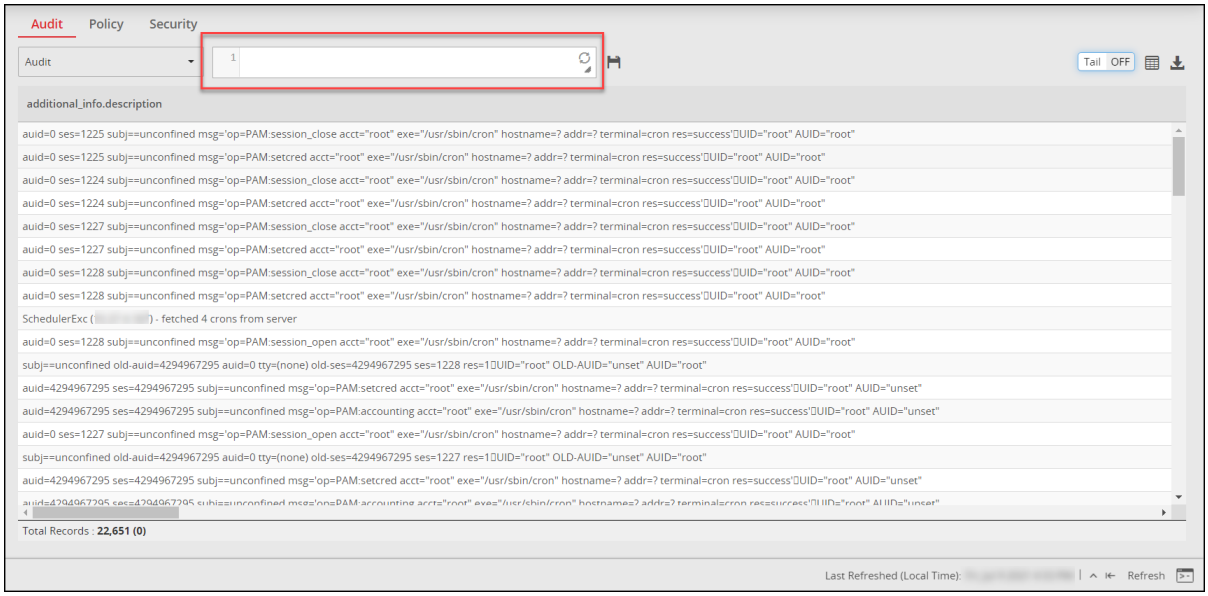

Figure 6-4: Audit tab

Use the following sytax for filtering logs:

```
select * from table where <field_name><operator><value>
```

Where:

- *field_name* is the column name for filtering. This value is case-sensitive.
- *operator* is the criteria that you require. Valid values are *=*, *>*, *<*, *<=*, *>=*, and *!=*.
- *value* is the text for filtering. Enclose strings within double quotes.

> **Note:** You can also specify the query for the current result set by only specifying <*field_name*><*operator*><*value*> in the search box.
>
> Ensure that strings are enclosed using double quotes. Using single quotes might not return the correct results for the query.
>
> Do not use quotes or double quotes for working with date and time values.

You can also chain and use advanced filtering by using the *AND* and *OR* operators.

Specify multi-line queries using *{*. To auto-format the query using a text area, select the query text and click the **Format** button or press Ctrl+Alt+F.

Autocomplete Queries

The filter field autocompletes queries when they are typed. Click the **refresh** () icon from the upper-right corner of the **filter** field to refresh and recreate the autocompletion index manually.

> **Note:** After refreshing the autocompletion index, refresh or reload the *Forensics* page manually.

SQL Filtering Query Examples

Here are a few sample search queries with the description.

- The following snippet displays all logs where *origin.time_utc* is greater than or equal to *2020-08-19T11:32:23.000000000Z*.

```
select * from table where origin.time_utc >= 2020-08-19T11:32:23.000000000Z
```

- The following snippet displays all logs generated from the IP address *192.168.12.10*.

```
select * from table where origin.ip="192.168.12.10"
```

- The following snippet displays all logs generated from the IP address *192.168.12.10*, where the index_time_utc is greater than *2020-08-19T11:32:23.000000000Z*.

```
select * from table where origin.ip="192.168.12.10" AND
index_time_utc>2019-10-16T06:20:50.311585Z
```

- The following snippet displays all logs where the index_time_utc is greater than *2020-08-19T11:32:23.000000000Z*, exclude logs from the IP address *192.168.12.10*.

```
select * from table where origin.ip!="192.168.12.10" AND
index_time_utc>2019-10-16T06:20:50.311585Z
```

JSON Filtering Query Examples

You can run JSON queries for retrieving data from the data store. Here are a few sample queries that you can use.

- The following snippet displays all the logs with the severity level set to *high*.

```
{
    "query" : {
        "query_string" : {
            "query" : "level:high"
        }
    }
}
```

- The following snippet displays all the logs with the severity level set to *high*. This is another way to write the code explained in the previous snippet.

```
{
    "query" : {
        "term" : {
            "level" : {
                "value" : "high"
            }
        }
    }
}
```

- The following snippet displays all the logs between the two time durations specified.

```
index_time_utc<2019-10-16T06:20:54.311585Z and index_time_utc>2019-10-16T06:20:50.311585Z
```

- The following snippet displays all the warning logs generated between the two time durations specified.

```
index_time_utc<2019-10-16T06:20:54.311585Z and index_time_utc>2019-10-10T06:20:50.311585Z
and level=warning
```

- The following snippet displays all the high and severe level logs generated between the two time durations specified.

```
index_time_utc<2019-10-16T06:20:54.311585Z and index_time_utc>2019-10-10T06:20:50.311585Z
and (level=high or level=severe)
```

- The following snippet displays all the logs excluding logs with the level *high* and *fine*. This is an example that uses the not (**!=**) operator.

```
level!=high and level!=fine
```

- The following snippet displays all the logs excluding logs with the level *info*, *low*, *warning*, and *fine*.

```
level!=info and level!=low and level!=warning and level!=fine
```

- The following snippet displays all the logs where the *client.ip* exists. This also returns logs where the *client.ip* contains a space.

```
_exists_=client.ip
```

- The following snippet displays all the logs where the *client.username* does not exist.

```
_exists_!=client.username
```

For more information about writing JSON queries, refer to *https://www.json.org/json-en.html*.

6.1.3 Working with Saved Queries

You can customize the log details displayed when you run a query. You can save the query and the settings for running a query, such as, the columns, row count, tail, and indexes for the query. The saved queries that you create are user-specific.

There are two global queries, *Policy* and *Security*, that are installed with the upgrade. The *Policy* query is available to filter and display all the policy logs. The *Security* query is available to display all protector logs. The *Policy* and *Security* global queries are available for all users, deleting or modifying these queries updates it for all users. The global saved queries are shown in the following figure.

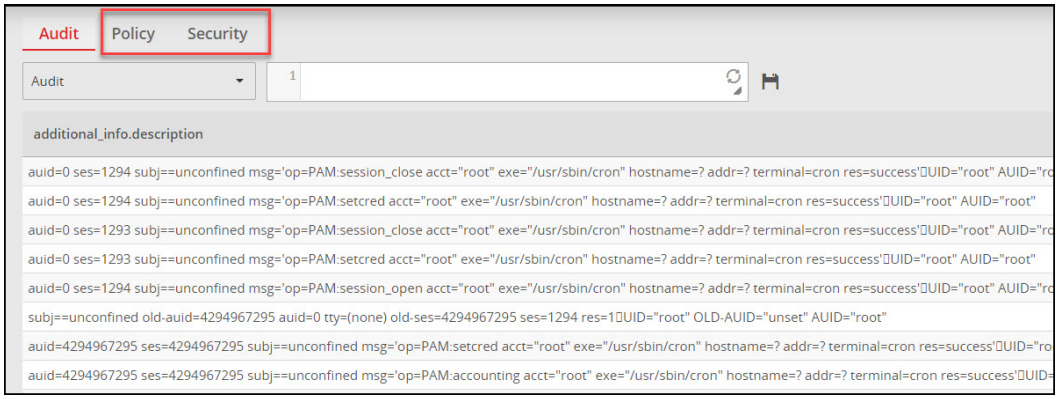

Figure 6-5: Global Saved Queries

1. In Analytics, navigate to the **Forensics** screen.

   > **Note:** The *viewer* role user or a user with the *viewer* role can only view and run saved queries. You need admin rights to create or modify query filters.

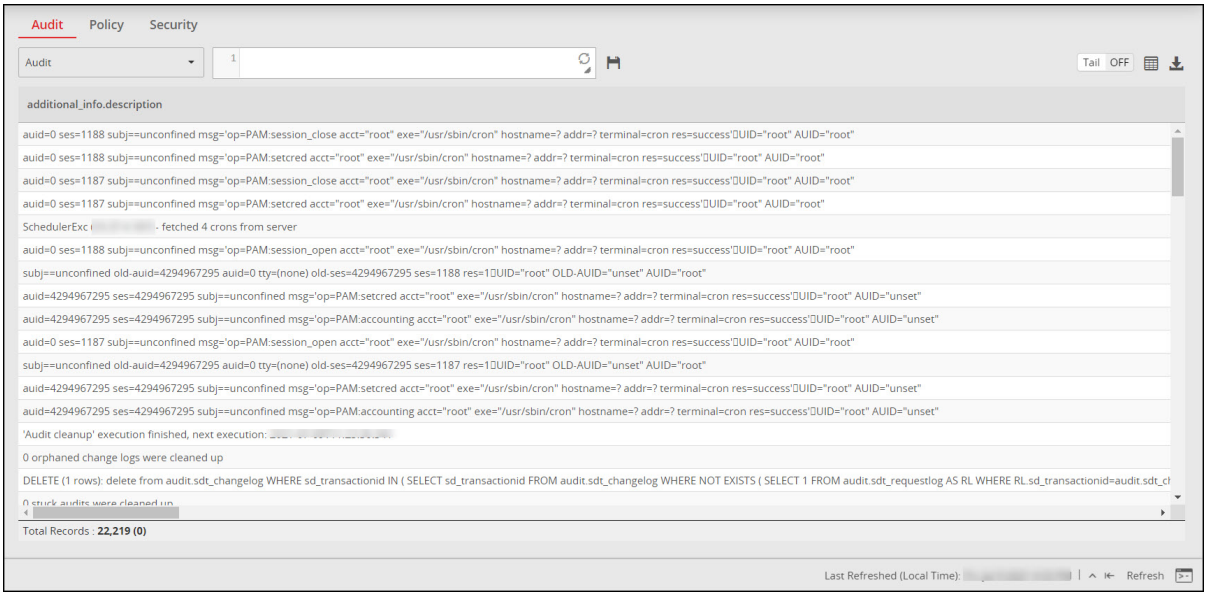

Figure 6-6: Forensics Screen

2. Select the index where you want to run the query.
3. Enter the query in the **filter** field.
4. Select the required tail option.
5. Click the **Table Settings** icon, select the table size, and the list of columns that you want to include in the query. You can select to show or hide the unselected fields. Additionally, you can click to drag and sort the arrangement of the columns from this screen. Finally, click **OK** from the **Table Settings** screen to apply the settings you specified.

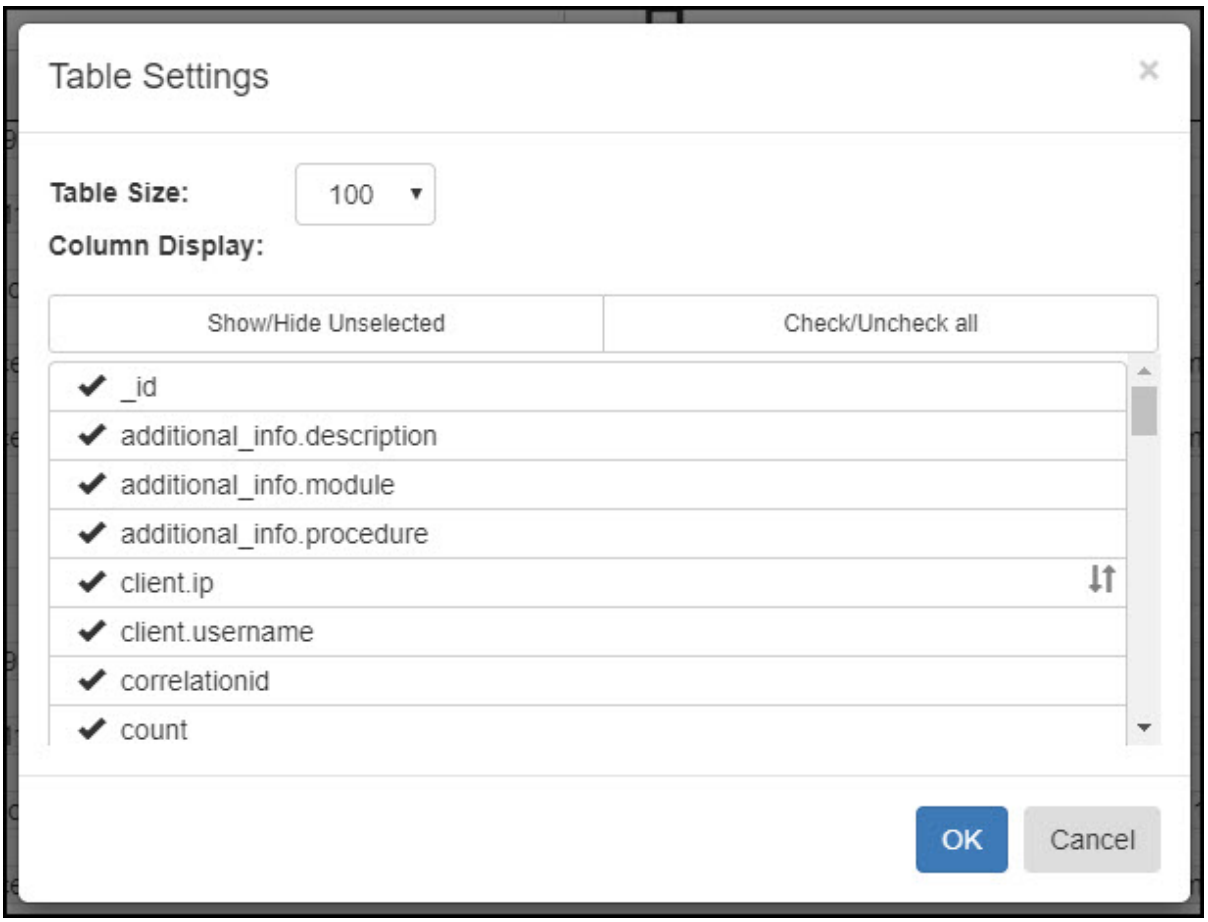

Figure 6-7: Table Settings Screen

6. Click the **Save** ( ) icon to save the query.
   The **Save Query** dialog box appears.

Figure 6-8: Save Query Dialog Box

7. Specify a name for the query. You can modify the name later.
   The name must satisfy the following conditions:
   - Query name must be unique.
   - Allowed characters are A-Z, a-z, 0-9, [space], and -.
   - Maximum length for the query name is 40 characters.

8. Click **Save** to save the query information, including the configurations that you specified, such as, the columns, row count, tail, indexes, and query.

   The query is saved and displayed as a shortcut on the toolbar.

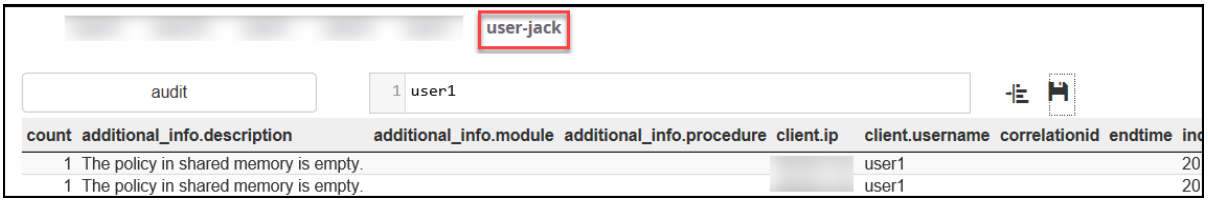

Figure 6-9: Saved Query

9. Click the query from the toolbar to run the saved query using the display options you specified.

> **Note:** Hover the mouse over the saved query, click the **Delete** ( ) icon, and confirm deletion to delete the saved query.
>
> Hover the mouse over the saved query, click the **Rename** ( ) icon, type a new unique name, and press **Enter** to rename the saved query. You can press **Esc** to exit the rename mode without updating the query name.

6.1.4 Exporting Logs to an External File

You can save the logs displayed on the Forensics screen to an external file in the `.csv` or `.json` file format. You can then use this information for further analysis or report generation using other tools.

