Auditing für MySQL – Aber wie?

Ralf Gebhardt
SkySQL Ab
Finnland - Esbo

Schlüsselworte

Auditing, Zugriffsprotokoll, Audit-Plugin, MySQL, MariaDB

Einleitung

Auditing für MySQL - die Foren sind voll von Fragen, wie ein Auditing für MySQL durchgeführt werden kann. Und die Varianten, was unter Auditing verstanden wird, sind vielschichtig. Angefangen von der reinen Aufzeichnung der durchgeführten Abfragen über die Protokollierung der Benutzer und deren Zugriff auf Objekte bis hin zu Wünschen, die komplette Änderung der Daten selbst nachvollziehen zu können – all dies ist in Foren zu finden.

In diesem Vortrag wollen wir daher einige Standard-Funktionaltäten von MariaDB® und MySQL® Server – im weiteren unter „MySQL“ zusammengefasst – betrachten, die für die Protokollierung von Zugriffen auf MySQL und dessen Daten verwendet werden. Zudem werden wir uns am Beispiel des MariaDB Audit Plugins anschauen, welche Möglichkeiten einer Protokollierung durch die Nutzung der in MySQL implementierten Audit Plugin-API und eines Audit-Plugins heute zur Verfügung stehen. Methoden und Tools, welche im nächsten Schritt die Auswertung des erstellten Protokolls durchführen und gegebenenfalls auch Aktionen im Datenbank-System durchführen, werden nicht betrachtet.

Warum Auditing?

Bevor wir uns über das *Warum* Gedanken machen ist es notwendig festzulegen, wie wir für diesen Vortrag den Begriff *Auditing* definieren.

Der Begriff *Auditing* wird, wie auch die Begriffe *Cluster* und *Cloud* und viele mehr, in verschiedenster Form interpretiert. Daher soll hier kurz beschrieben werden, was wir für diesen Vortrag darunter verstehen, am besten durch die Betrachtung des Nutzens, den wir daraus ziehen - also, *warum* Auditing benötigt werden könnte.

Verschiedenste Sicherheitsstandards verlangen heute die Nachvollziehbarkeit des Zugriffs von Benutzern auf Daten, teilweise noch dediziert für den Zugriff auf kritische Daten wie Kreditkarten-Informationen, vermehrt jedoch wird die allgemeine Protokollierung gefordert. Genutzt wird dieses Protokoll zur Nachvollziehbarkeit, wer wann auf welche Daten Zugriff hatte oder zugreifen wollte. Diese Informationen werden teilweise in einem zweiten Schritt nach Auswertung genutzt, um präventiv tätig werden zu können.

Es reicht also nicht mehr aus nachzuweisen, dass der Zugriff auf Daten durch Zugriffs-Privilegien reglementiert ist.

Es wird daher ein Protokoll des Zugriffs auf Daten, im Wesentlichen auf die Objekte, welche diese Daten enthalten, benötigt. Dieses Protokoll muss enthalten, wer wann von wo auf welche Daten bzw. Datenobjekte zugegriffen hat.

Unterschieden wird hier auch gerne zwischen Zugriffen „technischer" Benutzer und direkten Zugriffen durch Personen. Unter technischen Benutzern versteht man Benutzer, die von Anwendungen verwendet werden und für die schon durch die Anwendung geregelt ist, auf welche Daten zugegriffen werden kann. Hier reicht oft die Protokollierung des Verbindungsaufbaus zum Server aus, also Connect und Disconnect.

Auditing – was sonst noch zu beachten ist

Zugriff auf Daten limitieren

Mit der Anforderung des Auditings selbst geht meist noch eine weitere Anforderung einher. Der Zugriff auf die Protokolle muss getrennt vom Rechte-System der Datenbank selbst geregelt werden können. Daher kann eine Lösung auf Basis von Tabellen im Datenbank-Server selbst meist nicht verwendet werden. Somit sind Log-Dateien oder die Nutzung von System-Logs wie Syslog meist eine Bedingung für das Auditing. Es muss auch beachtet werden, dass das Audit-Protokoll selbst die Abfragen und somit kritische Daten enthalten kann.

