Netsurion | EventTracker®

How to - Configure Microsoft SQL Server to forward logs to EventTracker

EventTracker v9.0 and above

Publication Date: February 4, 2020

Abstract

This guide provides instructions to configure Microsoft SQL server auditing and forward relevant events to EventTracker Enterprise.

Scope

The configurations detailed in this guide are consistent with **EventTracker Enterprise** version 9.x and later, **Microsoft SQL Server 2012 and later** Edition.

Audience

Administrators who want to monitor the **Microsoft SQL Server** using **EventTracker**.

Table of Contents

1. Introduction

Microsoft SQL Server is a relational database management system with several features and services. With this coverage, there is a large surface area for attack and vulnerabilities.

SQL Server auditing is utilized to address requirements for compliance, analyze database actions to troubleshooting problems and investigate suspicious activity.

EventTracker can employ both server audit specifications and extended events to receive relevant events for auditing. Configuration techniques for both methodologies are shown below. Please configure any method of your choice in accordance with your infrastructure and audit requirements. Both techniques are compared below:

Audit Types	Pro's	Con's
Audit Specifications (available in Microsoft SQL Server 2008 or later)	• Alerts are received in real-time. • Events are received in Windows Event Viewer.	• Additional fields like client hostname, client application name are missing.
Extended Events (available in Microsoft SQL Server 2012 or later)	• Additionally, provides a client hostname, client application name and event category fields. • Lightweight and utilizes a few performance resources.	• Alerts are received with a maximum delay of two hours.

EventTracker MSSQL reports provide information about database activities. By using these reports, we can examine the user login success and login failures for further investigation. These reports can track the database changes in the tables, views, procedures, triggers, schema and can track any SQL query errors.

Dashboards display a graphical representation of the database object changes and actions carried out on that object.

Through dashboards, we can also easily track multiple/brute force login failures. Alerts trigger when a user performs any changes on the database, database view, schema, user management etc.

2. Server audit specifications

Auditing, an instance of SQL Server or a SQL Server database involves tracking and logging events that occur on the system. **Below mentioned configuration must be applied on all workstations, where SQL audit is required.**

2.1 Prerequisites

- **Microsoft SQL Server 2008 or later** must be installed.
- **Microsoft SQL Management Studio** for the respective version must be installed.
- **EventTracker Agent 9.x or later** must be installed on the SQL SERVER workstation.

2.2 Enabling logging for logins

1. Open **Microsoft SQL** management studio with appropriate credentials.
2. In **Object Explorer**, right-click on the database server and select **Properties**.

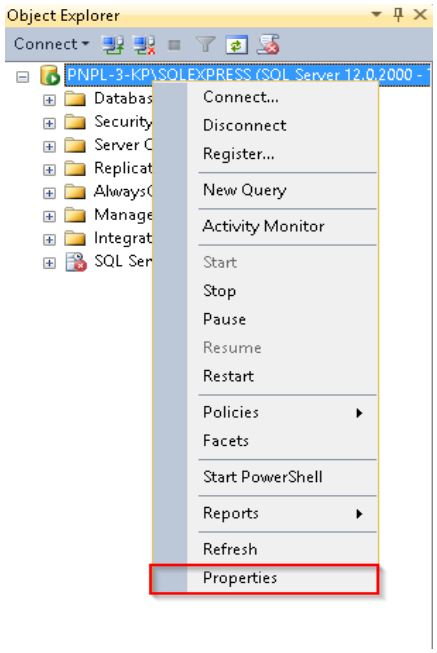

Figure 1

3. In the **Properties** panel, select **Security** in **Select a page** section.
4. In **Login auditing**, select **Both failed and successful logins.**

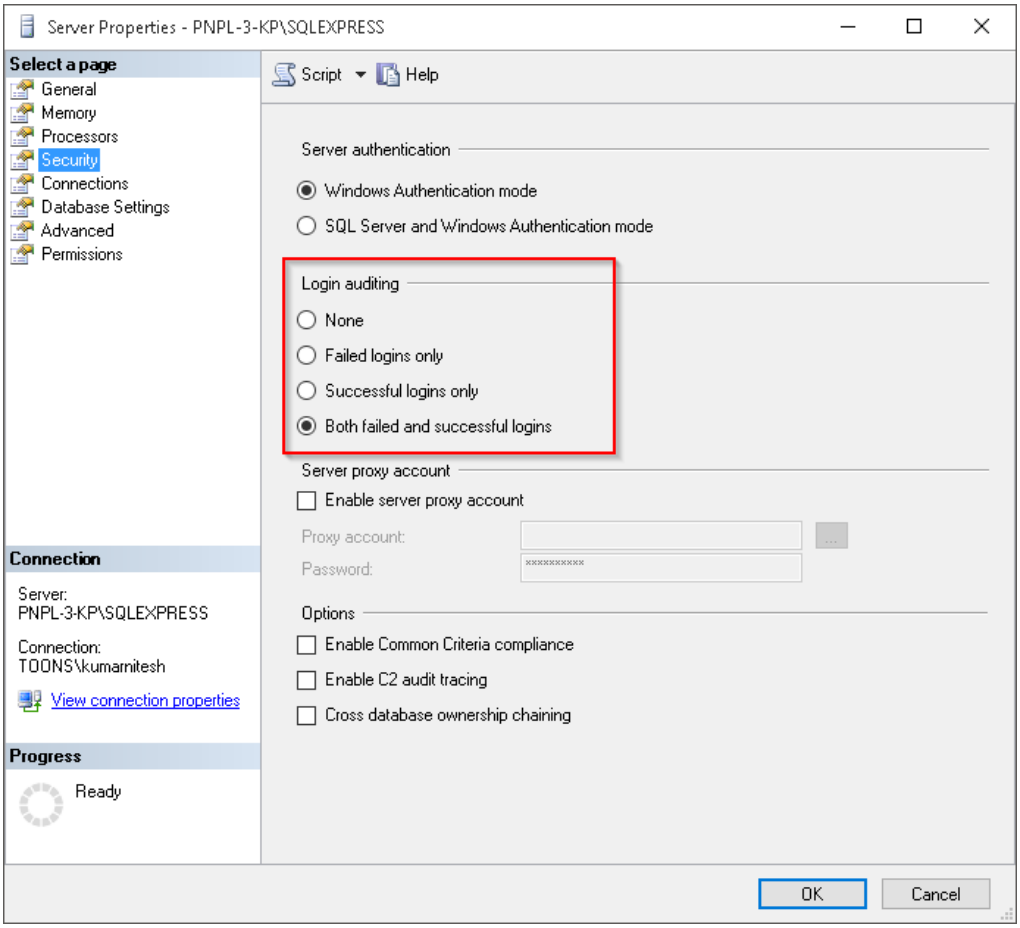

Figure 2

5. The above configuration generates event id "**18453"** (login success) and "**18456**" (login failure).

Note: Login success events are very noisy, enable with caution.

