# Configure Auditing for Logins

Auditing using just the SQL Server Audit is not ideal as it isn’t very flexible, you can only send the events to a file or either the Windows Application or Security logs. You cannot create an event log of your and send the events to that. The functionality logs a huge amount of data which quickly swamps the event logs eg when testing it was logging between 10 and 12 events for every successful login. The event logs are not so easy to generate reports from and whilst we might want to keep audit data for few months we don’t necessarily want to keep all of the data in the Windows event logs for that long.

If you log the audit data to file you can then manage it by archiving and using file retention and the data can be imported into a database for reporting.

# Audit Artifacts

| Type | Object Name | Script Name | Function | Role |
| --- | --- | --- | --- | --- |
| Server Audit | TrackLogins | CreateTrackLoginsAudit.sql | Audit logins at the instance level to a file. This file defines where the data is audited to, event log or file. | Defines where the audit data is logged and what action to take if the audit fails. |
| Server Audit | AuditDataAudit | CreateAuditDataServerAudit.sql | Audits the Master database being used for storing imported audit data if any data tampering is detected. Audit events logged in the Windows Application Event log. | Defines where the audit data is logged and what action to take if the audit fails. |
| Server Audit | AuditSQLAgentJobs | CreateSQLAgentJobServerAudit.sql | Audits the MSDB database for Agent jobs being disabled or deleted. Audit events logged to the Windows Application Event log. | Defines where the audit data is logged and what action to take if the audit fails. |
| Server Audit Specification | TrackAllLogins | CreateTrackLoginsServerAuditSpecification.sql | Audits the following actions:  Audt\_Change\_Group – audits to Windows application event log if the audit is disabled/stopped.  Failed\_Login\_Group – logs failed logins.  Successful\_Login\_Group – logs successful logins. | Defines what to audit at the instance level. |
| Database Audit | DataChangeAudit | CreateMasterDatabaseAuditSpecification.sql | Audits for data tampering – has the audit set to use the AuditDataAudit server audit. Audits updates to the AuditData table. | Defines what to audit in a database. |
| Database Audit | AuditAgentJobStatus | CreateMSDBDatabaseAuditSpecification.sql | Audits the MSDB database for job tampering and incidents. Audits execution of the the following stored procs:  Sp\_update\_job  Sp\_agent\_delete\_job | Defines what to audit in a database. |
| SQL Agent Job | CheckAuditStatus | CreateSQLAgentJob-CheckAuditStatus.sql | Monitors the Server Audits and Server Audit Specifications and if any are disabled raises an error with message Audit is disabled. Runs every 30 seconds by default. | To generate an error if the either server audit specification or a server audit is disabled. |
| SQL Agent Job | AuditStopped | CreateSQLAgentJob-AuditStopped.sql | If the Login Audit is stopped an archive and collect will proceed. This is done using a powershell script called AuditRestartRoutine.ps1. The script executes two PowerShell scripts :-  AuditFileArchive.ps1  AuditImportArchiveData.ps1  These two scripts run in the above order archive any audit files and then import data from the archived files.  To prevent the disk space being used up by audit files you need to set a limit on how many files can be created. If the audit either needs to create a new file because the current one is full or the audit is stopped, the audit breaks because it cannot create a new file. Archiving the files into an archive folder ensures that the max files limit is never breached. | To make sure that no data is lost in the event of auditing being stopped and to stop auditing from being broken by reaching the max allowed audit file limit. |