1. In Analytics, navigate to the **Forensics** screen.

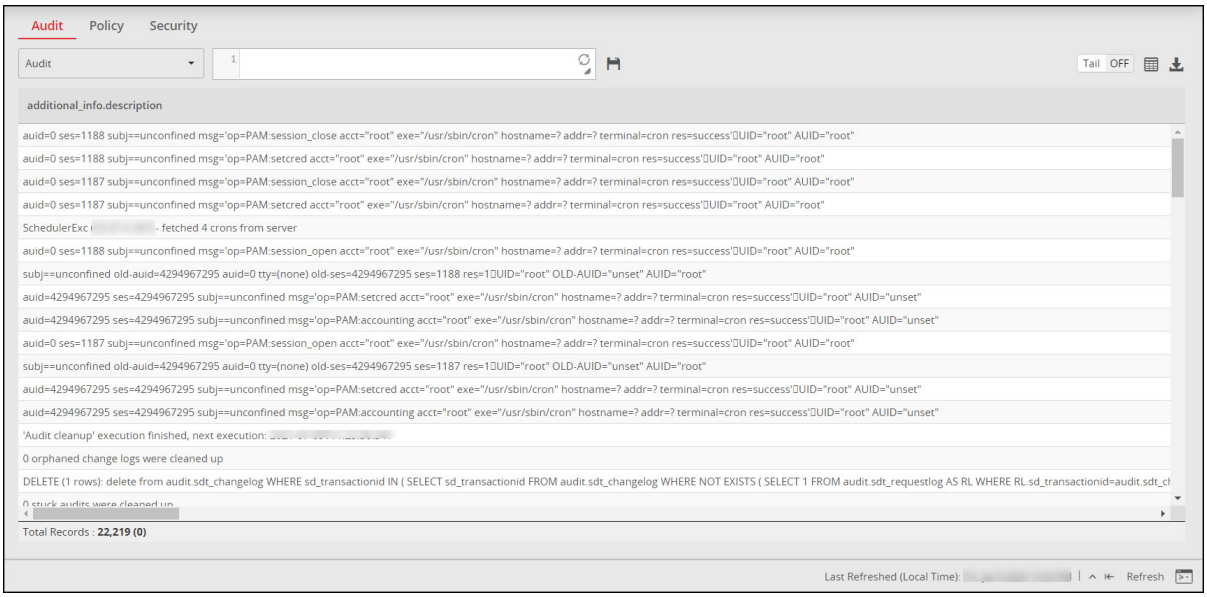

Figure 6-10: Forensics Screen

2. Select the index for the data you want to export.
3. If you want to filter records, then enter the query in the **filter** field.
4. From the top-right section, click the **Download Data** icon.

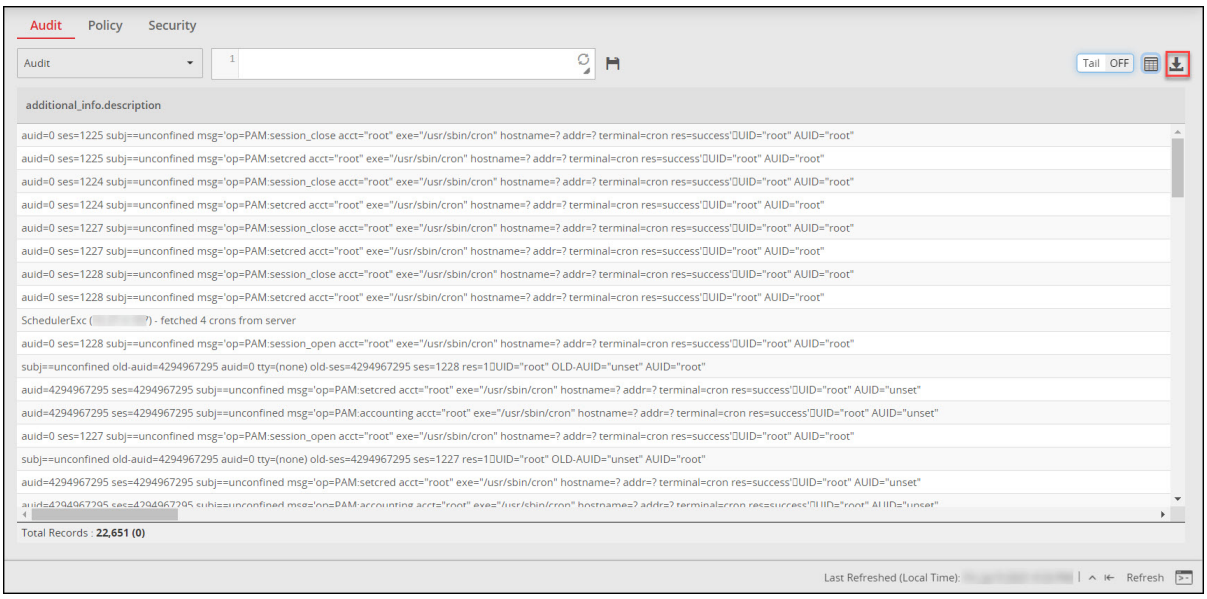

Figure 6-11: Downloading Data

The **Download** screen appears.

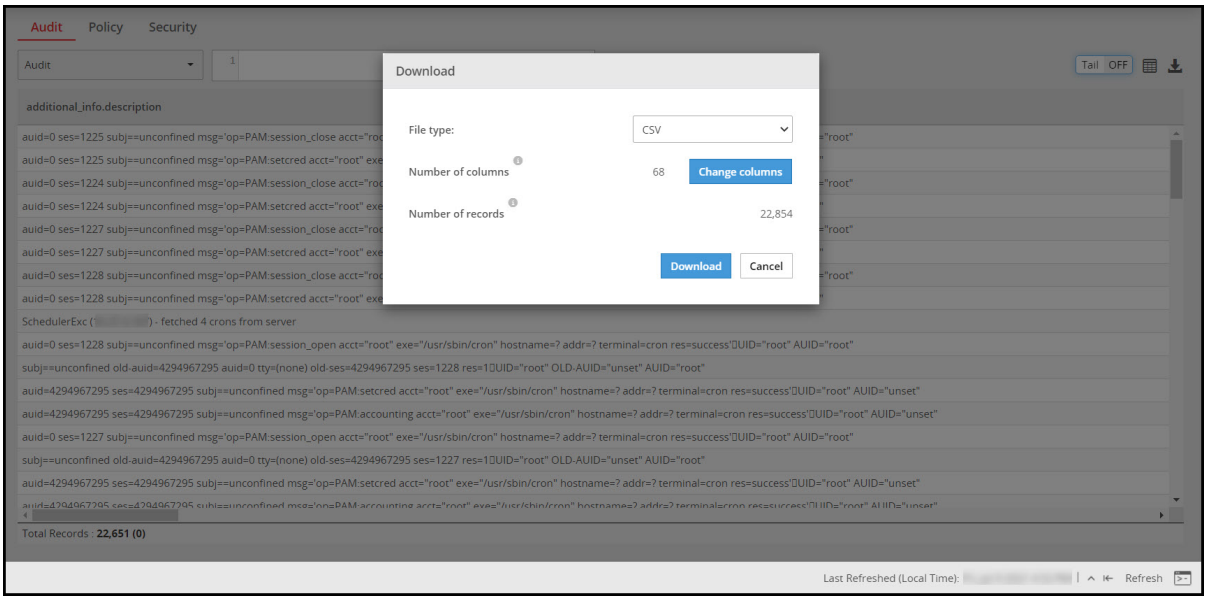

Figure 6-12: Download Screen

5. Select a file type. The available choices are `.csv` and `.json`.
6. Click **Change columns**, select the table size, list of columns that you want to include in the download file, and click **OK** from the **Table Settings** screen.

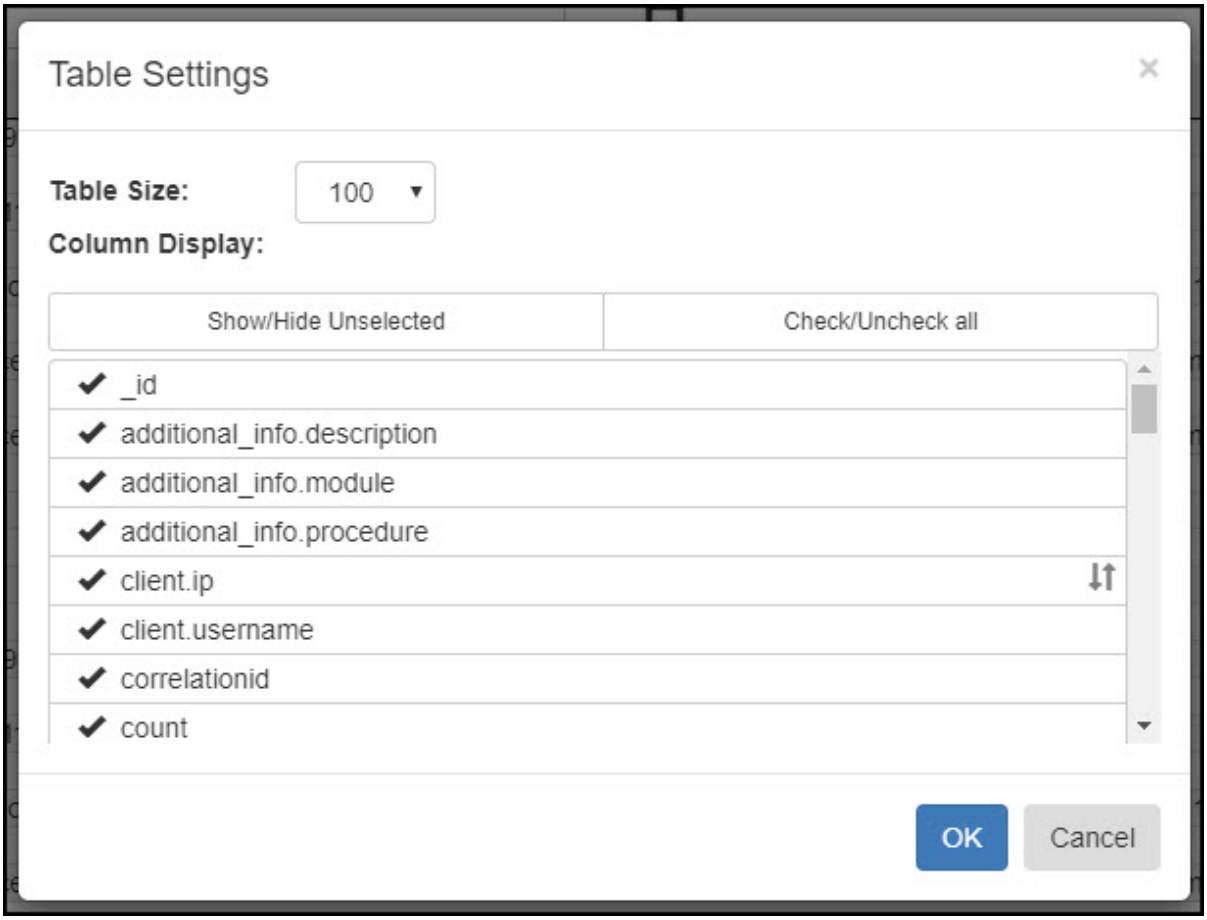

Figure 6-13: Table Settings Screen

7. View the number of columns and number of records that will be downloaded in the **Number of columns** and **Number of records** fields.
8. Click **Download**.

   The browser saves the log files to your machine based on the settings that you selected.

Chapter 7

Viewing Reports

7.1 Viewing Policy Reports
7.2 Working with Reports

The Audit Store contains all the logs and data related to the functioning of your systems. You can use the reporting tools to extract and view selective data from the Audit Store. These reports provide an insight to the data and shows you any discrepancies or irregular behavior that might have occurred.

7.1 Viewing Policy Reports

Policies control the access and rights provided to users over files and records. These access-related tasks are logged and presented to the user when required. It enables users to monitor the files and the data accessed. This report is generated by the triggering agent every time a policy or data store is added, modified, or deleted. It can be analyzed and used for an audit for ascertaining the integrity of policies. If a report is present where policies were not modified, then a breach might have occurred. These instances can be further analyzed to find and patch security issues. A new policy report is generated when this reporting agent is first installed on the ESA. This ensures that you have the initial state of all the policies on all the data stores in the ESA. A user can then use the Protegrity Analytics to list all the reports that were saved over time and select the required reports.

Before you begin
Ensure that the policies you want to view in the report are deployed. Complete the following steps to view the policies deployed.

1. Login to the ESA Web UI.
2. Navigate to **Policy Management** > **Policies & Trusted Application** > **Policies**.
3. Verify that the policies you need to track are deployed and have the **Deploy Status** as **OK**.

> **Note:** If you install the reporting tool when a policy is being deployed, then the policy status in the report might show up as *Unknown* or as a warning. In this case, you need to manually deploy the policy again so that it is displayed in the Policy Report.

➤ To view the policy report.

1. In Analytics, navigate to **Reporting** > **Policy**.
   The **Policy** screen appears.

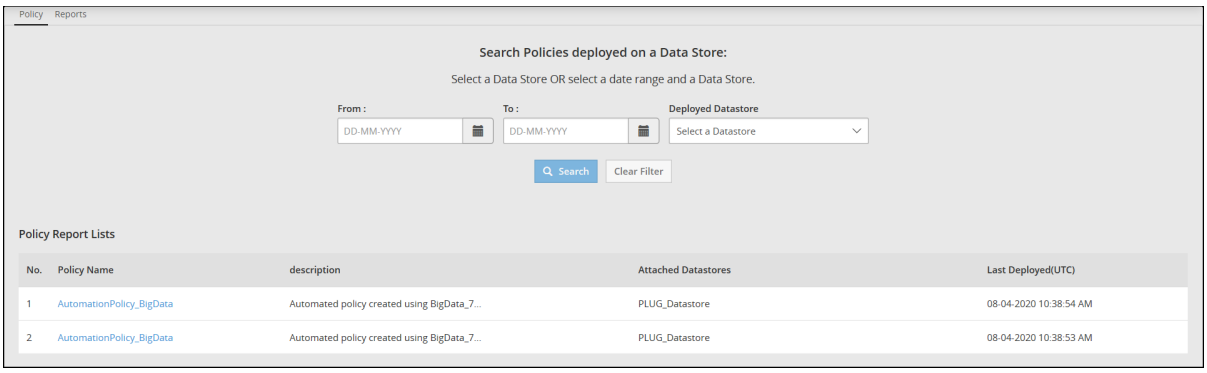

Figure 7-1: Policy Screen

2. Select a time period for the reports you want to view using the **From** and **To** date picker.

> **Note:** This step is optional. Select the time period to narrow your search results for the number of reports displayed for the selected data store.

3. Select a data store from the **Deployed Datastore** list.
4. Click **Search**.
   The reports are filtered and listed based on your selection.

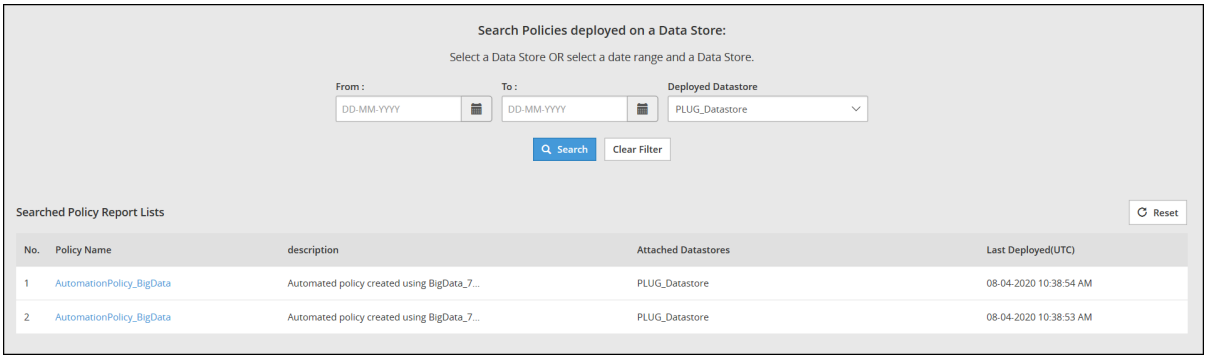

Figure 7-2: Filtering Reports

5. Click the link for the report you want to view.

   For every policy deployed, the following information is displayed:

   - **Policy details**: This section displays the name, type, status, and last modified time for the policy.
   - **List of Data Elements**: This table displays the name, description, type, method, and last modified date and time for a data element in the policy.
   - **List of Data Stores**: This table lists the name, description, and last modified date and time for the data store.
   - **List of Roles**: This table lists the name, description, mode, and last modified date and time for a role.
   - **List of Permissions**: This table lists the various roles and the permissions applicable with the role.

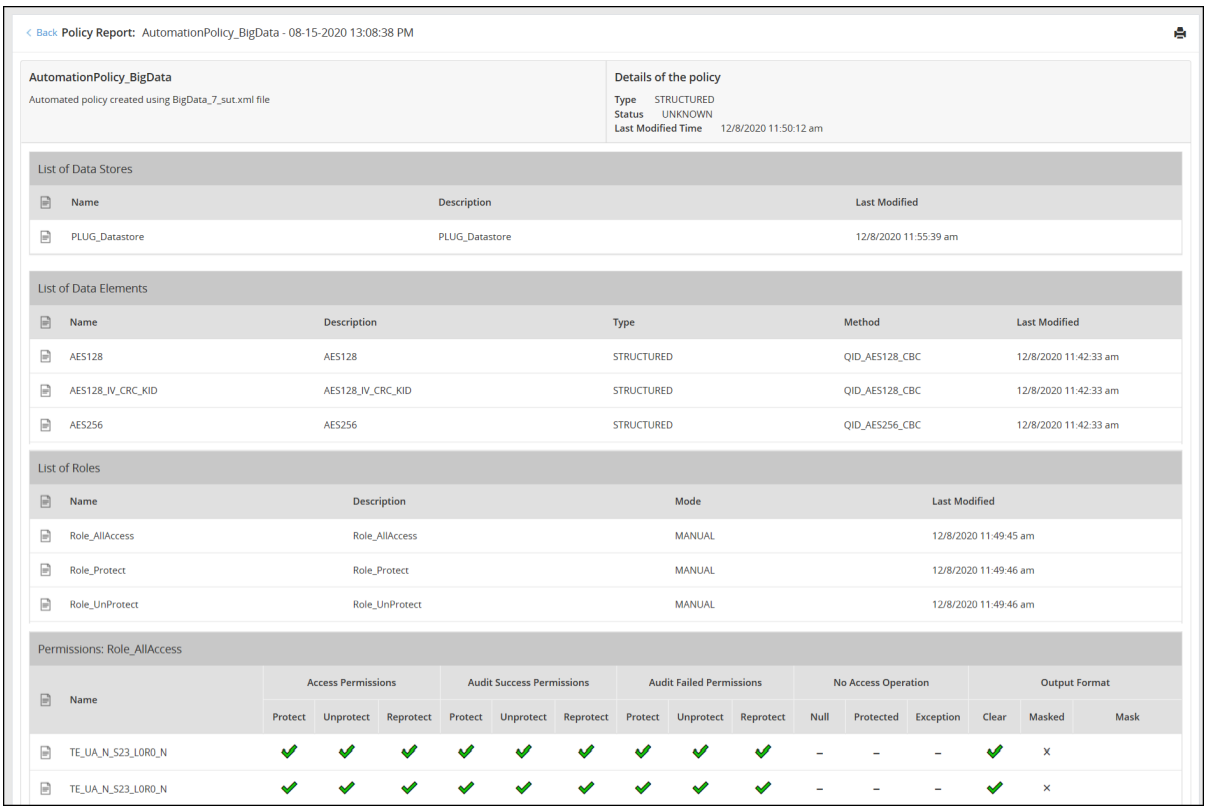

Figure 7-3: Report Displayed

6. You can print the report for comparing and analyzing the different reports that are generated when policies are deployed or undeployed. Alternatively, you can click the **Back** button to go back to the search results.