Steuerung des Auditing

Wenn Auditing zur Bedingung für den Betrieb eines Datenbank-Servers wird bedeutet dies, dass eine aktive Durchführung gewährleistet werden oder dessen Status zumindest überwacht werden muss, damit im Zweifelsfall geeignet reagiert werden kann, zum Beispiel durch das Herunterfahren des Servers.

Es sollte also nur einem sehr eingeschränkten Benutzerkreis möglich sein, ein Auditing zu de-aktivieren. Zudem sollte dies wiederum in einer Log-Datei protokolliert werden.

Der Status zum Auditing, aktiv oder nicht, sollte jederzeit abgefragt und in ein übergeordnetes Monitoring integrierbar sein.

Große Log-Dateien

Auditing einer Datenbank führt schnell zu großen Log-Dateien. Daher sollte die Verwaltung der Protokoll-Dateien über die Angabe von Name und Pfad hinausgehen. Anstatt der Verwendung von Log-Dateien können auch andere externe Systeme wie Syslog zur Protokollierung verwendet werden. Diese enthalten meist auch erweiterte Konfigurationsmöglichkeiten, welche weitere Optionen für eine garantierte Protokollierung mitbringen.

Das Auditing muss sichergestellt sein, auch wenn es zu Problemen im File-Handling der Log-Dateien kommt. Im Zweifelsfall muss das Auditing deaktiviert werden, dieser Status muss aber durch ein Monitoring-System erkennbar sein. Nur so kann geeignet – unter Umständen auch durch Abschalten des Servers - reagiert werden.

Auditing mit Hilfe von MySQL Log-Dateien

MySQL® Server und MariaDB® verwenden eine Reihe von Log-Dateien. Es soll daher hier betrachtet werden, in wie fern diese für Auditing genutzt werden könnten.

General Query Log

Zugriffe auf Daten oder Objekte (Tabellen,...) sowie der Verbindungsaufbau selbst können über Aktivierung des *General Query Log* ermittelt werden. Es ist jedoch ein komplexes Parsing der Datei nötig, um aus dem General Log ein für Auditing geeignetes Format zu erstellen. Es bestehen keine Möglichkeiten zu weiteren Konfigurationen für das General Query Log, auch das Handling der Log-Datei selbst erlaubt nur die Angabe eines Namens und Pfades.

Binary Log

Das Binary Log wird zur Replikation sowie Point In Time Recovery benötigt. Das Binary Log enthält allerdings nur Information betreffend der Abfragen, die zur Änderung von Daten führen. Lesende Abfagen (SELECTS) sind nicht enthalten. Das Binary Log ist daher kein geeignetes Auditing-Werkzeug.

Error Log

Der Fokus dieses Log-Files liegt, wie der Name schon sagt, auf der Aufzeichnung von Fehlern, Warnungen und Info-Meldungen zum Betrieb des Servers selbst. Für Auditing selbst kann höchstens interessant sein, dass hier fehlgeschlagene Verbindungsversuche protokolliert werden können.

Slow Query Log

Die Verwendung ist zur Aufspürung von Abfragen gedacht, welche länger als die Zeit x benötigen, um ein Ergebnis zu liefern. Connects und Disconnects sind nicht enthalten. Durch das Setzen von long_query_time auf 0 bekommt man alle Abfragen protokolliert, es gelten aber auch hier die unter „General Query Log“ beschriebenen Probleme.

Auditing mit Hilfe von Triggern?

Trigger können in MySQL pro Tabelle als BEFORE und AFTER Trigger definiert werden. Es existieren keine Trigger für SELECT und CONNECT. Ein Auditing könnte daher nur, wie auch mit dem Binary Log, für INSERT und UPDATE durchgeführt werden. Zudem wird diese Methode bei zunehmender Anzahl Tabellen sehr aufwendig. Das Auditing würde in MySQL Tabellen stattfinden müssen, Log-Dateien können nicht geschrieben werden.

MySQL und MariaDB Audit Plugins

Mit der Version 5.5.3 von MySQL wurde die Audit Plugin API vorgestellt. Diese kann verwendet werden, um eigene Audit Plugins zu schreiben. Die API liefert Informationen wie die Abfrage selbst, Benutzer und Art der Operation (QUERY, CONNECT, ..).