2.3 Enabling server auditing

1. Open **Microsoft SQL management studio** with appropriate credentials.
2. In **Object Explorer**, expand **the Security** tab to view **Audits** and **Server Audit Specifications** options.

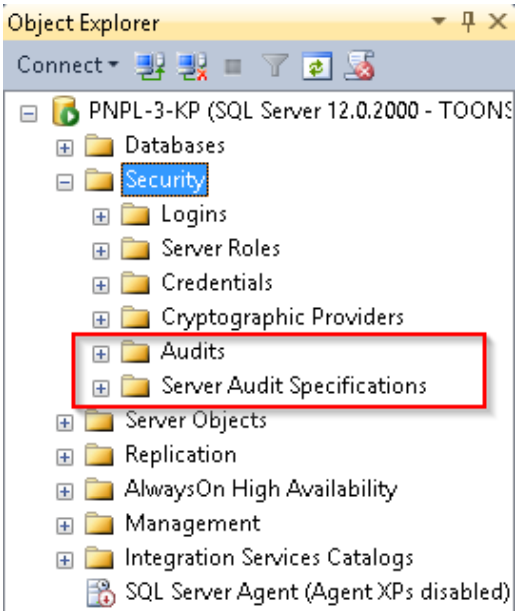

Figure 3

2.3.1 Creating audits

1. Right-click **Audits** to select **New Audit...**.

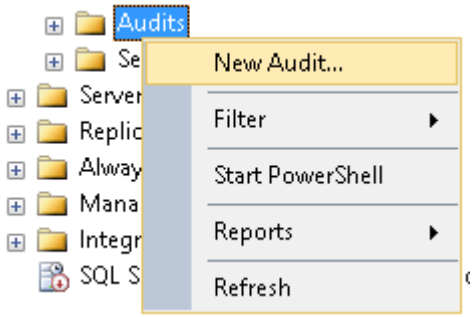

Figure 4

2. In **Audit Properties**, provide appropriate **audit name** and set audit destination as **application log**. The configured Audit properties pane is shown below:

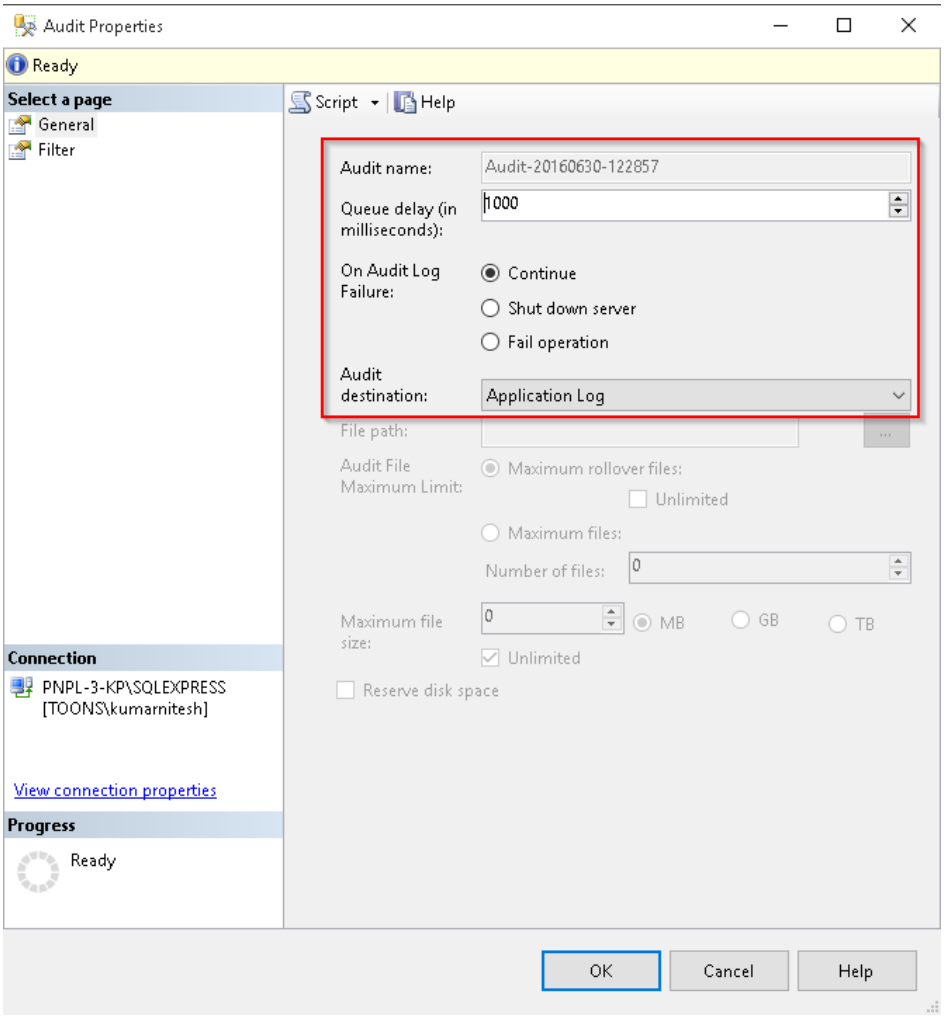

Figure 5

3. Click **OK** to apply settings.

2.3.2 Creating server audit specifications

1. Right-click **Server Audit Specifications** and select **New Server Audit Specification…**

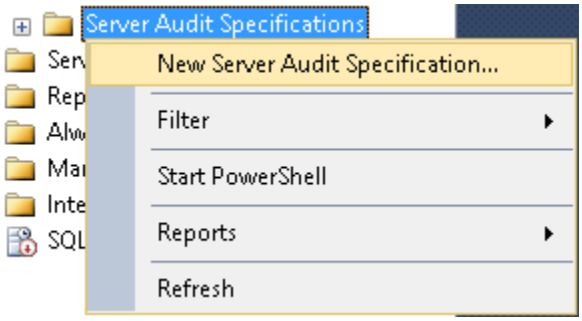

Figure 6

2. In Server **Audit Specification Properties**, provide an appropriate **specification name** and choose an earlier created **audit name** from the drop-down menu.
3. In **Actions** pane, select the following specifications from **Audit Action Type** drop-down.

To improve performance, please enable action types consistent with your audit requirements.

SN	Audit Action Type	Description
1.	**DATABASE_ROLE_MEMBER_CHANGE_GROUP**	This generates events whenever a login is added to or removed from a database role.
2.	**SERVER_ROLE_MEMBER_CHANGE_GROUP**	This generates events whenever a login is added or removed from a fixed server role.
3.	**BACKUP_RESTORE_GROUP**	This generates events whenever a backup or restore command is issued.
4.	**AUDIT_CHANGE_GROUP**	This generates events whenever any audit is created, modified or deleted.
5.	**DATABASE_PERMISSION_CHANGE_GROUP**	This generates events whenever a GRANT, REVOKE, or DENY is issued by any user in SQL Server for database-only events.
6.	**SCHEMA_OBJECT_PERMISSION_CHANGE_GROUP**	This generates events whenever a grant, deny, or revoke is issued for a schema object.
7.	**SERVER_PERMISSION_CHANGE_GROUP**	This generates events when a GRANT, REVOKE, or DENY is issued for permissions in the server scope.
8.	**DATABASE_CHANGE_GROUP**	This generates events when a database is created, altered, or dropped.

9.	**DATABASE_OBJECT_CHANGE_GROUP**	This generates events when a CREATE, ALTER, or DROP statement is executed on database objects.
10.	**DATABASE_PRINCIPAL_CHANGE_GROUP**	This generates events when principals are created, altered, or dropped from a database.
11.	**SCHEMA_OBJECT_CHANGE_GROUP**	This generates events when a CREATE, ALTER, or DROP operation is performed on a schema.
12.	**SERVER_OBJECT_CHANGE_GROUP**	This generates events for CREATE, ALTER, or DROP operations on server objects.
13.	**SERVER_PRINCIPAL_CHANGE_GROUP**	This generates events when server principals are created, altered, or dropped.
14.	**APPLICATION_ROLE_CHANGE_PASSWORD_GROUP**	This generates events whenever a password is changed for an application role.
15.	**LOGIN_CHANGE_PASSWORD_GROUP**	This generates events whenever a login password is changed by ALTER LOGIN statement.
16.	**DATABASE_OWNERSHIP_CHANGE_GROUP**	This generates events when ALTER AUTHORIZATION statement is used to change the owner of a database.
17.	**SCHEMA_OBJECT_OWNERSHIP_CHANGE_GROUP**	This generates events when the permissions to change the owner of schema object (such as a table, procedure, or function) is checked.
18.	**USER_CHANGE_PASSWORD_GROUP****	This generates events whenever the password of a contained database

		user is changed by using the ALTER USER statement.
19	**SUCCESSFUL_LOGIN_GROUP**	This generates events whenever a successful logon is done
20	**LOGOUT_GROUP**	This generates events whenever a logout is done
21	**FAILED_LOGON_GROUP**	This generates events whenever a Logon failure happens.
22	**SERVICE_STATE_CHANGE_GROUP**	This generates events whenever any SQL service is stopped or started.
23	**SERVER_PRINCIPAL_CHANGE_GROUP**	This report will provide details regarding users created, deleted or modified. It will also include events regarding role and permission changes for users.
24	**DATABASE_CHANGE_GROUP**	This report will provide details regarding database startup and shutdowns. It only includes events regarding status change of databases from offline or online.

Note: Only available in Microsoft SQL Server 2012 or later.

Configured Server Audit Specification Properties pane is shown below:

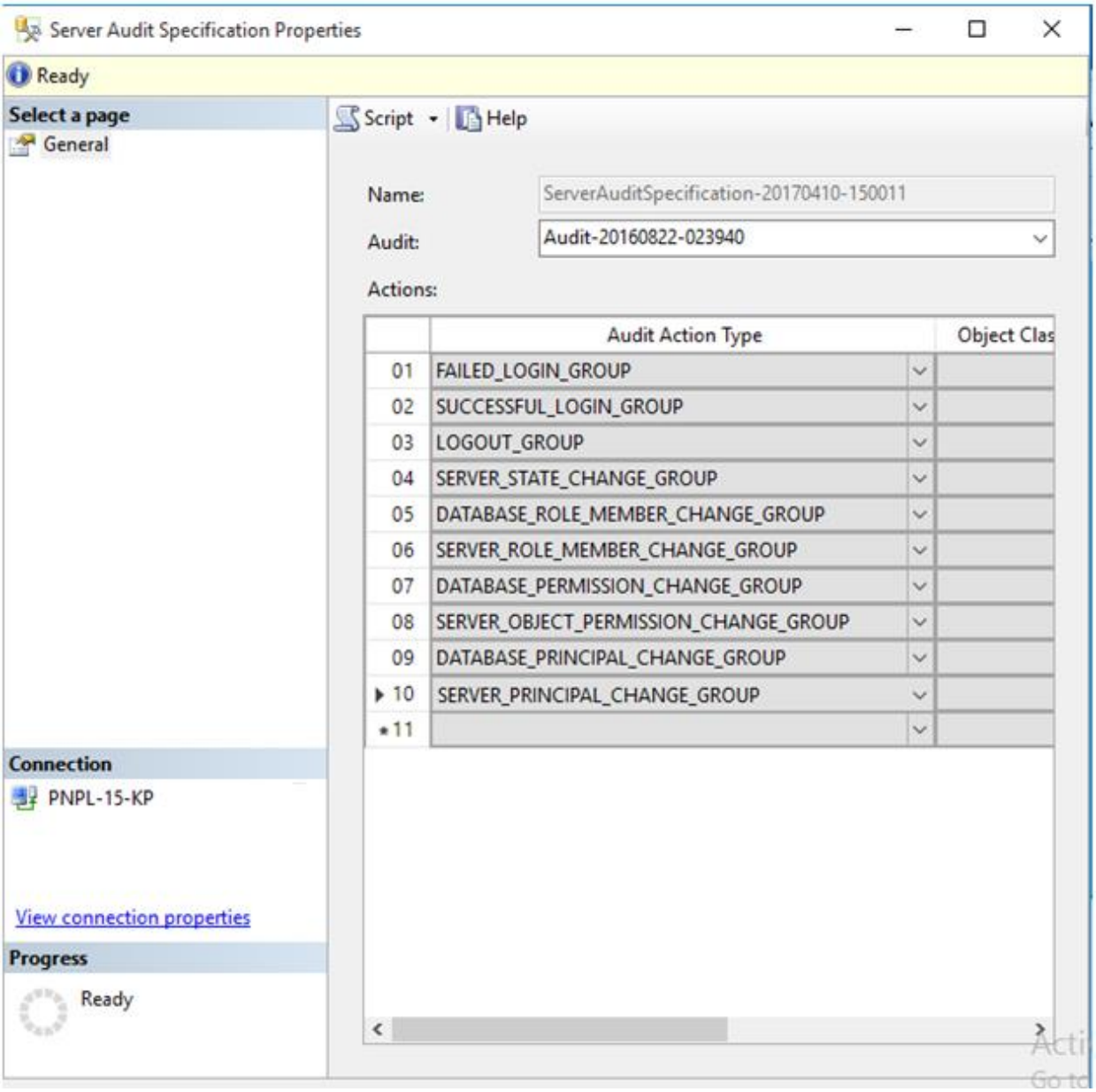

Figure 7

4. Click **OK** to apply settings.
5. Right-click on earlier created **audit** and select **Enable**.

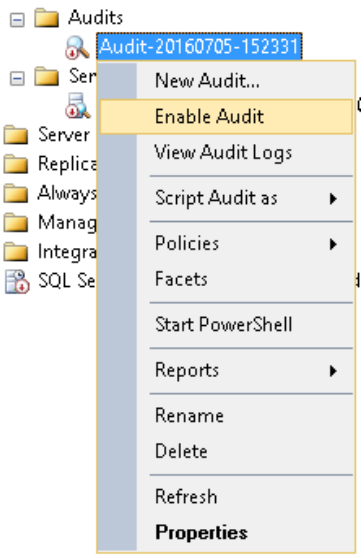

Figure 8

6. Right-click on earlier created **Server Audit Specification** and select **Enable Server Audit Specification**.

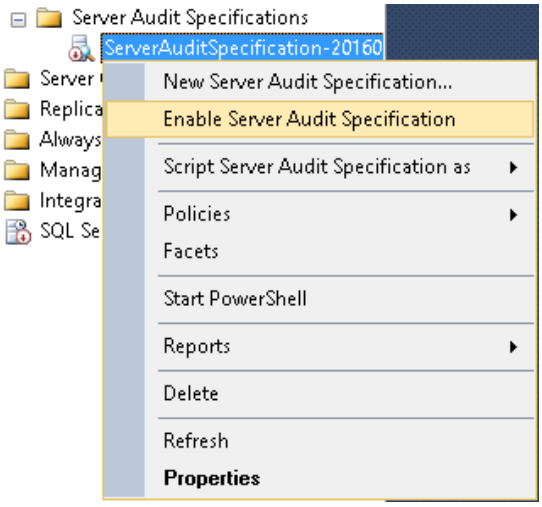

Figure 9

7. The above configuration generates event id "**33205**" for all configured audit specifications.

2.4 EventTracker agent configuration

2.4.1 Creating Event Filters

All the events generated by SQL through audit specifications are **information** events and are reported late. Thus, to aid the alerting of events in real-time event filters are to be configured. **Please note, log source matched in configurations below might change depending on the SQL instance name configured.**