> **Note:** Ensure that you print the report using the landscape mode.

7.2 Working with Reports

Reports allows you to retrieve information from the Audit Store. You can customize the reports using custom queries as per your requirements. Thus, you can create queries to retrieve the workings about a specific system, or you can query to find errors, or to identify abnormal operations from the logs. You can save these report queries and run them when necessary.

> **Note:** The *viewer* role user or a user with the *viewer* role can only view, execute, and download report data. You need admin rights to create or modify reports.

7.2.1 Viewing Reports

The various reports available on the system are displayed on the Reports page. In Analytics, navigate to **Reporting** > **Reports**. The Reports page appears. From this page, you can create, edit, run, search, and save the report data to an external file.

The Reports page is displayed here.

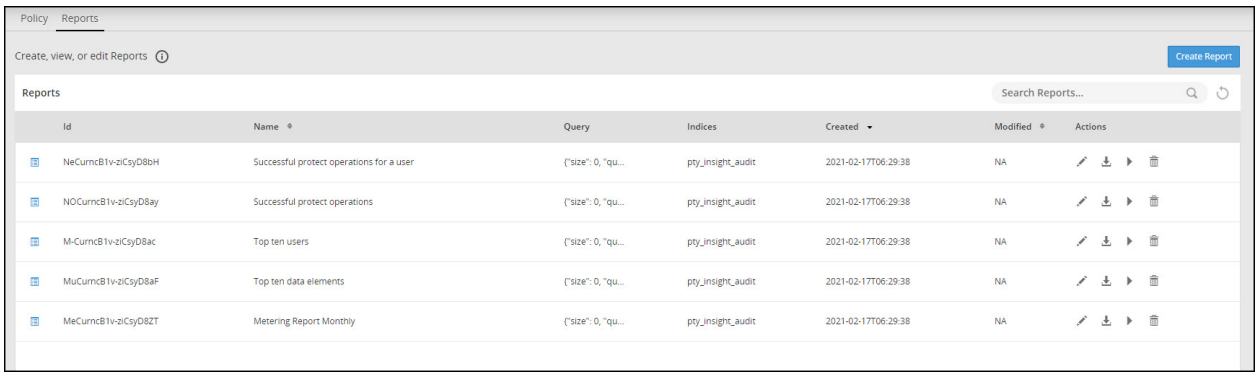

Figure 7-4: Reports Page

> **Note:** The *Metering Monthly*, *Top ten users*, *Top ten data elements*, *Successful protect operations*, and *Successful protect operations for a user* are the sample reports available. The output of these reports are also available in a graphical format on the dashboard.
>
> For more information about the overview dashboard, refer to the section *Viewing the Overview Dashboard*.

You can use the search field to filter and find the reports that you require. Click the **Reset Search** icon to clear the filter and view all reports.

The following columns are displayed for the reports listed on this page.

Column Name	Description
Name	This is the unique report name.
Description	This is the description text that states what the report retrieves. If the description text is not visible, then hover the mouse over the description to view the entire text.
Query	This is the query that is executed for retrieving data for the report. If the query text is not visible, then move the mouse over the query to view the entire text.
Created	This is the date and time when the report was created.
Modified	This is the date and time when the report was edited.
Last Run	This is the date and time when the report was last successfully executed.
Actions	The following actions can be performed on the reports: • Click the **Edit** ( ) icon to modify the report. • Click the **Download** ( ) icon to download the report data to a file. • Click the **Execute** ( ) icon to run the report and view the results. • Click the **Delete** ( ) icon to delete the report.

> **Note:** Click the **Name**, **Created**, **Modified**, or **Last Run** column names to sort the reports in the ascending order. Click the column name a second time to sort the reports in the descending order of the required column.

7.2.2 Creating a Report

You can specify a query to create a report with the information that you require. You can save the results of the report in a file as part of your audit or documentation requirements.

1. In Analytics, navigate to **Reporting** > **Reports**.
   The **Reports** screen is displayed.
2. Click **Create Report**.
   The **Create Report** screen is displayed.

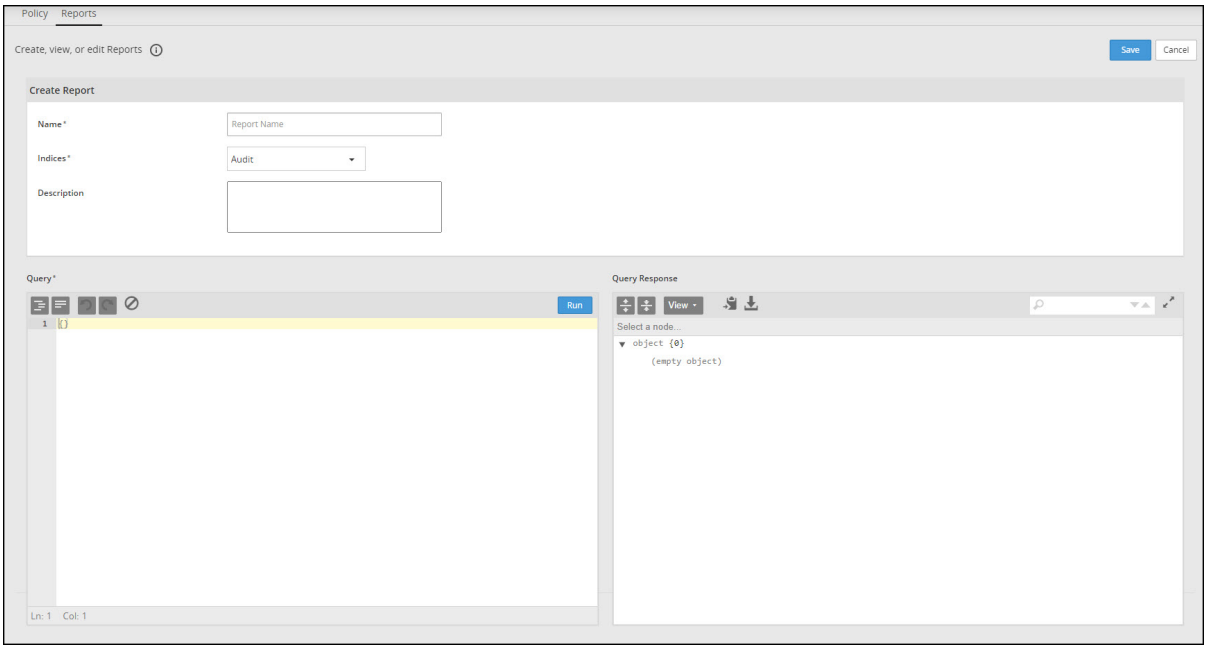

Figure 7-5: Create Report Page

3. Specify a unique name for the report in the **Name** field.
4. Select the index or alias to query from the **Indices** list. An alias is a reference to one or more indexes available in the **Indices** list. The alias is generated and managed by the system and cannot be created or deleted.
5. Specify a description for the report in the **Description** field.
6. Use the **Query** field to specify a JSON query. Errors in the code, if any, are marked with a red cross before the code line.
   The following options are available for working with the query:
   - **Indent code** (): Click to format the code using tab spaces.
   - **Remove white space from code** (): Click to format the code by removing the white spaces and displaying the query in a continuous line.
   - **Undo** (): Click to undo the last change made.
   - **Redo** (): Click to redo the last change made.
   - **Clear** (): Click to clear the query text.

   > **Note:** For sample JSON queries for building reports, refer to the section *Example Queries*.

7. Click **Run** to execute the query.

8. View the result displayed in the **Query Response** field.
   The following options are available to work with the output:
   - **Expand all fields** (): Click to expand all fields in the result.
   - **Collapse all fields** (): Click to collapse all fields in the result.
   - **Switch Editor Mode** (

     ): Click to select the editor mode. The following options are available:
     - **View**: Switch to the tree view.
     - **Preview**: Switch to the preview mode.
   - **Copy** (): Click to copy the contents of the output to the clipboard.

     Note: Use copy to quickly copy a section of the output for further processing. To download all the data, use the **Download** icon.

   - **Download** (): Click to download the contents of the output to a JSON file.
   - **Search fields and values** (): Search for the required text in the output.
   - **Maximize** (): Click to maximize the **Query Response** field. Click **Minimize** () to minimize the field to the original size when maximized.
9. Click **Save** to save the report and return to the **Reports** screen.

7.2.3 Editing a Report

You can update the query in the report to retrieve the data that you require.

1. In Analytics, navigate to **Reporting** > **Reports**.
   The **Reports** screen is displayed.
2. Locate the report that you want to update.
3. From the **Actions** column, click the **Edit** () icon.
   The **Edit Report** screen is displayed.

Figure 7-6: Edit Report Screen

4. Click the report name to expand the section to update the report name, description, and the indexes used.
5. Update the JSON query in the **Query** field, as required. Errors in the code, if any, are marked with a red cross before the code line.
   The following options are available for working with the query:
   - **Indent code** (): Click to format the code using tab spaces.

- **Remove white space from code** (): Click to format the code by removing the white spaces and displaying the query in a continuous line.
- **Undo** (): Click to undo the last change made.
- **Redo** (): Click to redo the last change made.
- **Clear** (): Click to clear the query text.

6. Click **Run** to execute the query.
7. View the result displayed in the **Query Response** field.
   The following options are available to work with the output:
   - **Expand all fields** (): Click to expand all fields in the result.
   - **Collapse all fields** (): Click to collapse all fields in the result.
   - **Switch Editor Mode** (

): Click to select the editor mode. The following options are available:
     - **View**: Switch to the tree view.
     - **Preview**: Switch to the preview mode.
   - **Copy** (): Click to copy the contents of the output to the clipboard.
   - **Search fields and values** (): Search for the required text in the output.
   - **Maximize** (): Click to maximize the **Query Response** field. Click **Minimize** () to minimize the field to the original size when maximized.
8. Click **Save** to save the report and return to the **Reports** screen.

7.2.4 Saving the Report Results

Use the download option to save the results of a report. When you perform the steps for saving the file, the report is run and the results are exported to a file. You can then use the file saved for further analysis or for archiving as per your requirements.

1. In Analytics, navigate to **Reporting** > **Reports**.
   The **Reports** screen is displayed.
2. Navigate to locate the report that you require. You can use the search field to narrow the list of reports displayed.
3. Click the **Download** () icon.
   The **Download** screen is displayed.

Figure 7-7: Download Screen

4. Select the output required from the **File type** list.
5. Click **Download**.
6. Specify a file name and click **Save** to save the results of the report.

7.2.5 Example Queries

You can modify and use the following examples to build your reports.

- The following query returns all successful protect operations. You can replace the value *protect* with *unprotect* or *reprotect* to modify the query as required.

```
{
    "size": 0,
    "query": {
        "bool": {
            "must": [
                {
                    "term": {
                        "level": "success"
                    }
                },
                {
                    "term": {
                        "protection.operation": "protect"
                    }
                }
            ]
        }
    },
    "aggs": {
        "numberOfOperation": {
            "sum": {
                "field": "cnt"
            }
        }
    }
}
```

- The following query returns all successful protect operations for *user1*. You can replace the value *protect* with *unprotect* or *reprotect* to modify the query as required.

```
{
    "size": 0,
    "query": {
        "bool": {
            "must": [
                {
                    "term": {
                        "level": "success"
```

```
                    }
                },
                {
                    "term": {
                        "protection.operation": "protect"
                    }
                },
                {
                    "term":{
                        "client.username": "user1"
                    }
                }
            ]
        }
    },
    "aggs": {
        "numberOfOperation": {
            "sum": {
                "field": "cnt"
            }
        }
    }
}
```

- The following query returns the monthly metering details.

```
{
  "size": 0,
  "query": {
    "bool": {
      "must": [
        {
          "term": {
            "level": "success"
          }
        },
        {
          "term": {
            "logtype": "protection"
          }
        }
      ]
    }
  },
  "aggs": {
    "meteringPUR": {
      "date_histogram": {
        "field": "origin.time_utc",
        "interval": "month"
      },
      "aggs": {
        "protectOperation": {
          "filter": {
            "term": {
              "protection.operation": "protect"
            }
          },
          "aggs": {
            "protectCount": {
              "sum": {
                "field": "cnt"
              }
            }
          }
        },
        "unprotectOperation": {
          "filter": {
            "term": {
              "protection.operation": "unprotect"
            }
          },
          "aggs": {
            "unprotectCount": {
```

```
              "sum": {
                "field": "cnt"
              }
            }
          }
        },
        "reprotectOperation": {
          "filter": {
            "term": {
              "protection.operation": "reprotect"
            }
          },
          "aggs": {
            "reprotectCount": {
              "sum": {
                "field": "cnt"
              }
            }
          }
        }
      }
    }
  }
}
```

- The following query returns the top 10 users by the operations performed.

```
{
  "size": 0,
  "query": {
    "bool": {
      "must": [
        {
          "term": {
            "level": "success"
          }
        }
      ]
    }
  },
  "aggs": {
    "topTenDE": {
      "terms": {
        "field": "client.username.keyword"
      },
      "aggs": {
        "numberOfOperation": {
          "sum": {
            "field": "cnt"
          }
        }
      }
    }
  }
}
```

- The following query returns the top 10 data elements by the operations performed.

```

{
  "size": 0,
  "query": {
    "bool": {
      "must": [
        {
          "term": {
            "level": "success"
          }
        }
      ]
    }
  },
  "aggs": {
```

```
    "topTenDE": {
      "terms": {
        "field": "protection.dataelement.keyword"
      },
      "aggs": {
        "numberOfOperation": {
          "sum": {
            "field": "cnt"
          }
        }
      }
    }
  }
}
```

Chapter 8

Information Lifecycle Management (ILM)

8.1 Exporting Logs
8.2 Importing Logs
8.3 Deleting Indexes

The Protegrity Data Security Platform enforces security policies at many protection points throughout an enterprise and sends logs to the Audit Store. The logs are stored in a log repository, in this case the Audit Store. Using the Information Lifecycle Management (ILM), you can manage the log repository. These logs are then available for reporting.

The following figure shows the ILM system components and the workflow.

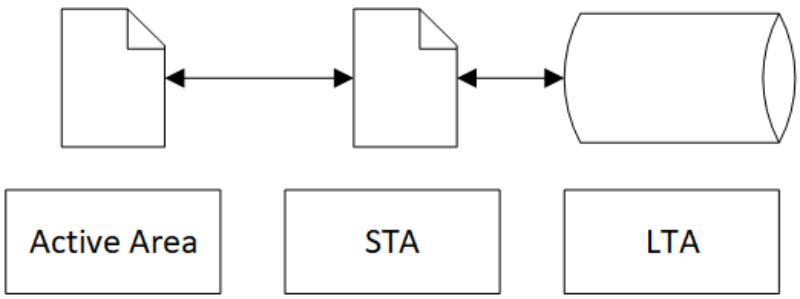

Figure 8-1: ILM System Components and Workflow

The ILM log repository is divided into the following parts:

- Active logs that may be required for immediate reporting or Forensics. These logs are accessed regularly for high frequency reporting.
- Logs that are pushed to Short Term Archive (STA). These logs are accessed occasionally for moderate reporting frequency.
- Logs that are pushed to Long Term Archive (LTA). These logs are accessed rarely for low reporting frequency. The logs are stored where they can be backed up by the backup mechanism used by the enterprise.

The ILM feature in Protegrity Analytics allows you to archive the log entries from the index. Use the search bar to filter logs that you want to find. Click the **Reset Search** ( ) icon to clear the search filter and view all the entries.

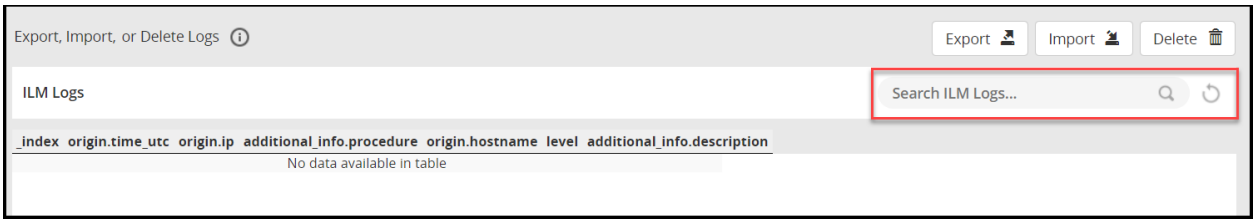

Figure 8-2: ILM Search

You can move entries out of the index when not required and import them back into the index when required. The ILM screen is shown in the following figure.