| SQL Agent Job | ArchiveAuditFiles | CreateSQLAgentJob-ArchiveAuditFiles.sql | Executes the AuditFileArchive.ps1 script to archive all except the current audit file.  Executes a powershell script called AuditImportArchiveData.ps1 to ingest all data from archived audit files that have not already been ingested into the AuditData table in the database. | To ensure that the max audit files limit is not breached.  Prevents loss of audit data. |
| SQL Agent Job | AuditArchiveHousekeeping | CreateSQLAgentJob-AuditArchiveHousekeeping.sql | Executes the DeleteAuditFiles.ps1 script which deletes audit files in the archive which have gone past their retention. The retention is set by a parameter in the script. | To prevent a build up of old audit files in the archive which would eventually lead to the disk space being used up. |
| SQL Agent Job | AuditDataHousekeeping | CreateSQLAgentJob-AuditDataHousekeeping.sql | Executes the usp\_AuditDataHousekeeping stored proc on a schedule so that the AuditData table is regularly purged of old data. | To make sure that the data in the AuditData table doesn’t grow excessively. |
| Windows Task Scheduler Task | Event ID 17162 | Event ID 17162.xml | Executes the AuditFileArchive.ps1 script to archive files the SQL Server service starts up. | To ensure that the max file limit on the audit files is not breached. |
| PowerShell Script | AuditFileArchive.ps1 | AuditFileArchive.ps1 | Archives any audit files it finds in the path defined in the script | To ensure that the max file limit on the audit files is not breached. |
| PowerShell Script | AuditImportArchiveData.ps1 | AuditImportArchiveData.ps1 | Imports data from the archived audit files. | To ensure that if a file is archived before the ImportAuditData job has run, it will still be imported. |
| PowerShell Script | AuditRestartRoutine.ps1 | AuditRestartRoutine.ps1 | Executes the AuditFileArchive.ps1 and AuditImportArchiveData.ps1. This script is executed from the AuditStopped agent job so if the audit is ever stopped an archive and data import is carried out. | To ensure that the max file limit on the audit files is not breached and data is not lost. |
| PowerShell Script | DeleteAuditFiles.ps1 | DeleteAuditFiles.ps1 | Deletes audit files from the archive when they get past their retention. | To prevent a build of files causing the disk to run out of space. |
| PowerShell Script | Parameter.ps1 | Parameter.ps1 | Contains the parameter values for the PowerShell scripts | To centralize parameters to maintain consistency and help with maintenance. |
| SQL Alert | AuditStopped | CreateAuditStoppedAlert.sql | Alert raised when an error of severity 010 – Information with the message text Audit disabled. It then executes the AuditStopped job. | To archive files and import archive data so that the max file is not breached and data not lost. |
| SQL Credential | SQLAudit | CreateCredential-SQLAudit.sql | Credential for the PowerShell proxy. | To enable the agent jobs to run otherwise they can’t access the file system. |
| SQL PowerShell Proxy | AuditStatusProxy | CreateProxy-AuditStatusProxy.sql | Powershell proxy for agent jobs to run under. | To enable the agent jobs to run otherwise they can’t access the file system. |
| SQL Table | AuditData | CreateAuditDataTable.sql | Table to store the imported audit data. | Allows audit data to be persisted for later anlalysis/scrutiny. |
| SQL View | Uvw\_AuditData | CreateAuditDataView.sql | Provides a clean view of the data excluding unnecessary columns. | To make reporting easier. |
| SQL Stored Proc | Usp\_AuditDataHousekeeping | CreateAuditDataHousekeepingProc.sql | When executed will go through and purge out all data which is older than the value set by the @retention parameter. | To make sure that the data in the AuditData table doesn’t grow excessively. |