1. Logon to EventTracker manager workstation.

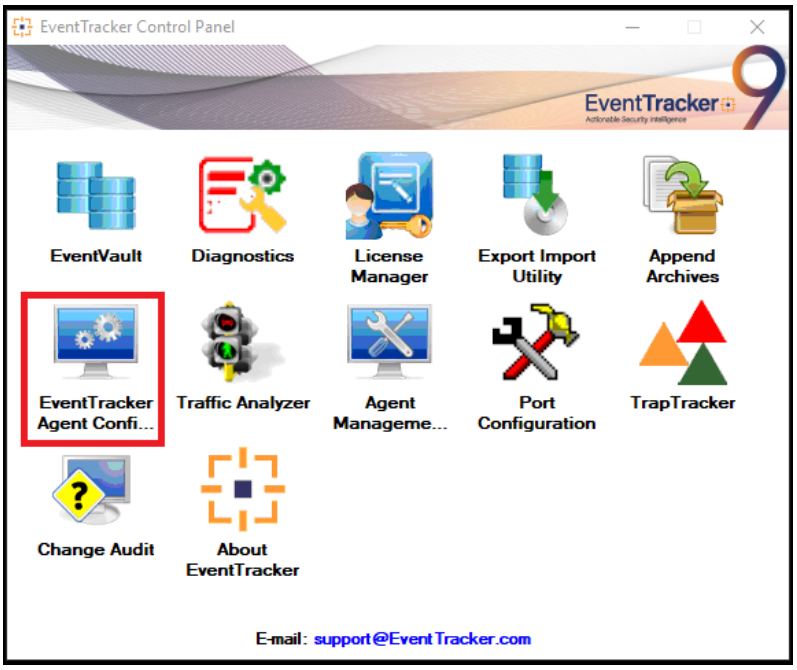

Figure 10

2. Open EventTracker control panel and click **EventTracker Agent Configuration**.

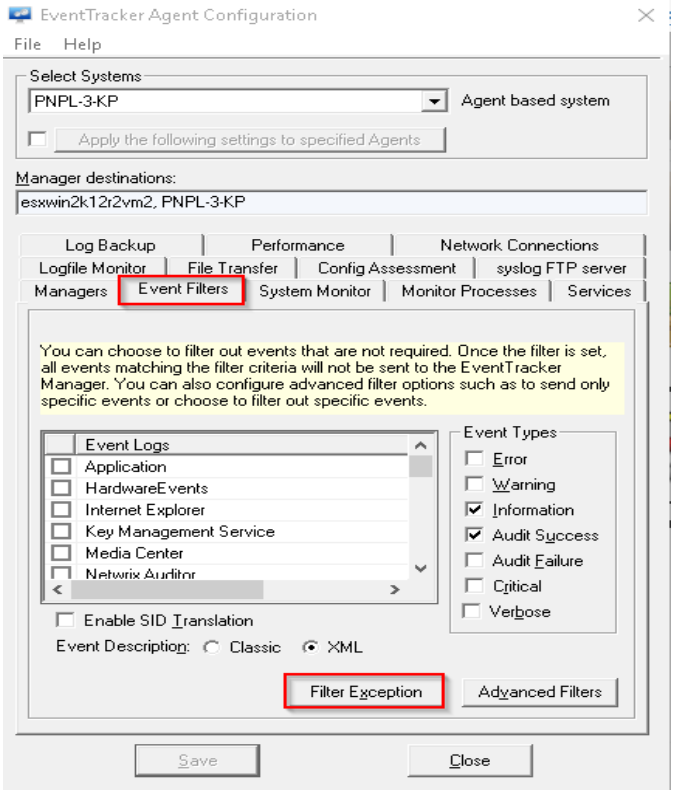

Figure 11

3. Select **Event Filters** tab and click **Filter Exception**.

   Filter exception window opens.

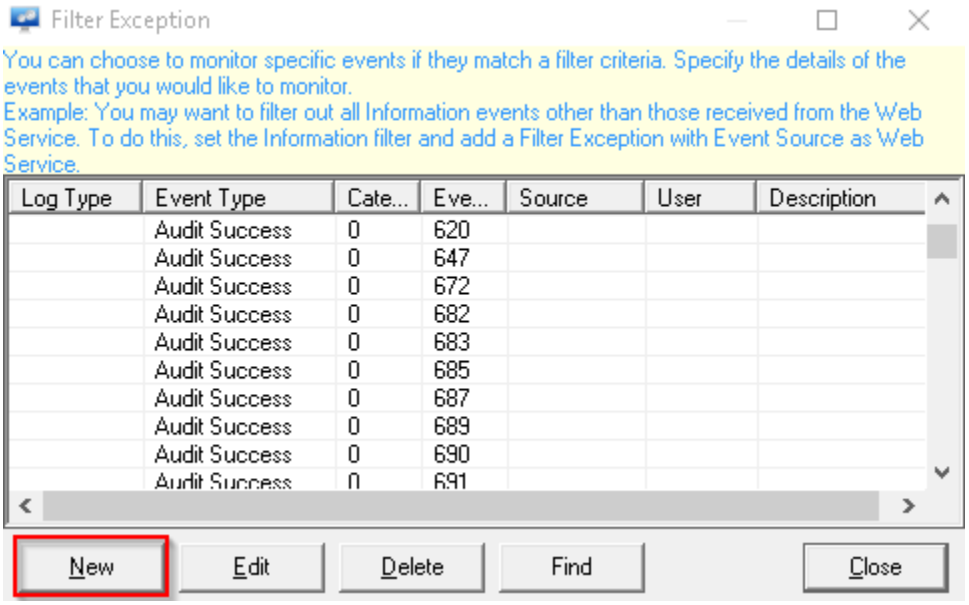

Figure 12

4. Click **New**, and configure as shown below:
5. Event filter properties for **audit events** are shown below:
6. For **Single Instance**, Match in Source should be named as corresponding instance name. For **multiple instances**, leave it blank.

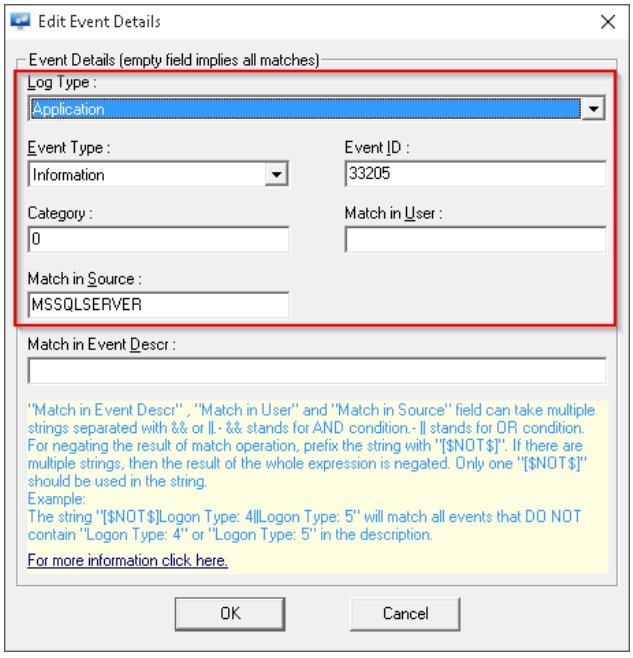

Figure 13

7. Click **OK** to apply.
8. If events are enabled for **login success and failure**, create filters with configurations as shown below:
9. Event filter properties for **login success** event are shown below:

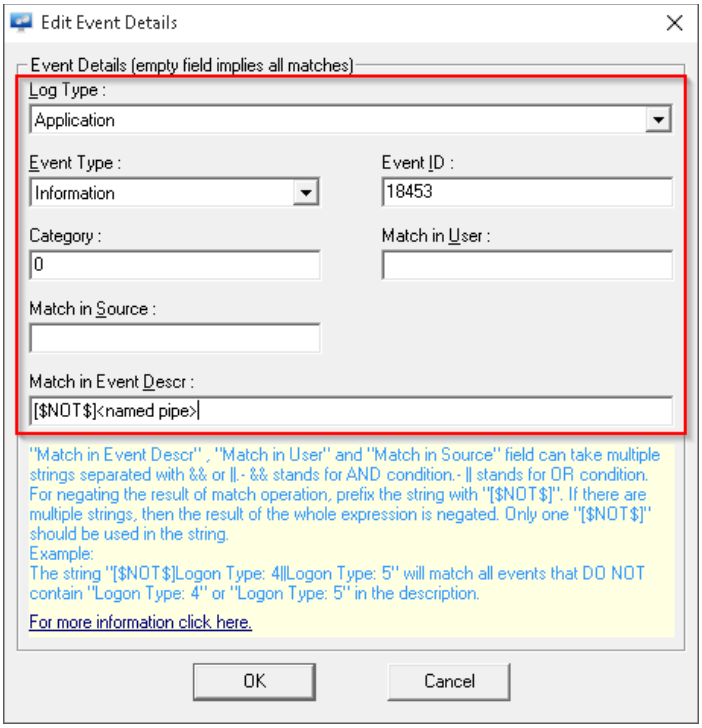

Figure 14

10. Event filter properties for **login failure** event are shown below:

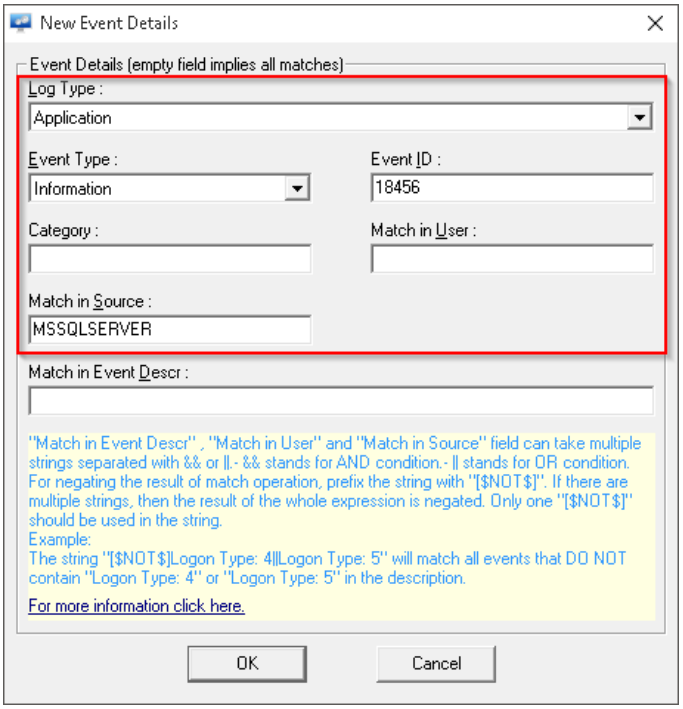

Figure 15

11. Click **SAVE** in agent configuration window to apply changes.

3. Extended events

Extended Events is a lightweight performance monitoring system that uses very few performance resources. It enables auditing for different actions, providing much granularity in the setup process and a wide coverage range of the SQL Server activity. **Below mentioned configuration must be applied to all the workstations, where SQL audit is required.**

3.1 Prerequisites

- **Microsoft SQL Server 2012 or later** must be installed.
- **Microsoft SQL Management Studio** for the respective version must be installed.
- **EventTracker agent 8 or later** must be installed on the SQL SERVER workstation.
- **PowerShell 5.1 or later** must be installed**.**
- **Administrative credentials** for script execution.

3.2 Creating extended event session

1. Contact **Support** for the **SQL extended events script pack** and download.
2. Extract downloaded zip file to the following path.