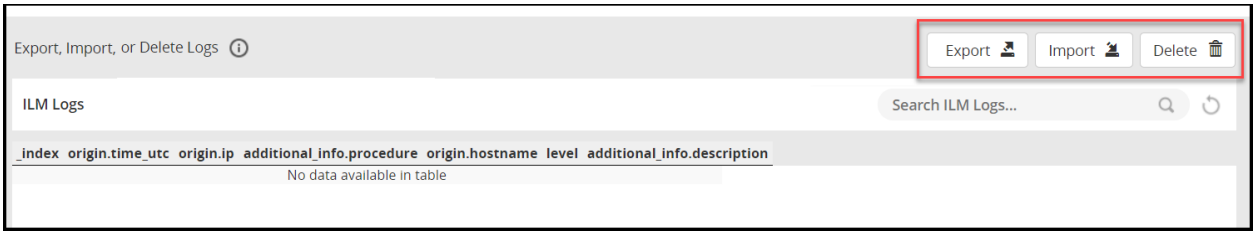

Figure 8-3: ILM screen

> **Note:**
> You can run only one operation at a time for each node for exporting logs or importing logs.
>
> The *Viewer* role user or a user with the *viewer* role can only view data on the ILM screen. You need admin rights to use the import, export, migrate, and delete features of the ILM.

8.1 Exporting Logs

As log entries fill the Audit Store, the size of the log index increases. This slows down log operations for searching and retrieving log entries. To speed up these operations, you can export log entries out of the index and store them in an external file. If required, you can import the entries again for audit and analysis.

Moving index entries out of the index file, removes the entries from the index file and places them in a backup file. This backup file is the STA and reduces the load and processing time for the main index. The backup file is created in the `/opt/protegrity/insight/archive/` directory. To store the file at a different location, you need to mount the destination in the */opt/protegrity/insight/archive/* directory. In this case, ensure that you specify the directory name, for example, */opt/protegrity/insight/archive/<directory>*. Also, ensure that the *<directory>* specified already exists inside the *archive* directory.

If the location is on the same drive or volume as the main index, then the size of the index would reduce. However, this would not be an effective solution for saving space on the current volume. To save space, you need to move the backup file to a remote system or into LTA.

> **Note:** You can run only one export operation at a time.

1. From the Analytics screen, navigate to **Repository** > **ILM**.
2. Click **Export**.

   The **Export Data** screen appears.

Figure 8-4: Export Data

3. Complete the fields for exporting the log data from the default index.
   The available fields are:
   - **From Index**: Select the index to export data from.
   - **Password**: Specify the password for securing the backup file.
   - **Confirm Password**: Specify the password again for reconfirmation.
   - **Directory (optional)**: Specify the location to save the backup file.

     > **Note:** If you do not specify a value, then the default directory `/opt/protegrity/insight/archive/` is used.

4. Click **Export**.
5. Specify the root password.
6. Click **Submit**.

The log entries are extracted, then copied to the backup file, and protected using the password. After a successful export, the exported index will be deleted from the Audit Store database.

After the export is complete, you can move the backup file to a different location till you need the log entries again. You can import the entries in the index again for analysis or audit.

8.2 Importing Logs

The exported log entires and secondary indexes are stored in a separate file. If you need these entries for analysis, then you must first import them back into the Audit Store. To be able to import, the archive file should be inside the `archive` directory or within a directory inside the `archive` directory.

Before you begin
Ensure that you have the passwords handy, in case the log entries were exported and protected using password protection.

> **Note:** Ensure that you do not rename the default index file name for this feature to work.
> Imported indexes are excluded and are not exported when the auto-export task is run from the scheduler.

1. From the Analytics screen, navigate to **Repository** > **ILM**.
2. Click **Import**.
   The **Import Data** screen appears.

Figure 8-5: Import Data

3. Complete the fields for importing the log data to the default index or secondary index.

   The available fields are:

   - **File Name**: Select the file name of the backup file.
   - **Password**: Specify the password for the backup file.

4. Click **Import**.

Data will be imported to an index that is named using the file name or the index name.

> **Note:** If you are importing a file which was exported in version 8.0.0.0, then the new index name will be the date range of the entries in the index file using the format `pty_insight_audit_ilm_(from_date)-(to_date)`. For example, `pty_insight_audit_ilm_20191002_113038-20191004_083900`.

8.3 Deleting Indexes

Use the **Delete** option to delete indexes that you no longer require. You can only delete custom indexes that you created and will be listed in the **Source** list.

1. From the Analytics screen, navigate to **Repository** > **ILM**.
2. Click **Delete**.
   The **Delete Index** screen appears.

Figure 8-6: Delete Index

3. Select the index you want to delete from the **Source** list.
4. Select the **Data in the selected index will be permanently deleted. This operation cannot be undone.** check box.
5. Click **Delete**.
   The **Authentication** screen appears.

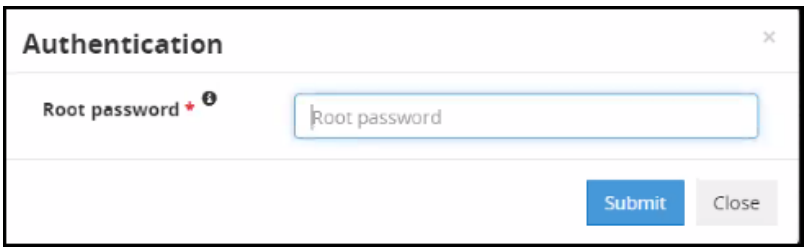

Figure 8-7: Authentication Screen

6. Enter the root password.
7. Click **Submit**.

Chapter 9

Using the Scheduler

9.1 Creating a Scheduled Task

9.2 Working with Scheduled Tasks

9.3 Viewing the Scheduler Monitor

9.4 Viewing the Scheduler Logs

An administrator can execute tasks for ILM, Reporting, and Forensics. These tasks that need to be executed regularly or after a fixed interval can be converted to a scheduled task. This ensures that the task is processed regularly at the set time leaving the administrator free to work on other more important tasks. To view the list of tasks that are scheduled, from the Analytics screen, navigate to **Scheduler** > **Tasks**.

> **Note:** The *viewer* role user or a user with the *viewer* role can only view logs and history related to the Scheduler. You need root or admin rights to create or modify schedules.
>
> Ensure that you disable scheduled tasks on all the nodes before upgrading to a higher version from version 8.0.0.0.

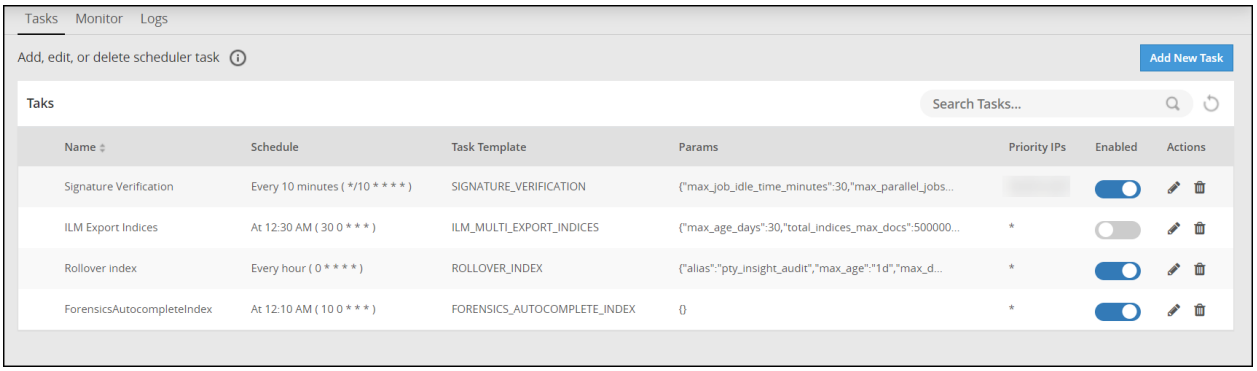

Figure 9-1: Viewing Scheduled Tasks

The list of scheduled tasks are displayed. You can create tasks, view, edit, enable or disable, and modify scheduled task properties from this screen. The following columns are available on this screen.

Column	Description
Name	A unique name for the scheduled task.
Schedule	The frequency set for executing the task.
Task Template	The task template for creating the schedule.
Priority IPs	A list of IP addresses of the machines on which the task must be run.
Params	The parameters for the task that must be executed.

Column	Description
Enabled	Use this toggle switch to enable or disable the task from running as per the schedule.
Action	The actions that can be performed on the scheduled task. The available options are: • Click the **Edit** icon ( ) to update the task. • Click the **Delete** icon ( ) to delete the task.

Note: For the default **ILM Export Indices** task, edit the task, specify the password parameter, and save the task before enabling it for the first time.

9.1 Creating a Scheduled Task

Use the repository scheduler to create scheduled tasks. You can set a scheduled task to run after a fixed interval, every day at a particular time, a fixed day every week, or a fixed day of the month.

Complete the following steps to create a scheduled task.

1. From the Analytics screen, navigate to **Scheduler** > **Tasks**.
2. Click **Add New Task**.
   The **New Task** screen appears.

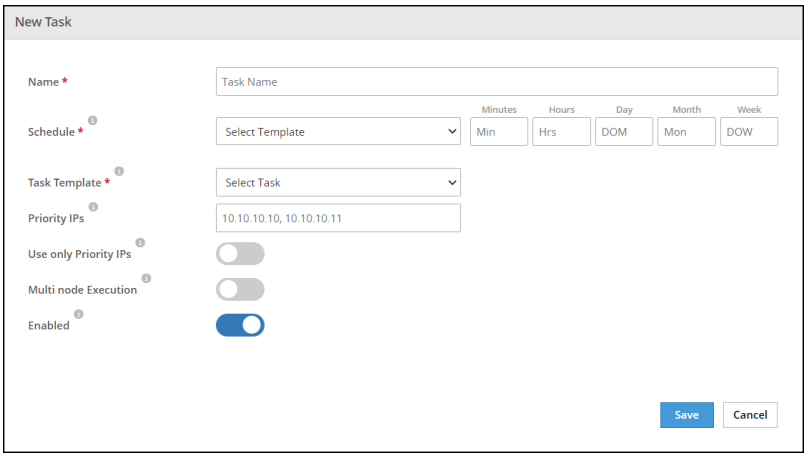

Figure 9-2: New Task

3. Complete the fields for creating a scheduled task.
   The following fields are available:
   - **Name**: Specify a unique name for the task.
   - **Schedule**: Specify the template and time for running the command using cron. The date and time when the command will be run appears in the area below the **Schedule** field. The following settings are available:
     - **Select Template**: Select a template from the list. The following templates are available:
       - **Custom**: Specify a custom schedule for executing the task.
       - **Every Minute**: Set the task to execute every minute.
       - **Every 5 Minutes**: Set the task to execute after every 5 minutes.

- **Every 10 Minutes**: Set the task to execute after every 10 minutes.
- **Every Hour**: Set the task to execute every hour.
- **Every 2 Hours**: Set the task to execute every 2 hours.
- **Every 5 Hours**: Set the task to execute every 5 hours.
- **Every Day**: Set the task to execute every day at 12 am.
- **Every Alternate Day**: Set the task to execute every alternate day at 12 am.
- **Every Week**: Set the task to execute once every week on Sunday at 12 am.
- **Every Month**: Set the task to execute at 12 am on the first day of every month.
- **Every Alternate Month**: Set the task to execute at 12 am on the first day of every alternate month.
- **Every Year**: Set the task to execute at 12 am on the first of January every year.

> **Note:** If you select a template and modify the date and time settings, then the **Custom** template is used.
>
> The scheduler runs only one instance of a particular task. If the task is already running, then the scheduler skips running the task again. For example, if a task is set to run every 1 minute, and the earlier instance is not complete, then the scheduler skips running the task. The scheduled task will be run again at the scheduled time after the current task is complete.

- Date and time: Specify the date and the time when the command must be executed. The following fields are available:
  - **Min**: Specify the time settings in minutes for executing the command.
  - **Hrs**: Specify the time settings in hours for executing the command.
  - **DOM**: Specify the day of the month for executing the command.
  - **Mon**: Specify the month for executing the command.
  - **DOW**: Specify the day of the week for executing the command.

  Some of the fields also accept the following special syntax:

Character	Definition	Fields	Example
,	Specifies a list of values.	All	1, 2, 5, 6.
-	Specifies a range of values.	All	3-5 specifies 3, 4, 5.
/	Specifies the values to skip.	All	*/4 specifies 0, 4, 8, and so on.
*	Specifies all values.	All	* specifies all the values in the field where it is used.
?	Specifies no specific value.	DOM, DOW	4 in the day-of-month field and ? in the day-of-week field specifies to run on the 4th day of the month.
#	Specifies the nth day of the month.	DOW	2#4 specifies 2 for Monday and 4 for 4th week in the month.
L	Specifies the last day in the week or month.	DOM, DOW	7L specifies the last Saturday in the month.
W	Specifies the weekday closest to the specified day.	DOM	12W specifies to run on the 12th of the month. If 12 is a Saturday, then run on Friday the 11th. If 12th is a Sunday, then run on Monday the 13th.

- **Task Template**: Select a task template to view and specify the parameters for the scheduled task. The following task templates are available:
  - **Forensics Autocomplete Index**: This task refreshes the column values in the Audit index for suggestions. This helps in autocompletion while building queries in the **Filter** field on the **Forensics** page.
  - **ILM Multi Export**: This task is used for automatically exporting logs when the criteria specified is fulfilled. It displays the required fields for specifying the criteria parameters for exporting indexes. This task is disabled by default, you need to enable it after it is created.

    > **Note:** You can enable the **Use Only Priority IPs** and specify only Secondary ESA machines in the **Priority IPs** field when you create this task to improve performance.
    >
    > Any indexes imported into ILM are not exported using this scheduled task.

  - **Reporting**: This task is used for automatically generating reports at regular intervals. The reports generated are saved in the `/opt/protegrity/insight/reports` directory. It displays the Reporting-related fields for configuring the scheduled task.

    > **Note:** Ensure that you set the frequency for scheduling the report generation task according to the number of logs that you have, such as setting the frequency more than a day. This is to ensure that a second scheduled task does not start till the first one is complete. You can estimate the time taken to complete a scheduled job by navigating to **Analytics** > **Scheduler** > **History** and viewing the **executed_seconds** field of an earlier scheduled task that you created.

  - **Rollover Index**: This task performs an index rollover on the index referred by the alias, when any of the specified conditions, such as, index age, number of documents in the index, or the index size crosses the specified value.
  - **Signature Verification**: This task runs the signature verification tasks after the time interval that is set. It runs the default signature-related job and the ad-hoc jobs created on the **Signature Verification** tab.
- **Priority IPs**: Specify a list of the ESA IP addresses in the order of priority for execution. The task is executed on the first IP address that is specified in this list. If the IP is not available to execute the task, then the job is executed on the next prioritized IP address in the list.
- **Use Only Priority IPs** : Enable this toggle switch to only execute the task on any one node from the list of the ESA IP addresses specified in the priority field. If this toggle switch is disabled, then the task execution is first attempted on the list of IPs specified in the **Priority IPs** field. If a machine is not available, then the task is run on any machine that is available on the Audit Store cluster which might not be mentioned in the **Priority IPs** field.
- **Multi node Execution**: If disabled, then the task is run on a single machine. Enable this toggle switch to run the task on all available machines.
- **Enabled**: Use this toggle switch to enable or disable the task from running as per the schedule.

4. Specify the parameters for the scheduled task.

   The following fields need to be filled based on the task template selected.

   - **ILM Multi Export**:
     - **Max Days**: The number of days to store indexes. Any index beyond this age is exported. The default age specified is *30 days*.
     - **Max Docs**: The maximum docs present over all the indexes. If the number of docs exceeds this number, then the indexes are exported. The default is *50000000* (50 Million).
     - **Max MB(size)**: The maximum size of the index in MB. If the size of the index exceeds this number, then the index is exported. The default is *50000* (50 GB).
     - **File password**: The password for the exported file. The password is hidden.

       > **Note:** Keep the password safe. If you lose the password, then you cannot retrieve it.

     - **Retype File password**: The password confirmation for the exported file.

- **Dir Path**: The directory for storing the exported index in the default path. The default path specified is */opt/protegrity/insight/archive/*. You can specify and create nested folders using this parameter. Also, if the directory specified does not exist, then the directory is created in the */opt/protegrity/insight/archive/* directory.

> **Note:** You can specify one or multiple options for the **Max Days**, **Max Docs**, and **Max MB(size)** parameters.

The fields for ILM entries is shown in the following figure.