# Install Order and Configuration

Create Audit folder and then inside it create the following folders:

AuditFilesArchive

AuditFiles

Extract the AuditInstall zip file contents into the Audit folder.

Create the SQLAudit AD login and make sure it has local admin rights otherwise it won’t have permission to execute the PowerShell scripts.

Create a login on the SQL instance for SQLAudit and put in the sysadmin role.

Install the files in the following order and make config changes where stated:

|  |  |  |
| --- | --- | --- |
| Order | Script | Notes for install |
| 1 | CreateCredential-SQLAudit.sql | Need enter username and password for login |
| 2 | CreateProxy-AuditStatusProxy.sql | Check credential name is correct |
| 3 | CreateAuditDataTable.sql |  |
| 4 | CreateAuditDataView.sql |  |
| 5 | CreateTrackLoginsAudit.sql | Enter path for files, maxsize and max\_files |
| 6 | CreateAuditDataServerAudit.sql | Enter path for files, maxsize and max\_files |
| 7 | CreateSQLAgentJobServerAudit.sql |  |
| 8 | CreateTrackLoginsServerAuditSpecification.sql |  |
| 9 | CreateMasterDatabaseAuditSpecification.sql |  |
| 10 | CreateMSDBDatabaseAuditSpecification.sql |  |
| 11 | AuditFileArchive.ps1 | Edit the parameter file path in this script to point to PowerShell script path. |
| 12 | AuditImportArchiveData.ps1 | Edit the parameter file path in this script to point to PowerShell script path. |
| 13 | AuditRestartRoutine.ps1 | Edit the parameter file path in this script to point to PowerShell script path. |
| 14 | DeleteAuditFiles.ps1 | Edit the parameter file path in this script to point to PowerShell script path. |
| 15 | Parameter.ps1 | Edit parameter values. |
| 16 | CreateSQLAgentJob-CheckAuditStatus.sql | Check the Schedule and amend to suit. |
| 17 | CreateSQLAgentJob-AuditStopped.sql | Edit the @command path to point to yours and check the @proxy name is correct. |
| 18 | CreateSQLAgentJob-ArchiveAuditFiles.sql | Edit the @command path to point to yours and check the @proxy name is correct. Check the Schedule and amend to suit. |
| 19 | CreateSQLAgentJob-AuditArchiveHousekeeping.sql | Edit the @command path to point to yours and check the @proxy name is correct. Check the Schedule and amend to suit. |
| 20 | CreateAuditStoppedAlert.sql | Go to the alert response page and tick execute job then select the AuditStopped job from the drop down list. Set it to notify operators and add the DBA operator. |
| 21 | Event ID 17162.xml | Import the task into Windows Task Scheduler:  Opening the Task Scheduler  Right clicking on Task Scheduler  Select Import Task  Point to the xml file  In the General tab configure the account that the schedule will run under and make sure that you select Run whether the user is logged on or not.  In the Action tab edit the path in the Add Arguments to point to the PowerShell scripts folder.  In the Triggers tab check the SQL Instance is correct. |
| 22 | CreateAuditDataHousekeepingProc.sql |  |
| 23 | CreateSQLAgentJob-AuditDataHousekeeping.sql | Set schedule to run once a day. Edit the job step with the value you want retention period set to. |
| 24 | Setup and configure email on instance | Create operator and in the agent properties enable email and select failsafe operator |

# Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test | Expected Outcome | Checks Completed | Actual Outcome | Post Test Actions |
| Disable the AuditDataAudit | This should trigger the AuditStopped alert. The history page should show the number of occurance should have increased by one and the “date of last response” should be the same date time stamp of the test.  If database mail is enabled then an email should have been sent.  An archive and collect should have been executed. | In the properties of the AuditStopped alert check the history for the number of occurrences and the date of last repsonse before and after disabling the AuditDataAudit audit.  Check the sysmail\_allitems table in MSDB; it should show that an email was sent.  Check that there are no audit files in the AuditFiles folder.  Check that that the AuditArchiveImport job executed successfully. Run a query over the AuditData table to check that the latest audit data has been imported from the previous step. |  | Enable the AuditDataAudit audit. |
| Disable ArchiveAuditFiles job | Audit archive files are not moved into the AuditArchive folder. | Check that the AuditFiles folder accumulates more than one audit file and the files are not moved within the time set by the schedule of the ArchiveAuditFiles job (15 minutes in this instance). If there is only one file in the folder then move a file from the AuditArchive into the AuditFiles folder for the test. |  | Re-enable the ArchiveAuditFiles job |
| Copy some archive files to a safe place then set the retention in the parameters.ps1 file to 1 day. Execute AuditArchiveHousekeeping job | Files past retention should be deleted from the AuditArchive folder when this job is executed. | Check the AuditArchive files are not older than one day. |  | Set the retention in the parametes.ps1 file back to 7 days and then move the audit files that were deleted back into the AuditArchive folder. |
| Set the retention parameter value to 1 in AuditDataHousekeeping job then execute it. | All data older than one day should be deleted from the AuditData table. | Run a select statement over the AuditData table to check that the data is not older than one day. |  | Set the retention parameter in the AuditDataHousekeeping job back to 30 days. |
| Delete a row of data from the AuditData table to check DataChangeAudit audit works | Should record an audit record to show the data was removed. | Check that |  |  |
| Disable TrackAllLogins Server Audit Spec | Should trigger the AuditStopped alert and create an event in the Windows Application event log to show it was disabled. | Check Windows Application Event log for Event ID 17061, Error: 50000 Severity: 10 State: 1 Audit is disabled  Check the AuditStopped alert was triggerd by looking at the date of last response in the alert properties.  Check the sysmail\_allitems table in MSDB; it should show that an email was sent. |  |  |
| Disable either AuditDateAudit or AuditSQLAgentJobs server audit | Should trigger the AuditStopped alert and create an event in the Windows Application event log to show it was disabled. | Check Windows Application Event log for Event ID 17061, Error: 50000 Severity: 10 State: 1 Audit is disabled  Check the AuditStopped alert was triggerd by looking at the date of last response in the alert properties.  Check the sysmail\_allitems table in MSDB; it should show that an email was sent. |  |  |