Auf Basis dieser API wurden verschiedene Audit Plugins implementiert. McAfee bietet ein unter Open Source gestelltes Plugin an, Oracle bietet seinen Kunden ein Closed-Source Produkt, das

MySQL Enterprise Audit Plugin. MariaDB stellt das MariaDB Audit Plugin bereit, wiederum ein Open Source Produkt.

Das MariaDB Audit Plugin

Das MariaDB Audit Plugin verwendet die Audit Plugin API und kann daher sowohl mit MariaDB als auch mit MySQL® Server verwendet werden. Mit MariaDB 5.5.31 und neueren Versionen wird vom Server außer den Abfragen selbst auch die Information zur Verfügung gestellt, welche Tabellen für eine Abfrage im lesenden oder schreibenden Zugriff verwendet werden. Somit kann auch bei Abfragen, welche Views oder Stored Procedures/Functions verwenden protokolliert werden, welche Tabellen dafür lesend und schreibend verwendet wurden. Dies ist für MySQL® Server nicht möglich.

Konfiguration und Status

Sowohl die Konfiguration-Variablen als auch die Status-Ermittlung erfolgt über Standard-Befehle von MySQL / MariaDB. Variablen können in der MySQL Konfigurationsdatei (my.cnf) sowie über den SET-Befehl gesetzt werden. Die Abfrage der aktuellen Werte erfolgt über

SHOW GLOBAL VARIABLES like 'server_audit%';

Außer den unter „Das Audit-Log“ angegebenen Konfigurationsmöglichkeiten stehen noch zur Verfügung:

- Definition, was im Auditing enthalten sein soll. Ausgewählt werden können CONNECT-, QUERY- und TABLE-Events
- Angabe von Benutzern, für die das Auditing durchgeführt werden oder die ignoriert werden sollen. Dies erlaubt es z.B., für „technische“ Benutzer reines CONNECT-Auditing durchzuführen.

Wichtig für das Monitoring der Audit-Funktion ist auch, dass der aktuelle Status ermittelt werden kann. Dies geschieht wiederrum, wie für MySQL Benutzer üblich, über

SHOW GLOBAL STATUS like 'server_audit%';

Als Status können angezeigt werden

- Auditing aktiv
- Aktuell genutzte Audit-Log-Datei
- Letzter gemeldeter Fehler

Das Audit-Log

Für die Protokollierung kann definiert werden, ob eigene Log-Dateien erstellt oder das Syslog verwendet werden soll. Die weitere Konfiguration bei Verwendung des Syslog wird über dessen Konfigurationsdatei durchgeführt. Für den Eintrag im Syslog können

- Ident
- Info
- Facility
- Priority

in der Plugin-Konfiguration angegeben werden.

Werden eigene Log-Dateien verwendet kann definiert werden:

- Pfad und Dateiname
- Maximale Größe einer Log-Datei
- Anzahl Log-Dateien bei Nutzung von Rotation

Kommt es zu einem Fehler beim Schreiben eines Log-Eintrags, so wird dies im Audit-Plugin-Status erkannt. Somit ist also auch ein Monitoring des Auditing möglich.

Das Auditing umfasst die folgenden Informationen

- Zeitstempel, wann die Abfrage erfolgt ist
- Server- Host
- Benutzername
- Client-Host
- Connection-ID
- Query-ID
- Operation (CREATE, READ, WRITE, ALTER, RENAME, DROP, QUERY, CONNECT, DISCONNECT, FAILED-CONNECT)
- Tabellen, auf die zugegriffen wurde
- Die Abfrage selbst
- Fehlercode für QUERY oder FAILED_CONNECT

Die Log-Datei wird im CSV Format gespeichert, was eine einfache Verwendung zur weiteren Analyse ermöglicht.

# # #

# # #

Kontaktadresse:

Ralf Gebhardt

SkySQL Ab

Tekniikantie 12

02150 Esbo

Finnland

Telefon:	+49 (0) 7457-930 646
Fax:	+49 (0) 7457-930 645
E-Mail	ralf.gebhardt@skysql.com
Internet:	www.skysql.com