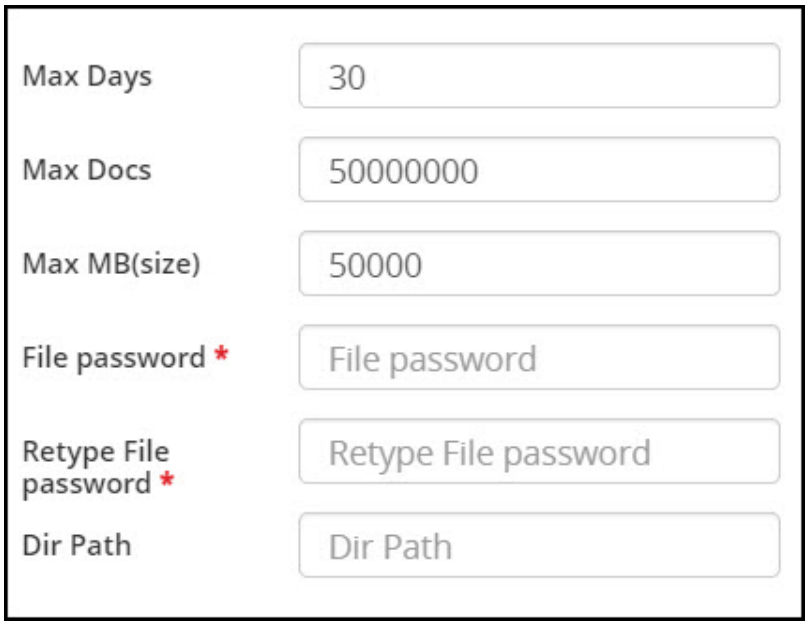

Figure 9-3: ILM Multi Export Entries

- **Reporting**
  - **Report ID**: This is the ID of the report to be exported. This value can be retrieved from the report listing on the **Reporting** > **Reports** page.
  - **File Type**: The type of output file. Acceptable values are json and csv. The default is *json*.
  - **File name Prefix**: The prefix for the exported report. The default is *myReport_*.
  - **Days**: The number of days data to consider for generating the report. If the value is 0, then the data from all the days will be used for generating the report.

    For example, if the value is 1, then the data from the previous 1 day will be used for generating the report. The default is *0*.

The fields for Reporting entries is shown in the following figure.

Figure 9-4: Reporting Entries

- **Rollover Index**:
  - **Rollover Alias**: The alias name for the index that must be rolled over. The default specified is *pty_insight_audit*. This is the alias used to refer to the audit index.
  - **Max Age**: The maximum age after which the index must be rolled over. This default is *1d*, that is one day.

    The values supported are, *y* for years, *M* for months, *w* for weeks, *d* for days, *h* or *H* for hours, *m* for minutes, and *s* for seconds.

  - **Max Docs**: The maximum number of docs that an index can contain. An index rollover is performed when this limit is reached. The default is *50000000*, that is 50 million.
  - **Max Size**: The maximum index size of the index that is allowed. An index rollover is performed when the size limit is reached. The default is *50gb*.

    The units supported are, *b* for bytes, *kb* for kilobytes, *mb* for megabytes, *gb* for gigabytes, *tb* for terabytes, and *pb* for petabytes.

  The fields for the Rollover Index entries is shown in the following figure.

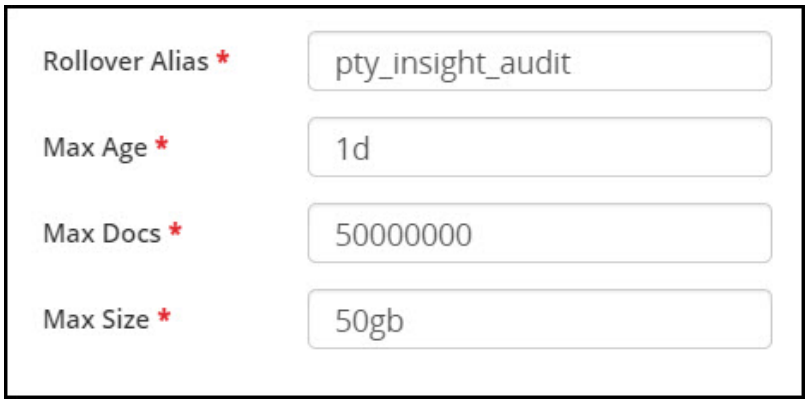

Figure 9-5: Rollover Index Entries

- **Manage Signature Verification Jobs**:
  - **Max Job Idle Time Minutes**: The maximum time to keep the jobs idle. After the jobs are idle for the time specified, the idle jobs are cleared and re-queued. The default specified is *2* minutes.
  - **Max Parallel Jobs Count**: The maximum number of signature verification jobs to run in parallel on the system. If number of jobs specified here is reached, then new scheduled jobs are not started. This default is *4* jobs.

For example, if 10 jobs are queued to run on 2 ESAs, then 4 jobs are started on the first ESA, 4 jobs are started on the second ESA, and 2 jobs will be queued to run till an ESA job slot gets free to accept and run the queued job.

The fields for the Manage Signature Verification Jobs entries is shown in the following figure.

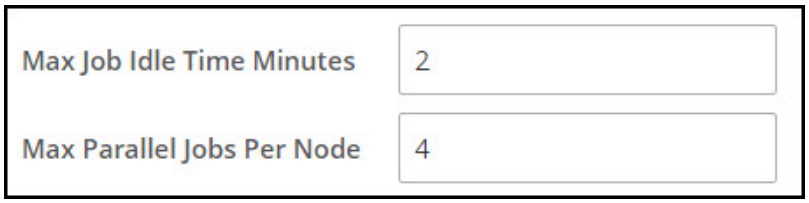

Figure 9-6: Manage Signature Verification Jobs

5. Click **Save**.

The scheduled task is created and enabled. The job executes on the date and time set.

9.2 Working with Scheduled Tasks

After you create a scheduled task, you can control whether the task must be enabled or disabled for running. You can edit the task to modify the commands or the task schedule.

Complete the following steps to modify a task.

1. From the Analytics screen, navigate to **Scheduler** > **Tasks**.
The list of scheduled tasks appears.

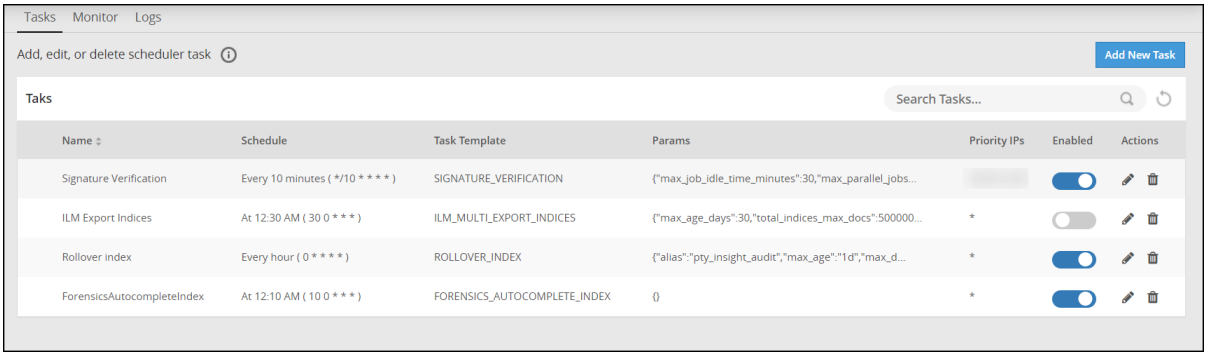

Figure 9-7: Viewing Scheduled Tasks

> **Note:** You can use the search field to search for a specific task from the list.

2. Click the **Enabled** toggle switch to enable or disable the task for running as per the schedule.
Alternatively, clear the **Enabled** toggle switch to prevent the task from running as per the schedule.

3. Click the **Edit** icon ( ) to update the task.
The **Edit Task** page is displayed.

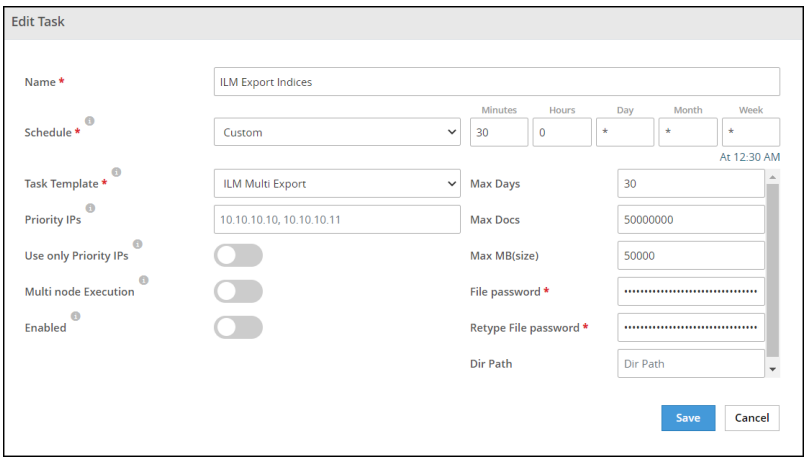

Figure 9-8: Updating Tasks

4. Update the task as required and click **Save**.

The task is saved and run as per the defined schedule.

9.3 Viewing the Scheduler Monitor

The **Monitor** screen shows a list of all the scheduled tasks. It also displays whether the task is running or was executed successfully. You can also stop a running task or restart a stopped task from this screen.

Complete the following steps to monitor the tasks.

1. From the Analytics screen, navigate to **Scheduler** > **Monitor**.
   The list of scheduled tasks appears.

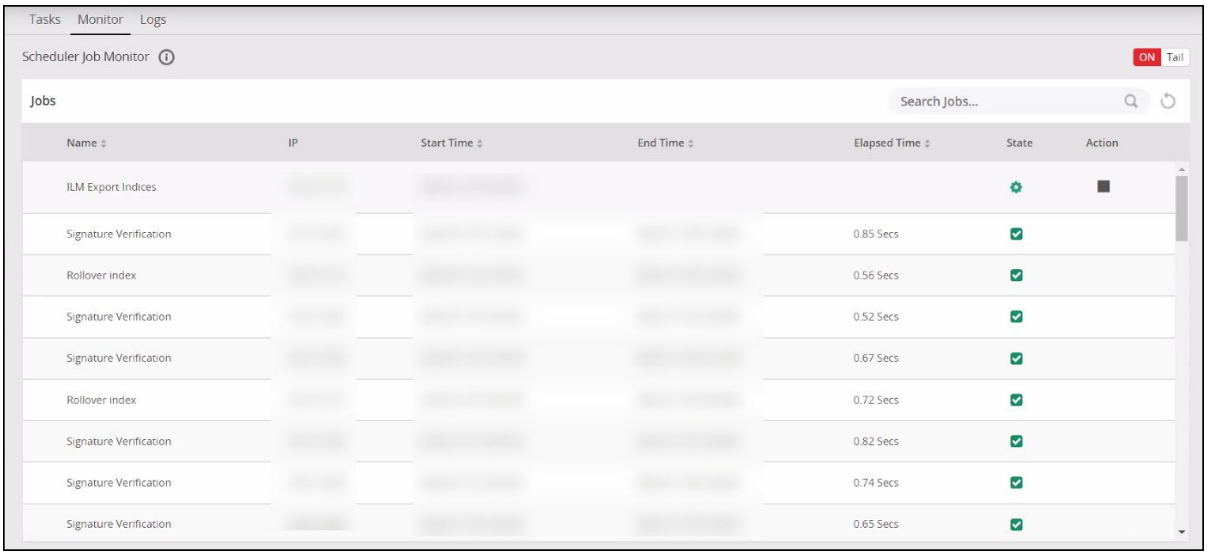

Figure 9-9: Viewing the Task Monitor

> **Note:** You can set the *Tail* option from the upper-right corner of the screen. Setting the *Tail* option to **ON** updates the scheduler history list with the latest scheduled tasks that are run.

> You can use the search field to search for specific tasks from the list.

2. Scroll to view the list of scheduled tasks executed. The following information appears:
   - **Name**: This is the name of the task that was executed.
   - **IP**: This is the host IP of the system that executed the task.
   - **Start Time**: This is the time when the scheduled task started executing.
   - **End Time**: This is the end time when the scheduled task finished executing.
   - **Elapsed Time**: This is the execution time in seconds for the scheduled task.
   - **State**: This is the state displayed for the task. The available states are:
     -  : Running. The task is running. You can click **Stop** from **Actions** to stop the task.
     -  : Queued to stop. The task processing will stop soon.
     -  : Stopped. The task has been stopped.

       > **Note:** The job might take about 20 seconds to stop the process.
       > If an *ILM Multi Export* job is stopped, then the next *ILM Multi Export* job cannot be started within 2 minutes of stopping a previous running job.
       >
       > If a signature verification scheduler job is stopped from the **Scheduler** > **Monitor** page, then the status might be updated on this page after about 5 minutes.

     -  : Completed. The task is complete.
   - **Action**: Click **Stop** to abort the running task. This button is only displayed for tasks that are running.

9.4 Viewing the Scheduler Logs

The **Logs** screen shows a list of all the logs received on the ESA or appliance related to the scheduled task. It shows the information logs for all the scheduled tasks processed on the appliance.

➤ Complete the following steps to view the task logs.

1. From the Analytics screen, navigate to **Scheduler** > **Logs**.
   The logs for the scheduled tasks appears.

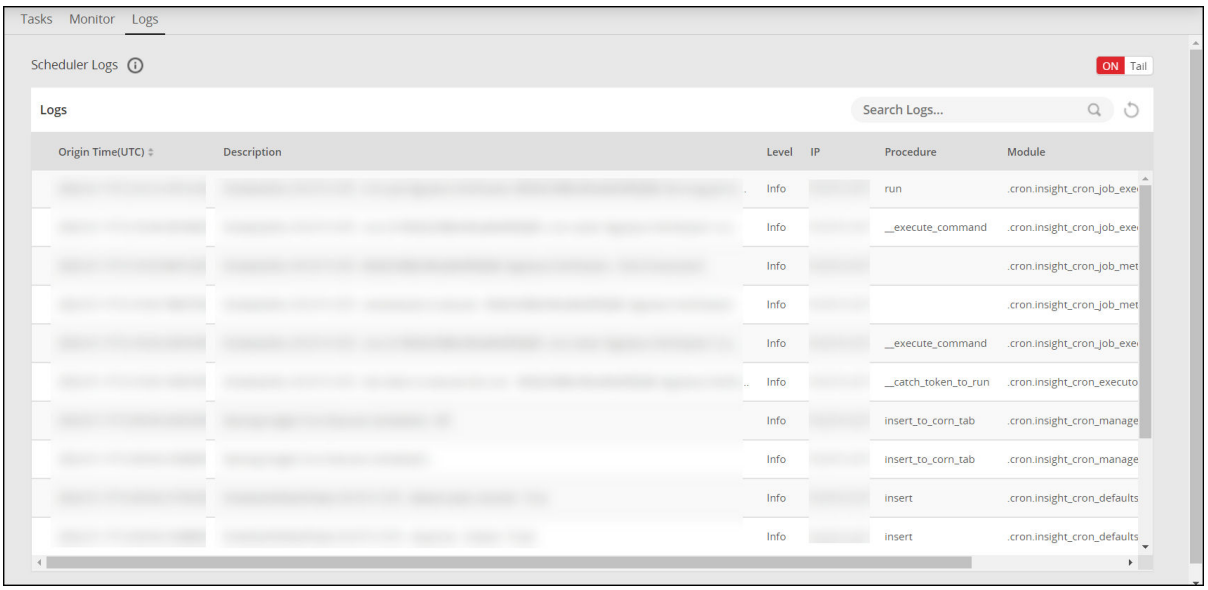

Figure 9-10: Viewing the Task Logs

> **Note:** You can set the *Tail* option from the upper-right corner of the screen. Setting the *Tail* option to **ON** updates the list with the latest data.
>
> You can use the search field to search for specific logs from the list.

2. Scroll to view the logs for the scheduled tasks executed. The following information appears:
   - **Origin Time(UTC)**: This is the time when the scheduled task was processed.
   - **Description**: This is the description of the scheduled task that was executed.
   - **Level**: This is the level of the log message.
   - **IP**: This is the IP address of the system that initiated the request.
   - **Procedure**: This is the procedure that was executed.
   - **Module**: This is the module that was executed.
   - **Host**: This is the hostname of the system that initiated the scheduled task.

Chapter 10

Verifying Signatures

10.1 Working with Signatures
10.2 Creating a Signature Verification Job
10.3 Editing a Signature Verification Job
10.4 Viewing Signature Verification Logs

Logs are generated on the Protectors. The log is then processed using the signature key and a hash value, and a checksum is generated for the log entry. The hash and the checksum is transmitted to the Audit Store for storage and further processing. When the log entry is received by the Audit Store, a check can be performed when the signature verification job is executed to verify the integrity of the logs. The log entries having checksums are identified. These entries are then processed using the signature key and the checksum received in the log entry from the Protector is checked. If both the checksum values match, then the log entry has not been tampered with. If a mismatch is found, then it might be possible that the log entry was tampered or there is an issue receiving logs from a Protector. These can be viewed on the **Forensics** screen by using the following search criteria.

```
logtype = "Verification"
```

The Signature Verification screen allows you to create jobs. These jobs can be run as per a schedule using the scheduler.

For more information about scheduling signature verification jobs, refer to the section *Creating a Scheduled Task*.

To view the list of signature verification jobs created, from the Analytics screen, navigate to **Signature Verification** > **Jobs**.

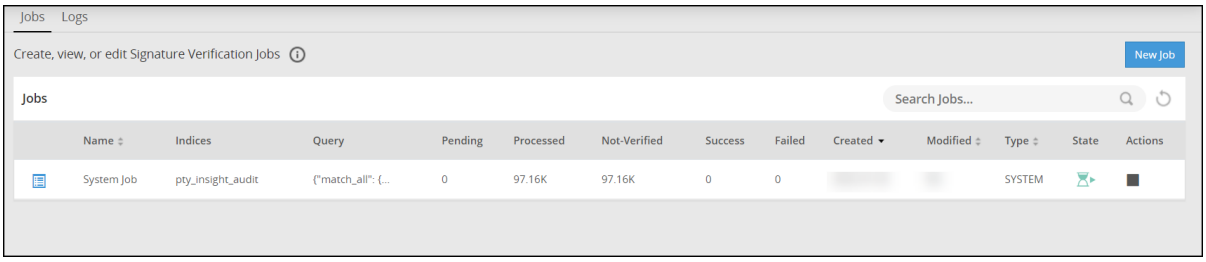

Figure 10-1: Signature Verification

The lifecycle of an Ad-Hoc job is shown in the following figure.