<EventTracker Installation Path>\MSSQL Server\

3. From extracted file location, double-click to open "**SQL Extended Events.sql**".

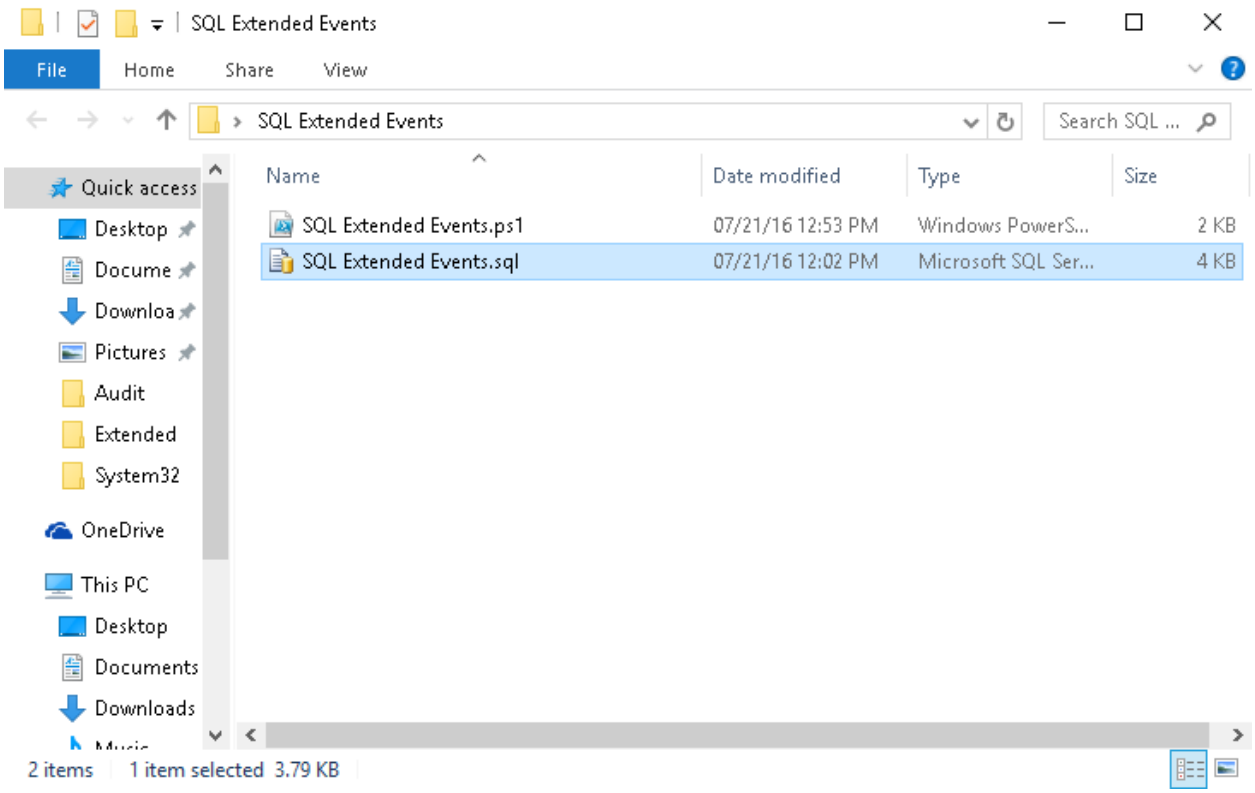

Figure 16

4. In **Microsoft SQL management studio**, login with appropriate credentials.
5. In the **SQL query**, change the highlighted path to the desired location of the "**.xel**" file. Click **Execute** to create and implement an extended event session.
   Preferred Location for ".xel" events:
   <EventTracker Installation Path>\MSSQL Server\
   Note: Create "MSSQL Server" folder if not existed in the EventTracker Installation Path.

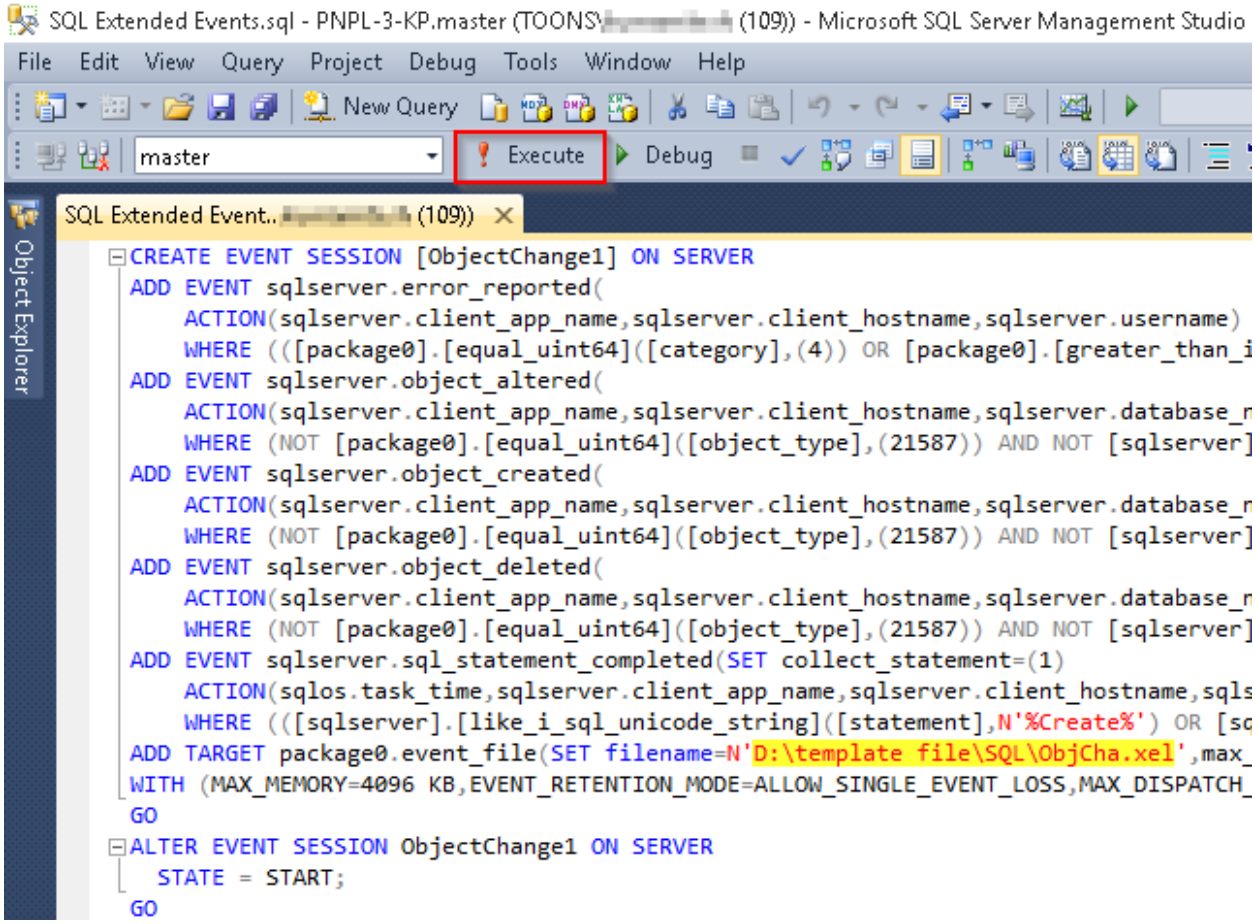

Figure 17

6. To view created session, navigate to **Object Explorer> Server Name> Management> Extended Events> Sessions> ObjectChange**.

