CSIT115 Data Management and Security

CSIT882 Data Management Systems

Database Auditing

Subject Coordinators: Dr Chen Chen, Dr Thanh Le

School of Computing and Information Technology -
University of Wollongong

Database Auditing

Outline

- Concepts
- Auditing Categories
- Database Auditing in MySQL

Concepts

Database auditing involves observing a database so as to be aware of the actions of database users

Database administrators and consultants often set up auditing for security purposes, for example, to ensure that those without the permission to access information do not access it (Wikipedia: https://protect-au.mimecast.com/s/N9iFCE8wRRtnNoYNfpnGV5?domain=en.wikipedia.org)

Database Activity Monitoring (DAM) is a database security technology for monitoring and analyzing database activity that operates independently of the database management system (DBMS) and does not rely on any form of native (DBMS-resident) auditing or native logs such as trace or transaction logs.

In HTML view press 'p' to see the lecture notes

Concepts

Database Activity Monitoring is typically performed continuously and in real-time

An audit trail also called audit log is an outcome of Database Activity Monitoring

An Audit trail a security-relevant chronological sequence of records, that provide documentary evidence of the sequence of activities, that have affected at any time a specific operation, procedure or event

Database Auditing

Outline

- Concepts
- Auditing Categories
- Database Auditing in MySQL

Auditing Categories

Audit logon/logoff into a database

- Record two events: an event for sign-on and an event for sign-off
- Save login name, timestamps, TCP/IP address of client, and a program used to initiate connection
- Record failed login attempts

Audit sources of database usage

- It includes which network node a user is connected to (IP address and host name)
- Which application is being used to access a database

In HTML view press 'p' to see the lecture notes

Auditing Categories

Audit database activities outside normal operation hours

- Activities performed outside normal operating hours are usually suspect and maybe the result of unauthorized access to modify data
- Audit off-hours activities include logons/logoffs and all SQL activities
- Audit off-hours activities do not need to include the activities that are always scheduled to run off-hours. for example Extract, Transform, Load (ETL) data warehousing activities

Audit DDL activities

- DDL commands of SQL are potentially the most damaging commands that exists
- Auditing DDL activities is also done to eliminate errors of developers and database administrators
- Auditing database schemas changes can be done by: (1) using database audit features, (2) using external auditing system, (3) comparison of schema snapshots, (4) some system allow for using database triggers

In HTML view press 'p' to see the lecture notes

Auditing Categories

Audit database errors

- Auditing database errors is important because in many cases hackers use a technique of “trial-and error” to investigate a structure of a database, a well written and tested database application does not return errors
- Failed logins is a good example of errors that must be monitored
- Audit database errors may also lead to identification of weak points in database applications

Audit changes to sources of stored procedures and triggers

- Change to sources of already developed software may mean the attempts to incorporate malicious code
- Audit can be implemented by comparison of source code developed earlier with the present one, for example using `diff` program
- The second option is to use an external database security and auditing system, which alerts on any modify or create command used by a database user
- The third option is to use built in database features, for example ability of a system to trace “recompile” events to track changes to stored procedure and triggers

In HTML view press 'p' to see the lecture notes

Auditing Categories

Audit changes to privileges, user logins and other security attributes

- Audit deletion and addition of users, logins and role
- Audit changes to the mapping between logins and users/roles
- Audit privilege changes over user and roles
- Audit password changes
- Audit changes to security attributes at a server, database, statement, or object level

In HTML view press 'p' to see the lecture notes

Auditing Categories

Audit creation, changes and usage of database links and replication

- Simple implementations using daily differences
- Comparing snapshots
- Using database internal mechanisms

Audit changes to sensitive data (data change audit trails to enforce accuracy in financial data)

- Fully recording “old” and “new” values for each DML activity
- Due to the large number of DML operations the audits must be done very selectively
- Carefully chose data objects to be audited
- Use database system capabilities, external audit systems, or database triggers

In HTML view press 'p' to see the lecture notes

Auditing Categories

Audit `SELECT` statements for privacy sets

- Record where `SELECT` statements come from
- Who (username) retrieved the data
- What data was actually retrieved
- Identify so called privacy sets, for example which associations of data are really important
- Implementation uses database traces and external auditing systems

Audit any changes made to the definitions of what to audit

- Audit changes made to the definitions of audit trails and changes to the audit trails themselves
- Use built-in database features and external auditing systems