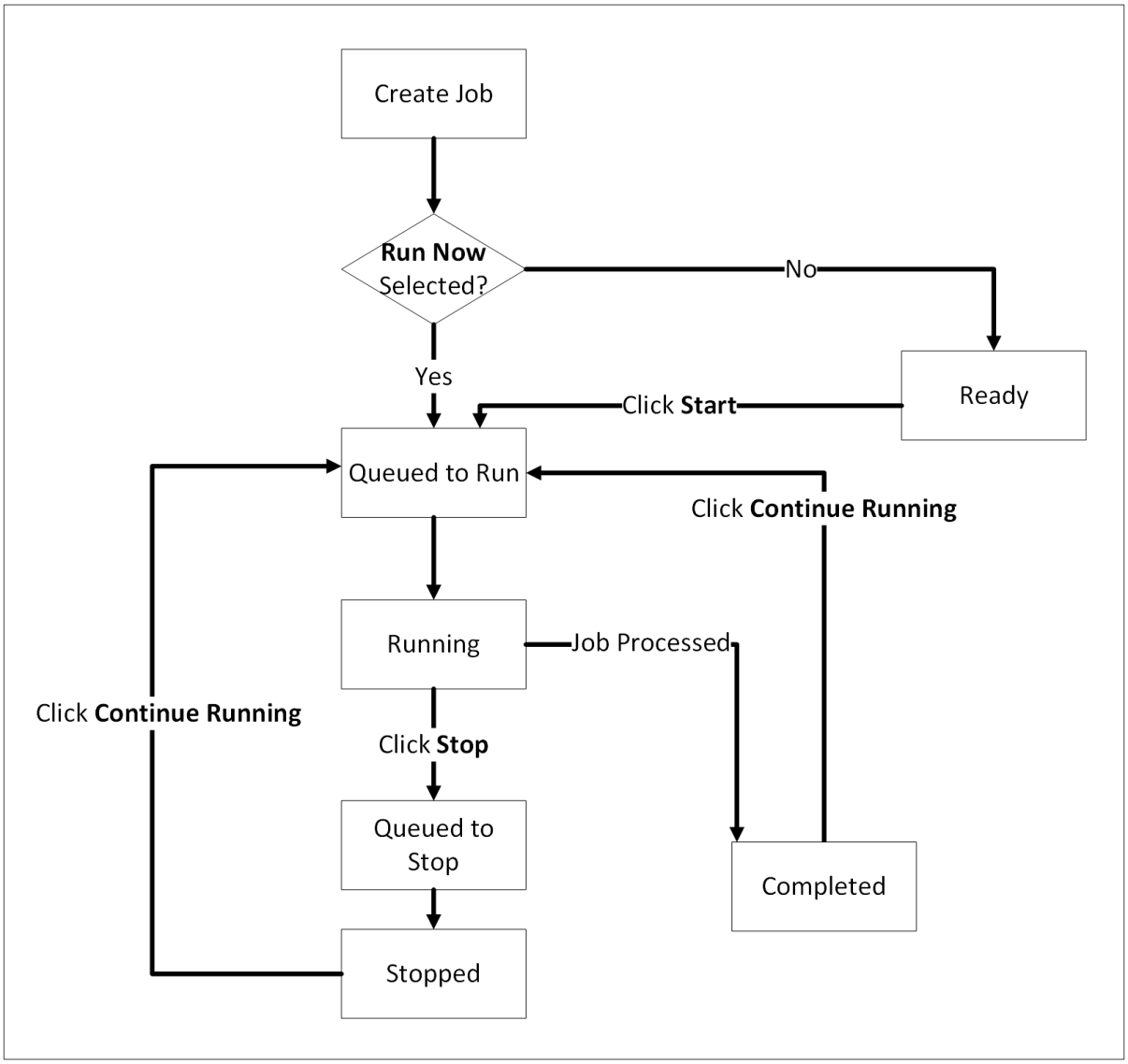

Figure 10-2: Ad-Hoc Job Lifecycle

The Ad-Hoc job lifecycle is described here.

1. A job is created.
2. If **Run Now** is selected while creating the job, then the job enters the *Queued to Run* state.

   If **Run Now** is not selected while creating the job, then the job enters the *Ready* state. The job will only be processed and enters the *Queued to Run* state by clicking the **Start** button.

3. When the scheduler runs, based on the scheduler configuration, the *Queued to Run* jobs enter the *Running* state.
4. After the job processing completes, the job enters the *Completed* state. Click **Continue Running** to move the job to the *Queued to Run* state for processing any new logs generated.
5. If **Stop** is clicked while the job is running, then the job moves to the *Queued to Stop* state, and then moves to the *Stopped* state.
6. Click **Continue Running** to re-queue the job and move the job to the *Queued to Run* state.

A System job is created by default for verifying signatures. This job runs as per the signature verification schedule to processes the audit log signatures.

The logs that fail verification are displayed in the following locations for analysis.

- In Forensics using the query *logtype = "Verification"*.
- On the **Signature Verification** > **Logs** tab.
- On the **Alerting** > **Alerts** dashboard.

When the signature verification for an audit log fails, the failure logs are logged in Forensics. Alerts can be generated by using monitors that query the failed logs in Forensics. A monitor is available by default that is scheduled to run every 30 minutes to generate alerts if there is any failure.

For more information about alerts, refer to the section *Viewing Alerts*.

The lifecycle of a System job is shown in the following figure.

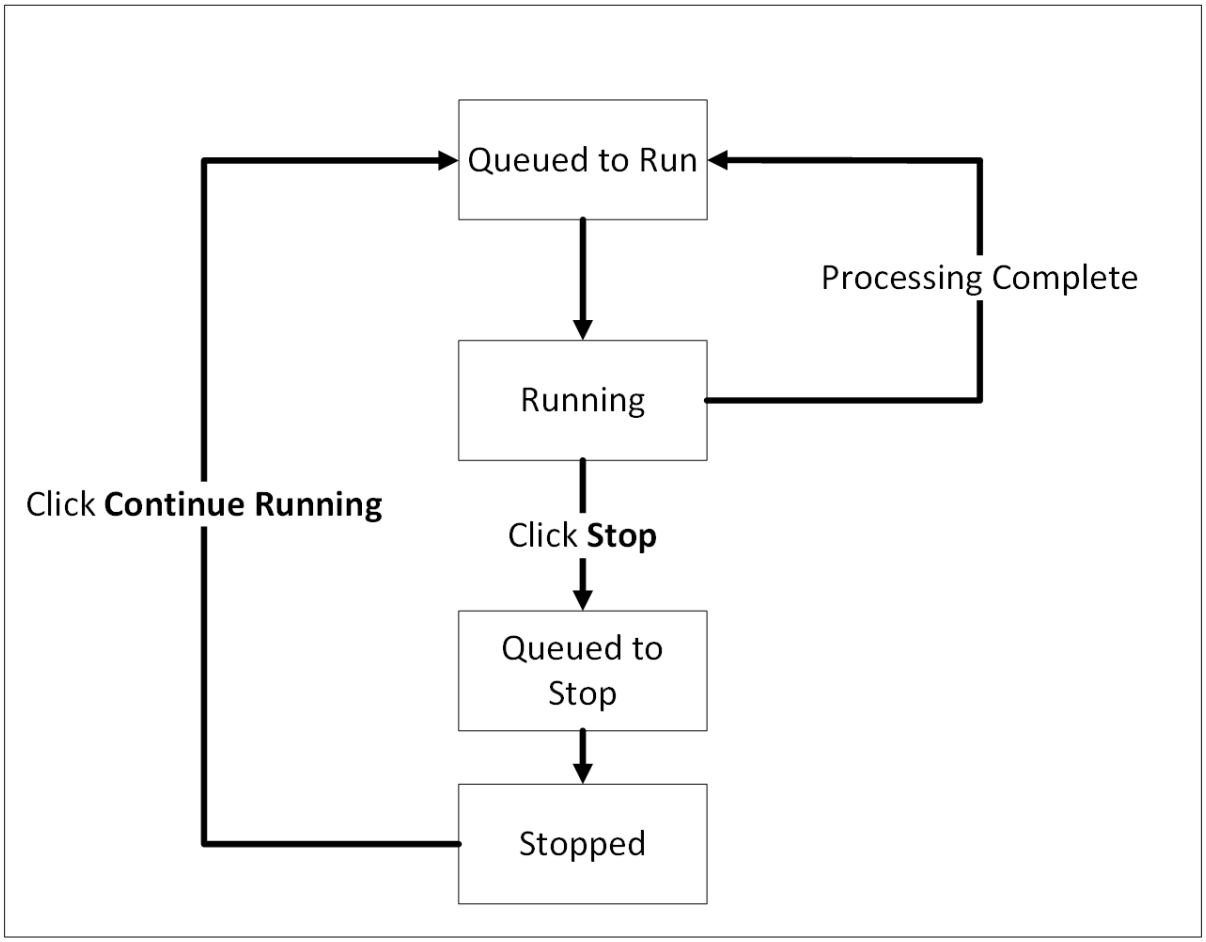

Figure 10-3: System Job Lifecycle

The System job lifecycle is described here.

1. The System job is created when Analytics is initialized or the ESA is upgraded and enters the *Queued to Run* state.
2. When the scheduler runs, then the job enters the *Running* state.
3. After processing is complete, then the job returns to the *Queued to Run* state because it is a system job that needs to keep processing records as they arrive.
4. While the job is running, clicking **Stop** moves the job to the *Queued to Stop* state followed by the *Stopped* state.
5. If the job is in the *Stopped* state, then clicking **Continue Running** moves the job to the *Queued to Run* state.

10.1 Working with Signatures

The list of signature verification jobs created is available on the **Signature Verification** tab. From this tab, you can view, create, edit, and execute the jobs. You can also stop or continue a job from this tab.

To view the list of signature verification jobs, from the Analytics screen, navigate to **Signature Verification** > **Jobs**.

> **Note:** The *viewer* role user or a user with the *viewer* role can only view the signature verification jobs. You need admin rights to create or modify signature verification jobs.
>
> After initializing Analytics during a fresh installation, ensure that you update the priority IP list for the default signature verification jobs by editing the task from **Analytics** > **Scheduler** > **Signature Verification Job**. During the upgrade from an earlier version of the ESA, if Analytics is initialized on an ESA, then the ESA will be used for the priority IP, else you need to update the priority IP for the signature verification job after the upgrade is complete. If you have multiple ESAs in the priority list, then more ESAs are available to process the signature verifications jobs that must be processed.
>
> For example, if the max jobs to run on an ESA is set to 4 and 10 jobs are queued to run on 2 ESAs, then 4 jobs are started on the first ESA, 4 jobs are started on the second ESA, and 2 jobs will be queued to run till an ESA job slot gets free to accept and run the queued job.

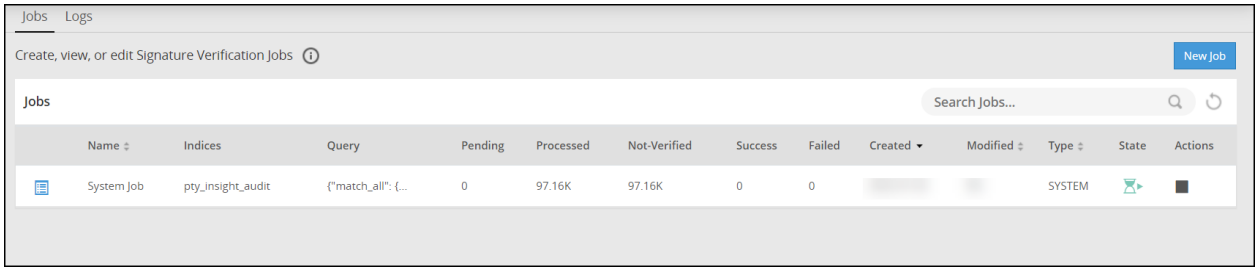

Figure 10-4: Viewing Signature Verification Jobs

You can use the search field to filter and find the verification job that you require. Click the **Reset Search** icon to clear the filter and view all jobs.

The following columns are available on this screen. You can click a label to sort the items in the ascending or descending order. Sorting is available for the **Name**, **Created**, **Modified**, and **Type** columns.

Column	Description
Name	A unique name for the signature verification job.
Indices	A list of indices on which the signature verification job will run.
Query	The signature verification query.
Pending	The number of logs pending for signature verification.
Processed	The current number of logs processed.
Not-Verified	The number of logs that could not be verified. **Note:** Only protector and PEP server logs for version 8.1.0.0 and higher can be verified.
Success	The number of verifiable logs where signature verification succeeded.
Failure	The number of verifiable logs where signature verification failed.
Created	The creation date of the signature verification job.

Column	Description
Modified	The date on which the signature verification job was modified.
Type	The type of the signature verification job. The available options are: • *SYSTEM*: The job is created by the system. • *ADHOC*: The job is a custom job created by a user.
State	Shows the job status. The available statuses are: •  : Queued to run. The job will run soon. •  : Ready. The job will run when the scheduler initiates the job. •  : Running. The job is running. You can click **Stop** from **Actions** to stop the job. •  : Queued to stop. The job processing will stop soon. •  : Stopped. The job has been stopped. You can click **Continue Running** from **Actions** to continue the job. **Note:** If a signature verification scheduler job is stopped from the **Scheduler** > **Monitor** page, then the status might be updated on this page after about *5* minutes. •  : Completed. The job is complete. You can click **Continue Running** from **Actions** to run the job again.
Action	The actions that can be performed on the signature verification job. The available options are: • Click the **Edit** icon () to update the job. • Click the **Start** icon () to run the job. • Click the **Stop** icon () to stop the job. • Click the **Continue Running** icon () to resume the job.

Note: You need root or admin rights to create or modify signature verification jobs.

10.2 Creating a Signature Verification Job

You can specify a query for creating the signature verification job. Additionally, you can select the indices that the signature verification job needs to run on.

1. In Analytics, navigate to **Signature Verification** > **Jobs**.
   The **Signature Verification Jobs** screen is displayed.
2. Click **New Job**.
   The **Create Job** screen is displayed.

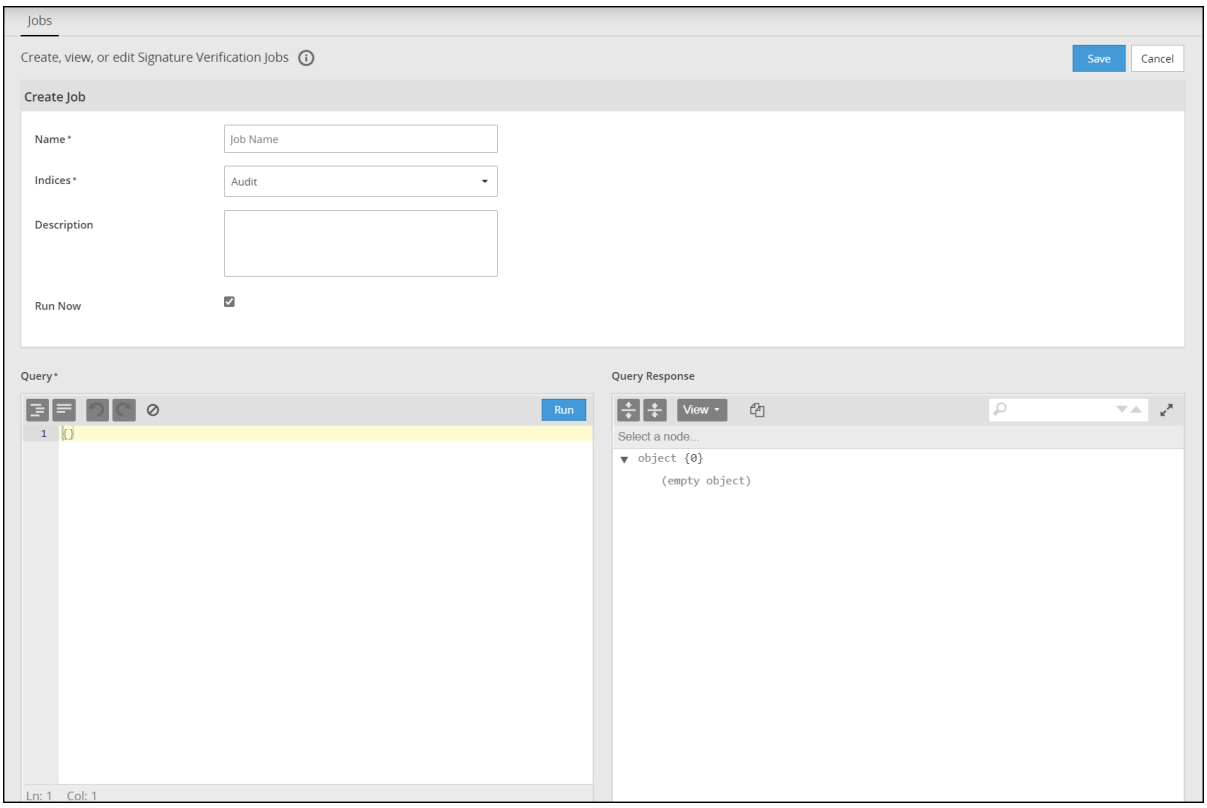

Figure 10-5: Create Job

3. Specify a unique name for the job in the **Name** field.
4. Select the index or alias to query from the **Indices** list. An alias is a reference to one or more indexes available in the **Indices** list. The alias is generated and managed by the system and cannot be created or deleted.
5. Specify a description for the job in the **Description** field.
6. Select the **Run Now** check box to run the job after it is created.
7. Use the **Query** field to specify a JSON query. Errors in the code, if any, are marked with a red cross before the code line.