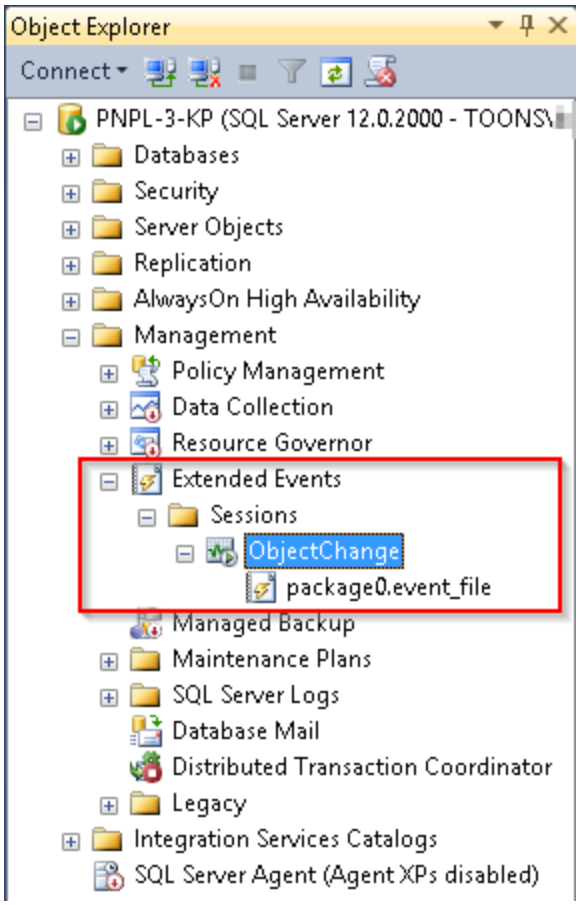

Figure 18

7. The above configuration will create "**. xel**" file with all relevant audit events at the earlier mentioned location.

3.3 Parse extended event session log file

Xel files are readable only through SQL Management Studio. Thus, PowerShell is deployed for file format conversion and custom parsing in the interest of EventTracker.

1. From earlier mentioned extracted file location, find "**EventTacker Task (SQL Extended Task).ps1**".

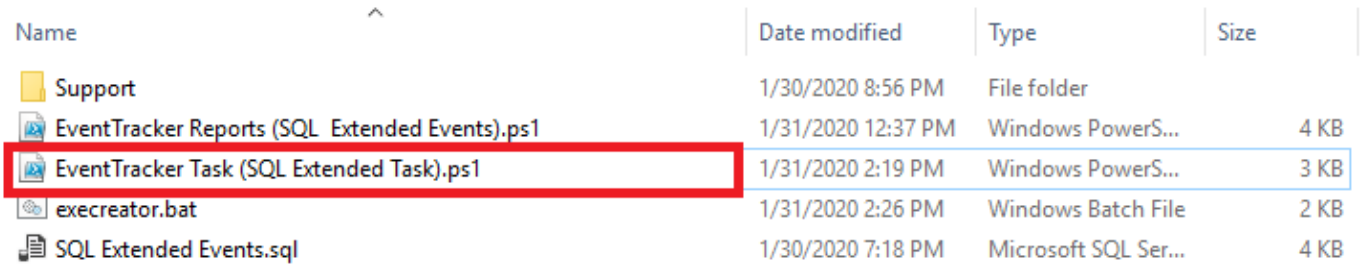

Figure 19

2. Logon to EventTracker Manager workstation with the administrative privileges.
3. Please run above the "**EventTacker Task (SQL Extended Task).ps1**" program it will start creating the task and logs will start sending to EventTracker.
4. Please verify Task is created for SQL Extended events in TaskSchedular.

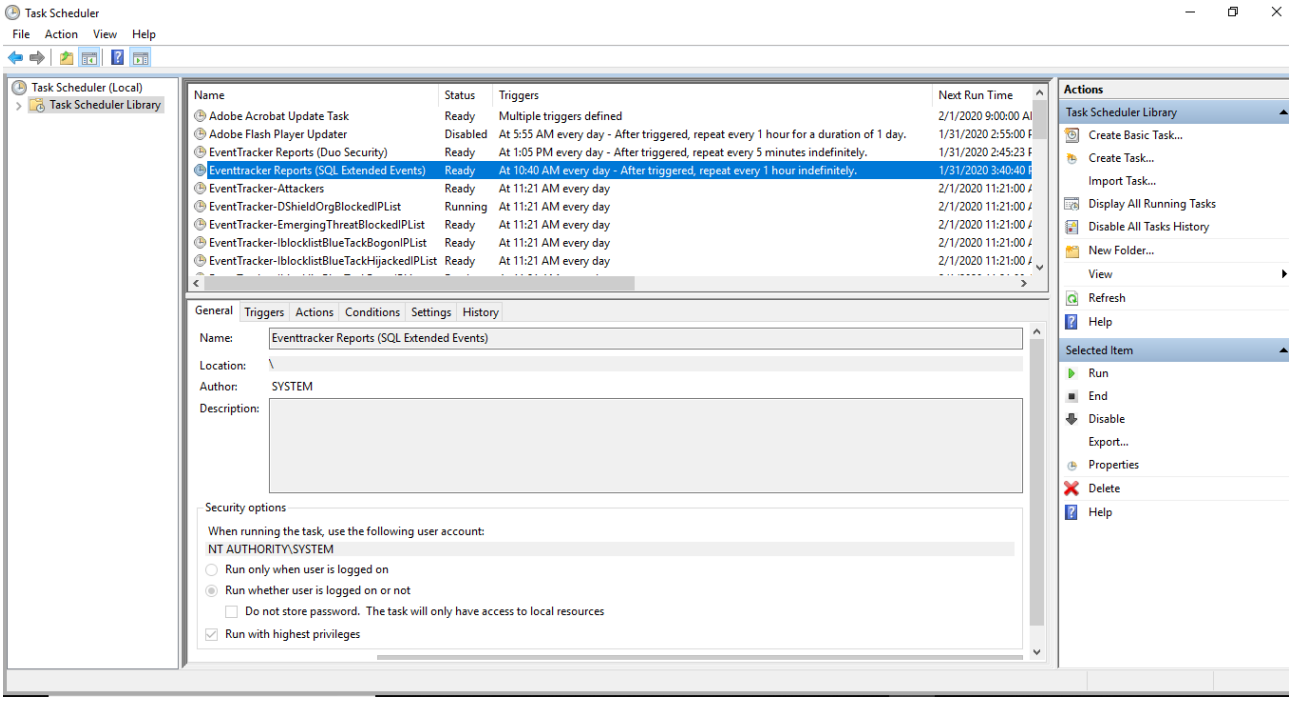

Figure 20