   The following options are available for working with the query:

   - **Indent code** ( ): Click to format the code using tab spaces.
   - **Remove white space from code** ( ): Click to format the code by removing the white spaces and displaying the query in a continuous line.
   - **Undo** ( ): Click to undo the last change made.
   - **Redo** ( ): Click to redo the last change made.
   - **Clear** ( ): Click to clear the query text.

> **Note:** You can specify the contents of the *query* tag for creating the JSON query. For example, you would specify the query

```
{
   "query":{
      "match" : {
         "field_name":"field_value"
      }
   }
}
```

as

```
{
      "match" : {
         "field_name":"field_value"
      }
   }
```

.

8. Click **Run** to test the query.
9. View the result displayed in the **Query Response** field.
   The following options are available to work with the output:
   - **Expand all fields** (): Click to expand all fields in the result.
   - **Collapse all fields** (): Click to collapse all fields in the result.
   - **Switch Editor Mode** (

): Click to select the editor mode. The following options are available:
     - **View**: Switch to the tree view.
     - **Preview**: Switch to the preview mode.
   - **Copy** (): Click to copy the contents of the output to the clipboard.
   - **Search fields and values** (): Search for the required text in the output.
   - **Maximize** (): Click to maximize the **Query Response** field. Click **Minimize** () to minimize the field to the original size when maximized.
10. Click **Save** to save the job and return to the **Signature Verification Jobs** screen.

10.3 Editing a Signature Verification Job

You can edit an adhoc signature verification job to update the name and the description of the job.

1. In Analytics, navigate to **Signature Verification** > **Jobs**.
   The **Signature Verification Jobs** screen is displayed.
2. Locate the job that you want to update.
3. From the **Actions** column, click the **Edit** () icon.
   The **Job** screen is displayed.

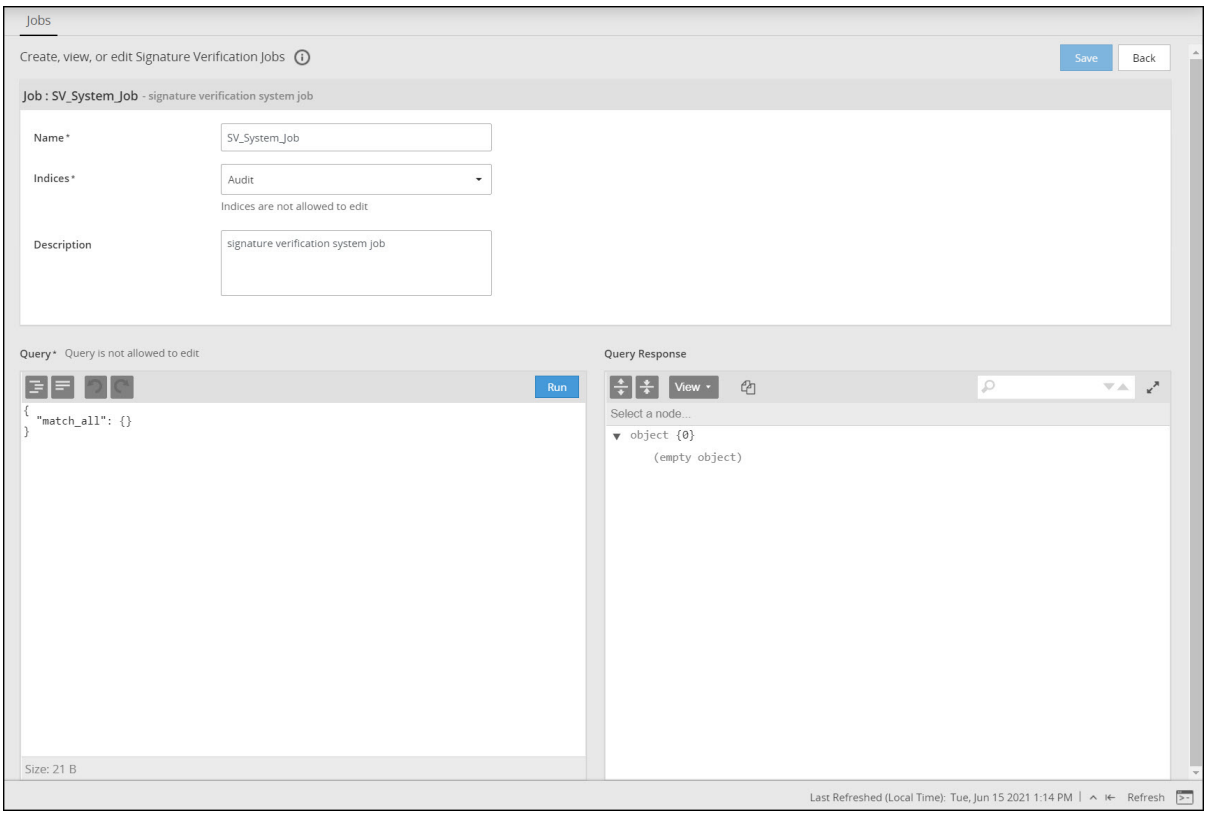

Figure 10-6: Job Screen

4. Update the name and description as required.

> **Note:** The **Indices** and **Query** options can be edited if the job is in the *Ready* state, else they are available in the read-only mode.

5. View the JSON query in the **Query** field.
   The following options are available for working with the query:
   - **Indent code** (): Click to format the code using tab spaces.
   - **Remove white space from code** (): Click to format the code by removing the white spaces and displaying the query in a continuous line.
   - **Undo** (): Click to undo the last change made.
   - **Redo** (): Click to redo the last change made.
6. Click **Run** to test the query, if required.
7. View the result displayed in the **Query Response** field.
   The following options are available to work with the output:
   - **Expand all fields** (): Click to expand all fields in the result.
   - **Collapse all fields** (): Click to collapse all fields in the result.
   - **Switch Editor Mode** (): Click to select the editor mode. The following options are available:
     - **View**: Switch to the tree view.
     - **Preview**: Switch to the preview mode.

- **Copy** (): Click to copy the contents of the output to the clipboard.
- **Search fields and values** (): Search for the required text in the output.
- **Maximize** (): Click to maximize the **Query Response** field. Click **Minimize** () to minimize the field to the original size when maximized.

8. Click **Save** to update the job and return to the **Signature Verification Jobs** screen.

10.4 Viewing Signature Verification Logs

You can view a list of failed records on this tab. Additionally, log information for the signature verification job is also displayed on this tab. To view the logs for the signature verification jobs, from the Analytics screen, navigate to **Signature Verification** > **Logs**.

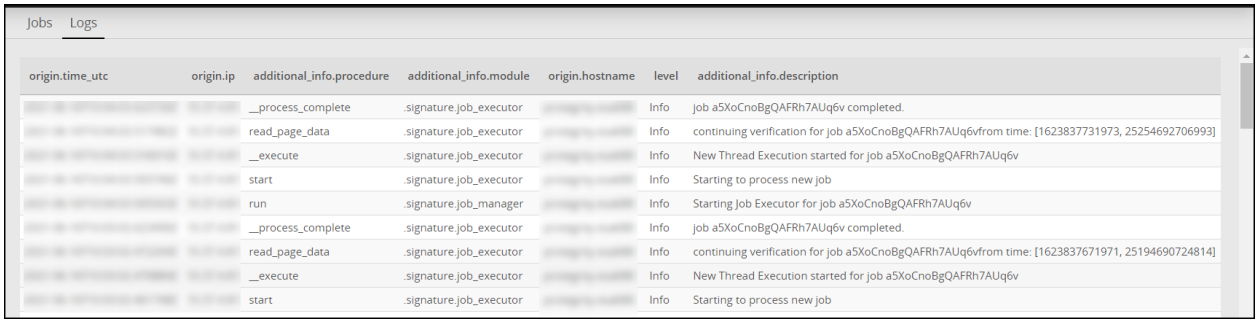

Figure 10-7: Viewing Signature Verification Logs

The following columns are available on this screen.

Column	Description
origin.time_utc	The origin time for the log in UTC.
origin.ip	The IP of the system where the log was generated.
additional_info.procedure	The procedure name that executed.
additional_info.module	The module that generated the log.
origin.hostname	The hostname of the machine where the log was generated.
level	The level of the log.
additional_info.description	The description for the job.

Appendix A

Troubleshooting

11.1 Known Issues for Protegrity Analytics

This section describes the problems that you might face while working with logging and the solutions or workarounds to resolve those problems.

11.1 Known Issues for Protegrity Analytics

A list of known issues with their solution or workaround is provided here. The steps provided to resolve the known issues ensure that your product does not throw errors or crash.

- **Known Issue:** The Audit Store node remains uninitialized and the message *Open Distro Security not initialized.* appears.

  Issue:

  When you stop the Audit Store while running the leave_cluster operation, then the security on the Audit Store node might remain uninitialized. In this case, the security service for the Audit Store does not get re-initialized after starting or restarting the Audit Store and the Audit Store is not accessible.

  Resolution:

  Perform the following steps to re-enable security.

  1. From the ESA Web UI, navigate to **System** > **Services** > **Audit Store**.
  2. Start the Audit Store Repository service.
  3. Open the ESA CLI.
  4. Navigate to **Tools**.
  5. Click **Apply Audit Store Security Configs**.

     The security service on the Audit Store node is started.

- **Known Issue:** In **Alerts**, an incorrect evaluation in the monitor, trigger, or alert query fills the alert dashboard with multiple alerts.

  Issue:

  An incorrect query shows up as an error alert on the alert dashboard. However, the same issue appears multiple times on the alert dashboard every time the monitor is scheduled to run.

Resolution:

Fix this issue by correcting the query for the monitor, trigger, or alert. Use the test query feature for the monitor, trigger, or alert to ensure that the query is built accurately.

- **Known Issue:** Client side validation is missing on the **Join ES Cluster** page.

  Issue:

  Login to the ESA Web UI and navigate to the Audit Store Management page. When you specify an invalid IP, enter a username or password more than the 36-character limit that is accepted on the Appliance, and click **Join**, then no errors are displayed and the request is processed.

  Observation:

  The **Join ES Cluster** page request is processed without any client-side validation, hence, an invalid IP address or a username or password more than 36 characters does not display any error.

- **Known Issue:** Alerts might keep appearing on the Alerting dashboard after restarting the Audit Store or the machine.

  Issue:

  After restarting the Audit Store or the machine, an alert might be raised every time the monitor is scheduled to run and the trigger condition is met. This causes multiple alerts being raised on the dashboard for a monitor. These duplicate alerts also increase the index size.

  Workaround:

  To avoid this issue, disable all monitors before restarting the machine or the Audit Store. Enable the monitors again after the restart is complete. Additionally, if multiple alerts are raised after restarting the machine or the Audit Store, then disable all the monitors and enable the monitors again.

  To enable or disable a monitor:

  1. From the ESA Web UI, navigate to **Analytics** > **Alerting** > **Monitors**.
  2. Click the **Edit** () icon to enter the edit mode for updating the monitor.
  3. Click the **Enabled** slider to enable or disable the monitor.
  4. Click **Update Monitor** to save the changes.

- **Known Issue:** Multiple instances of the InsightCronExecutor process remain in the running state on the system every time the reporting scheduler job runs.

  Issue:

  The scheduler runs a task as per the schedule set. If the scheduled time for generating reports is too short, then the Scheduler might start a scheduled task even before an earlier task is complete. This will result in the system slowing down if too many tasks are running in the memory.

  Workaround:

  Complete the steps provided here to troubleshoot and avoid multiple scheduled tasks from running in the memory.

  Set the frequency of the scheduled task as required, to avoid the same task from running multiple times:

  1. From the ESA Web UI, navigate to **Analytics** > **Scheduler** > **History**.
  2. View the time taken for executing the scheduled task from the **executed_seconds** column.
  3. Navigate to **Analytics** > **Scheduler** > **Tasks**.
  4. Click the **Edit** () icon to enter the edit mode for updating a scheduled task.
  5. Update the schedule accordingly, ensure that you increase the time so that the scheduler does not run the same task till the earlier task is complete.

6. Click **Save**.

Verify the number of scheduled reporting tasks running in the memory. If too many tasks are running, then disable the scheduled task till the earlier tasks complete.

1. Login to the CLI Manager on the ESA or the Appliance.
2. Navigate to **Administration** > **OS Console**.
3. Enter the password for the root user and select **OK**.
4. Run the following command.

   ```
   ps aux | grep "insight_cron_executor"
   ```

   All the instances of scheduler tasks that are running appears. Each instance is displayed using two lines, one line for the OS initiated task and one line for the task that is running.

5. If multiple instances of the task are running, then complete the following steps to disable the required scheduled task.
   a. From the ESA Web UI, navigate to **Analytics** > **Scheduler** > **Tasks**.
   b. Click the **Enabled** () switch to disable the scheduled task.

> **Note:** For more information about Open Distro and Elasticsearch, refer to *https://opendistro.github.io/for-elasticsearch-docs/docs/elasticsearch/*.

- **Known Issue:** High memory usage on the ESA.

  Issue:

  When using the Audit Store, the memory usage is high on the ESA.

  Workaround:

  Complete the steps provided here to reduce the memory usage by updating the memory allocated to the Audit Store on the ESA and transferring scheduled task requests from the Primary ESA to the Secondary ESAs.

  1. Update the memory size for the Audit Store to **4** GB using the **Set Audit Store Repository Total Memory** CLI option.

     For more information about the **Set Audit Store Repository Total Memory** CLI option, refer to the section *Setting the Total Memory for the Audit Store Repository* in the *Audit Store Guide 9.1.0.5*.

  2. Transfer processing of scheduled tasks from all the ESAs to the Secondary ESAs using the following steps:
     a. From the ESA Web UI, navigate to **Analytics** > **Scheduler** > **Tasks**.
     b. Click the **Edit** () icon to enter the edit mode for updating a scheduled task.
     c. Specify the IP addresses of the ESAs in the Audit Store cluster in the **Priority IPs** field. Separate the IP addresses using commas.

        > **Note:** If you remove an ESA from the Audit Store cluster, then remove the IP address from the **Priority IPs** list.

     d. Click the **Use only Priority IPs** () switch to execute from priority IPs.
     e. Click **Save**.
     f. Repeat the steps for all the remaining scheduled tasks, except the *Signature Verification* scheduled task.

- **Known Issue:** Alerting: Email destinations fail after upgrade to release 9.1.0.0

  Issue:

After upgrading from the ESA v9.0.0.0 to the ESA 9.1.0.0, destinations that have the **Email** setting in Alerting fail. No email message is received by the SMTP server and an alert is raised on the **Analytics** > **Alerting** > **Alerts** page. On the **Alerts** page, moving the mouse over the  (Information) icon displays the *Failed running action: java.io.IOException: Failed: HttpResponseProxy(HTTP/1.1 401 Unauthorized [Server: Protegrity Audit Store 9.1...])* message.

Workaround:

Complete the steps provided here to resolve the issue.

1. From the ESA Web UI, navigate to **Analytics** > **Destination**.
2. Click a destination of the type **Email**.
3. Click **Edit**.
4. Change the email addresses in the **To**, **Cc**, and **Bcc** fields with dummy values, such as, *a@a.com*.
5. Click **Update Destination**.
6. Click **Back**.
7. Click the same destination again.
8. Click **Edit**.
9. Update the email addresses in the **To**, **Cc**, and **Bcc** fields with your required values.
10. Click **Update Destination**.

- **Known Issue:** In Analytics, on the Index Lifecycle Management page, after exporting, importing, or deleting an index one of the following scenarios occurs:
  - The index operation performed does not appear in the other operation lists. For example, an exported index does not appear in the import index list.
  - Performing the same operation on the same index again displays an error message.
  - If the index appears in another operation list, performing the operation displays an error message. For example, an exported index appears in the delete index list and deleting the index displays an error.

  Issue:

  After performing an operation, the index list for the export, import, and delete operations does not refresh automatically.

  Workaround:

  Refresh the Index Lifecycle Management page after performing an export, import, or delete operation.

Appendix B

Basics of Audit Store Querying

12.1 Creating Extraction Query Scripts
12.2 Creating Triggers
12.3 Creating Alert Messages
12.4 Sample Queries for Building Monitors

You can use HTTP request parameters for simple search requests. You can also use the Audit Store query domain-specific language (DSL) to create complex and powerful queries. This gives you more control over queries and you can modify queries to provide the information that you require.

For a detailed explanation about writing queries, refer to the OpenDistro documentation at *https://opendistro.github.io/for-elasticsearch-docs/docs/elasticsearch/full-text/*.

12.1 Creating Extraction Query Scripts

Define a query using the Audit Store query DSL that the monitor must execute in the **Extraction Query** field. The query defined should be well formed. If there is an error in the query, then a message appears on the screen for resolving the error. Click the **Run** button while creating the Monitor to run the query and view the results of the query in the **Query Response** field. A few samples with the explanation for the same is provided in this section.

For more information about monitors, refer to the section *Configuring the Monitor*.

The following snippet shows a simple query to retrieve all records.

```
{
   "query":{
      "match_all":{}
   }
}
```

The following snippet shows a query to match a specific field.

```
{
   "query":{
      "match" : {
         "field_name":"field_value"
```

```
        }
    }
}
```

Use the *field_name* and *field_value* for the values that you want to retrieve.

The following snippet shows how you can specify multiple parameters using the **`multi_match`** switch.

```
{
   "query":{
      "multi_match" : {
         "query": "value_to_search",
         "fields": [ "field_to_search_1", "field_to_search_2" ]
      }
   }
}
```

In this scenario, you search the fields *field_to_search_1* and *field_to_search_2* for the value *value_to_search*.

For more information about the fields to use, refer to the section *Fields Logged*.

12.1.1 Using Variables

The query returns a list of variables and values. You can access these values and use them for further analysis and for creating triggers and alert messages. This section lists the variables that you can use for creating triggers and alerts. The *ctx* variable holds all the results of the query. Use **`ctx.`** before the variable name to use the variable value.

For example, to list the error, you would use *ctx.error*. Similarly, you can build up the variable array, to work with an individual value.

For example, **`ctx.results[0].hits.hits[i]._source.http_status_code`** would return the HTTP status code of the source in the first element of the result.

The following variables are made available for working with trigger and alerts:

Table B-1: Available Variables

Variable	Description
`ctx.results`	Returns an array with the result of the query. If no results are returned, then an empty result is returned. The results are returned in the **`ctx.results`** variable. Iterate through the array to view the data contained in the array elements. For example, **`ctx.results[element_number]`** or **`ctx.results[0].hits.total.value`**.
`ctx.monitor`	Includes the following monitor-related return variables: • **`ctx.monitor.name`** • **`ctx.monitor.type`** • **`ctx.monitor.enabled`** • **`ctx.monitor.enabled_time`** • **`ctx.monitor.schedule`** • **`ctx.monitor.inputs`** • **`triggers`** • **`ctx.monitor.last_update_time`**
`ctx.trigger`	Includes the following trigger-related return variables: • **`ctx.trigger.name`**

Variable	Description
	• `ctx.trigger.severity` • `ctx.trigger.condition` • `ctx.trigger.actions`
`ctx.periodStart`	Specifies the beginning UNIX time stamp for the period when the alert was raised.
`ctx.periodEnd`	Specifies the end UNIX time stamp for the period when the alert was raised.
`ctx.error`	Specifies the error message for incorrect trigger queries. If the query is accurate, then this holds the value *Null*.
`ctx.alert`	Specifies the current active alert values in the variables `ctx.alert.id`, `ctx.alert.version`, and `ctx.alert.isAcknowledged`. If no alert is active, then this holds the value *Null*.

Another variable that you can use is the *track_total_hits* variable. You can use this value for comparing hit values.

12.2 Creating Triggers

You can specify the trigger condition in the **Define Condition** field using an Audit Store query. You can code triggers using Painless scripts. Painless is a scripting language that allows you to execute queries faster. It is also more secure and easy to use. It is very similar to the Java or Groovy scripting languages.

For more information about triggers, refer to the section *Configuring the Trigger*.

12.2.1 Basics for Creating Trigger Condition Scripts

In Painless scripting, you can declare variables in the same way that you declare them in Java.

For more information about Painless scripting, refer to *https://www.elastic.co/guide/en/elasticsearch/painless/7.10/painless-guide.html*.

You need to use single quotes to declare strings.

For example, to declare a string with the value Protegrity, you would use the following code:

```
def mystring = 'Protegrity';
```

You can declare and use an array for scripting. In Painless, the array index starts with 0.

```
def mylist = [1,2,3,4];
return mylist[1];
```

In the example, the value 2 is returned.

You use the syntax `doc['field_name'].value` to retrieve values from the table. You can also use the variable `ctx._source.<field-name>` for retrieving values.

In Painless, dates are provided using `ZonedDateTime`, hence, you can use methods, such as, `getYear` and `getDayOfWeek` to work with dates.

12.2.2 Building the Trigger

The trigger query is executed and when the result of the trigger is *true*, then the action for the alert is executed. Use the output values obtained from the extraction query of the monitor to build and evaluate the required trigger condition.

For example, to set the trigger to *true* when your extraction query returns a request, use the following query.

```
{
    "script": {
        "source": "ctx.results[0].hits.total.value > 0",
        "lang": "painless"
    }
}
```

Thus, if your extraction query retrieves the number of unprotect operations performed by a user, then use the following trigger query for the trigger condition to return *true* and run the alert every time the result count is more than 10000.

```
{
    "script": {
        "source": "ctx.results[0].hits.total.value > 10000",
        "lang": "painless"
    }
}
```

You can also use conditional statements to evaluate the trigger condition. Ensure that the result of the query is either *true* or *false* for raising the trigger and executing the alert accordingly. The following snippet is a sample of a conditional statement in the trigger query to activate the trigger when the avg_cpu value is more than 75.

```
{
      "script": {
          "source": "if (ctx.results[0].aggregations.avg_cpu.value > 75) { return true; }",
          "lang": "painless"
    }
}
```

12.3 Creating Alert Messages

You can create an alert message using the **Message Template** field when you create the action for the monitor. The basics for creating alert messages are provided in this section.

For more information about the action, refer to the section *Configuring the Action*.

12.3.1 Coding Basics for Alert Messages

You can use Mustache to code the information that appears in the subject and the message of the alert. Mustache is a logic-less system, because it does not have control statements. However, you can use the tags that it provides to achieve a flow of control for the code execution.

For more information about Mustache templates, see *https://mustache.github.io/mustache.5.html*.

For more information about the list of trigger variables that are valid, see the *Available Variables* section at *https://opendistro.github.io/for-elasticsearch-docs/docs/alerting/monitors/*.

In Mustache, you assign values to a list of variables.

For example, consider the code provided here for saving a few values in variables.

```
{
"Name": "Jack Fleetwood",
"Test_value":"true"
}
```

Now you can use the variable name within double brackets to use the values stored in the variable:

```
{
Hello {{Name}}, How are you.
}
```

The output of the code is provided here:

```
Hello Jack Fleetwood, How are you.
```

To control the flow of the program, you specify the boolean code and the code to execute using the following syntax:

```
{{#Boolean_variable}}
Code_to_execute
{{/Boolean_variable}}
```

For example, consider the code provided here:

```
{{#Test_value}}
I am fine.
{{/Test_value}}
```

If on processing, the variable ***`Test_value`*** holds the value *true*, then the text ***`I am fine`*** will be displayed, else the code will be skipped.

12.3.2 Building the Alert

The alert consists of a subject and message. When you use webhooks for a destination, then the subject and message are sent as a single alert. However, if the destination is an email address, then the alert information is sent as an email. The subject text is processed as the email subject and the message text is processed and sent as the email body.

Use the following sample template for creating an alert message:

```
{
"lang": "mustache",
"source": {
          "subject": "Subject text here",
          "message": "Message text here"
          }
}
```

> **Note:** The *lang*, *source*, and *message* elements are required in the hierarchy. The *subject* element is optional and is used when the *destination* element is set as email. In this case, the *subject* element will be used in the subject of the email.
>
> Specify the value of the *message* element in one line to avoid any JSON errors.

You can use the coding guidelines from the section *Coding Basics for Alert Messages* and the variables from the section *Using Variables* to build and use alerts.

For example, to send an alert with a subject and include details about the alert in the message, such as, the trigger that raised the alert, severity, and period start and end, use the following code.

```
{
"lang": "mustache",
"source": {
          "subject": "Alert raised!",
```

```
          "message": "Alert in Monitor: {{ctx.monitor.name}}.
          - Trigger: {{ctx.trigger.name}}
          - Severity: {{ctx.trigger.severity}}
          - Period start: {{ctx.periodStart}}
          - Period end: {{ctx.periodEnd}}"
          }
}
```

Additionally, you can add HTML formatting to the message by adding *<html></html>* in the *message* tag. Thus, you can format the code as shown in the following example:

```
{
"lang": "mustache",
"source": {
          "subject": "Alert raised!",
          "message": "<html><body><b>Alert in Monitor:</b> {{ctx.monitor.name}}.
          <ul><li><b>Trigger:</b> {{ctx.trigger.name}}</li>
          <li><b>Severity:</b> {{ctx.trigger.severity}}</li>
          <li><b>Period start:</b> {{ctx.periodStart}}</li>
          <li><b>Period end:</b> {{ctx.periodEnd}}</li></ul></body></html>"
          }
}
```

12.4 Sample Queries for Building Monitors

Modify and use the following sample queries to create a monitor. Ensure that you update the extraction query, trigger query, and alert query as per your requirements.

12.4.1 Sample: Alert For Unauthorized Access

The query provided here raises an alert when a user without permissions tries to perform some operations. A warning is logged and the query raises an alert for the warning.

Monitor Query:

```
{
    "query": {
        "bool": {
            "must": [
                {
                    "match": {
                        "level": {
                            "query": "WARNING",
                            "boost": 1
                        }
                    }
                }
            ],
            "filter": [
                {
                    "range": {
                        "origin.time_utc": {
                            "from": "{{period_start}}",
                            "to": "{{period_end}}",
                            "include_lower": true,
                            "include_upper": true,
                            "format": "epoch_millis",
                            "boost": 1
                        }
                    }
                }
            ],
            "adjust_pure_negative": true,
            "boost": 1
```

```
        }
    },
    "aggregations": {}
}
```

Trigger Condition:

```
{
    "script": {
        "source": "ctx.results[0].hits.total.value > 0",
        "lang": "painless"
    }
}
```

Action:

```
{
    "source": {
        "message": "Monitor Name: {{ctx.monitor.name}},\n
                   - Period\n
                   - start: {{ctx.periodStart}}\n
                   - end: {{ctx.periodEnd}}\n
                   - {{#ctx}}{{#results}}{{#hits}}{{#hits}}{{#_source}}
\n
                   - Message: {{#additional_info}}{{description}}{{/additional_info}}
\n
                     User Name: {{#client}}{{username}}{{/client}}
\n
                   - Host Name: {{#origin}}{{hostname}}{{/origin}}{{/_source}}{{/hits}}{{/
hits}}{{/results}}{{/ctx}}",
        "subject": "Unauthorise User is trying to access the system"
    },
    "lang": "mustache"
}
```

12.4.2 Sample: Alert When A Particular User Performs a Large Number of Operations

The query provided here retrieves all unprotect operations performed by a user named *User1*. The trigger checks the number of operations and raises an alert when the operations performed is more than 1000.

> **Note:**
> Update *user1* in the monitor with the name of the user you want to monitor. Also, update *1000* in the trigger and action with the maximum number of operations allowed.

Monitor Query:

```
{
    "query": {
        "bool": {
            "must": [
                {
                    "match": {
                        "client.username": {
                            "query": "Yigal"
                        }
                    }
                },
                {
                    "match": {
                        "protection.operation": {
                            "query": "Unprotect"
                        }
```

```
                    }
                }
            ],
            "filter": [
                {
                    "range": {
                        "origin.time_utc": {
                            "from": "{{period_start}}",
                            "to": "{{period_end}}",
                            "include_lower": true,
                            "include_upper": true,
                            "format": "epoch_millis",
                            "boost": 1
                        }
                    }
                }
            ],
            "adjust_pure_negative": true,
            "boost": 1
        }
    },
    "aggregations": {}
}
```

Trigger Condition:

```
{
    "script": {
        "source": "ctx.results[0].hits.total.value > 1000",
        "lang": "painless"
    }
}
```

Action:

```
{
    "source": {
        "message": "Monitor Name: {{ctx.monitor.name}},\n
                   - Period\n
                   - start: {{ctx.periodStart}}\n
                   - end: {{ctx.periodEnd}}\n
                   - User Name: {{#client}}{{username}}{{/client}} Has unprotected the data
more that 1000",
        "subject": "User has unprotected the data more the than usual"
    },
    "lang": "mustache"
}
```

12.4.3 Sample: Alert When Requests are Raised at Odd Hours

The query provided here retrieves all unprotect operations performed in the past 12 hours. If any operation is performed, then an alert is raised containing the details of the user that performed the operation.

> **Note:**
> For off hour tracking, this monitor must be set to run everyday at 8 am to track operations performed from 8 pm to 8 am.

Monitor Query:

```
{
    "query": {
        "bool": {
            "must": [
```

```
                {
                    "match": {
                        "protection.operation": {
                            "query": "Unprotect",
                            "boost": 1
                        }
                    }
                }
            ],
            "filter": [
                {
                    "range": {
                        "origin.time_utc": {
                            "from": "{{period_start}}",
                            "to": "{{period_start}}||-12h",
                            "include_lower": true,
                            "include_upper": true,
                            "format": "epoch_millis",
                            "boost": 1
                        }
                    }
                }
            ],
            "adjust_pure_negative": true,
            "boost": 1
        }
    },
    "aggregations": {}
}
```

Trigger Condition:

```
{
    "script": {
        "source": "ctx.results[0].hits.total.value > 0",
        "lang": "painless"
    }
}
```

Action:

```
{
    "source": {
        "message": "Monitor Name: {{ctx.monitor.name}},\n
                            - Period\n
                            - start: {{ctx.periodStart}}\n
                            - end: {{ctx.periodEnd}}\n
                            - {{#ctx}}{{#results}}{{#hits}}{{#hits}}{{#_source}}\n
                            - User Name: {{#client}}{{username}}{{/client}} Has unprotect
the data\n
                            - Host Name: {{#origin}}{{hostname}}{{/origin}}{{/_source}}{{/
hits}}{{/hits}}{{/results}}{{/ctx}}",
        "subject": "User has unprotected the data after office hours"
    },
    "lang": "mustache"
}
```

12.4.4 Sample: Alert for User Access from Multiple Nodes

The query provided here retrieves the logs generated and raises an alert if logs are generated on multiple hosts by the same user.

Monitor Query:

```
{
    "size": 0,
    "aggs": {
```

```
        "ips": {
            "terms": {
                "field": "client.username.keyword"
            },
            "aggs": {
                "range_selector": {
                    "filter": {
                        "range": {
                            "origin.time_utc": {
                                "from": "{{period_start}}",
                                "to": "{{period_end}}",
                                "include_lower": true,
                                "include_upper": true,
                                "format": "epoch_millis",
                                "boost": 1
                            }
                        }
                    },
                    "aggs": {
                        "multiple_hosts": {
                            "terms": {
                                "field": "client.ip.keyword",
                                "min_doc_count": 2
                            }
                        }
                    }
                }
            }
        }
    }
}
```

Trigger Condition:

```
{
    "script": {
        "source": "if (ctx.results[0].aggregations.ips.buckets.length > 0 ){for(def ib :
ctx.results[0].aggregations.ips.buckets) {if(ib.range_selector.multiple_hosts.buckets.length
> 0 ) {for (def dcb : ib.range_selector.multiple_hosts.buckets) {if (dcb['doc_count'] > 1 )
{return true;}}}}}",
        "lang": "painless"
    }
}
```

Action:

```
{
    "source": {
        "message": "Monitor Name: {{ctx.monitor.name}},\n
                            - Period\n
                            - start: {{ctx.periodStart}}\n
                            - end: {{ctx.periodEnd}}\n
                            - User Name: {{#client}}{{username}}{{/client}} is operating on
more than 1 working station",
        "subject": "User is using more than 1 machine for security operation"
    },
    "lang": "mustache"
}
```

12.4.5 Sample: Alert for Signature Verification Failures

The query provided here raises an alert when the signature verification on a log being processed fails.

Monitor Query:

```
{
    "query": {
        "bool": {
            "must": [
                {
                    "match": {
                        "logtype": {
                            "query": "Verification"
                        }
                    }
                }
            ],
            "filter": [
                {
                    "range": {
                        "origin.time_utc": {
                            "from": "{{period_start}}",
                            "to": "{{period_end}}",
                            "include_lower": true,
                            "include_upper": true,
                            "format": "epoch_millis"
                        }
                    }
                }
            ],
            "adjust_pure_negative": true
        }
    },
    "aggregations": {}
}
```

Trigger Condition:

```
{
    "script": {
        "source": "ctx.results[0].hits.total.value > 0",
        "lang": "painless"
    }
}
```

Action:

```
{
    "source": {
        "message": "Monitor Name: {{ctx.monitor.name}},\n
                            - Period\n
                            - start: {{ctx.periodStart}}\n
                            - end: {{ctx.periodEnd}}\n
                            - Signature varification failed.",
        "subject": "{{ctx.results[0].hits.total}} signature verifications failed."
    },
    "lang": "mustache"
}
```

Appendix C

Removing Analytics from the Appliance

You can use the steps provided in this section to remove Analytics from the ESA.

> **Caution:** This feature cannot be reinstalled after it is removed.

1. Stop the *td-agent* service using the following steps.
   a. Login to the ESA Web UI.
   b. Navigate to **System** > **Services** > **Misc**.
   c. Stop the **td-agent** service.

   > **Note:**
   > If the *td-agent* service is not visible, then complete the following steps.
   > i. Login to the CLI Manager on the Appliance.
   > ii. Run the *PLUG - Forward logs to Audit Store* tool to set up *td-agent*. Specify *localhost* in the **Target Audit Store Addresses** field during the configuration.
   >
   > For more information about running the *PLUG - Forward logs to Audit Store* tool, refer to the steps provided in the section *Forwarding Logs to a Remote Audit Store*.

2. Set the **Mode** for the following service to **Manual**:
   - **Analytics**
3. Stop the following service:
   - **Analytics**
4. Login as an administrator to the ESA CLI Manager.
5. Navigate to **Administration** > **Add/Remove Services**.
6. Enter the root password and select **OK**.
7. Select **2. Remove already installed applications** and select **OK**.
8. Press the space bar to select the following applications from the list:
   - **Analytics**
   - **PLUG Triggering Agent**
9. Select **OK** to uninstall the applications.
10. Disable the scheduled task, that will no longer work, using the following steps.

a. Login to the Web UI of the Appliance.
b. Navigate to **System** > **Task Scheduler**.
c. Disable the following scheduled task by selecting the task, clicking **Edit**, clearing the **Enable** check box, and clicking **Save**:
   - **Analytics syslog logrotate**

11. Start the td-agent service using the following steps.
    a. Login to the Web UI of the Appliance.
    b. Navigate to **System** > **Services** > **Misc**.
    c. Start the **td-agent** service.