Database Auditing

Outline

Concepts

Auditing Categories

Database Auditing in MySQL

Database Auditing in MySQL

MySQL Enterprise Audit (plugin not available within Community Edition)

- Dynamically enable/disable audit stream
- Implement policies that log all or selected login or query activities
- Automatically rotate audit log files based on size
- Integrate XML-based audit log stream with MySQL, Oracle and other third party solutions

Using stored procedures

Using database triggers

External audit plugin libraries

Analysis of database logs

In HTML view press 'p' to see the lecture notes

Database Auditing in MySQL

Using database logs

- MySQL Server has several logs that contain information about the user activities
- Error log: problems encountered starting, running, or stopping MySQL server
- General query log: Established client connections and statements received from clients
- Binary log: Statements that change data (also used for replication)
- Relay log: Data changes received from a replication master server
- Slow query log: Queries that took more than a given period of time to execute
- DDL log (metadata log): Metadata operations performed by DDL statements

In HTML view press 'p' to see the lecture notes

Database Auditing in MySQL

Error log

- Error log contains information indicating when the database server was started and stopped and also any critical errors that occurred while the server was running
- A system variable `log_error` determines a location a file `error.log` with information included in error log

Listing a value of system initialization variable 'log_error'

```
show variables like 'log_error';
```

A value of system initialization variable 'log_error'

```
+---------------+--------------------------+
| Variable_name | Value                    |
+---------------+--------------------------+
| log_error     | /var/log/mysql/error.log |
+---------------+--------------------------+
```

In HTML view press 'p' to see the lecture notes

Database Auditing in MySQL

Error log

- It is possible to list the contents of a file `error.log` in the following way

Listing the contents of Error log

```
cat /var/log/mysql/error.log | more
```

Sample contents of error log

```
151110 14:44:15 mysqld_safe Starting mysqld daemon with
                        databases from /var/lib/mysql
2015-11-10T03:44:15.916141Z 0 [Warning] Changed limits: max_open_files: 1024
(requested 5000)
2015-11-10T03:44:15.916275Z 0 [Warning] Changed limits: table_open_cache: 431
(requested 2000)
2015-11-10T03:44:16.831947Z 0 [Warning] 'NO_ZERO_DATE', 'NO_ZERO_IN_DATE' and
 'ERROR_FOR_DIVISION_BY_ZERO' sql modes should be used with strict mode.
They will be merged with strict mode in a future release.
2015-11-10T03:44:16.832003Z 0 [Warning] 'NO_AUTO_CREATE_USER' sql mode was not set.
2015-11-10T03:44:16.837216Z 0 [Note] /usr/sbin/mysqld (mysqld 5.7.9) starting as
process 22405 .
…             …             …              …             …
```

In HTML view press 'p' to see the lecture notes

Database Auditing in MySQL

General query log

- A system variable `general_log` controls logging to a general query log

Listing a value of system initialization variable 'general_log'

```
show variable like 'general_log';
```

A value of system initialization variable 'general_log'

```
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| general_log   | OFF   |
+---------------+-------+
```

- To start logging execute the following statement

Setting a value of system initialization variable 'general_log'

```
set global general_log='ON';
```

A value of system initialization variable 'general_log'

```
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| general_log   | ON    |
+---------------+-------+
```

In HTML view press 'p' to see the lecture notes

Database Auditing in MySQL

General query log

- General query log is written either to a file or to a relational table
- A variable `log_output` determines whether general query log is written either to a file or to a relational table

```
show variables like 'log_output';
```

Listing a value of system initialization variable 'log_output'

A value of system initialization variable log_output'

```
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| log_output    | FILE  |
+---------------+-------+
```

- A name of file for a general query log is determined by a variable `general_log_file`

```
show variables like 'general_log_file';
```

Listing a value of system initialization variable 'general_log_file

A value of system initialization variable general_log_file'

```
+------------------+---------------------------------------+
| Variable_name    | Value                                 |
+------------------+---------------------------------------+
| general_log_file | /var/lib/mysql/csit115-VirtualBox.log |
+------------------+---------------------------------------+
```

In HTML view press 'p' to see the lecture notes

Database Auditing in MySQL

General query log

- To redirect a general query log to a relational table we change a value of variable `log_output` in the following way:

```
set global log_output='TABLE';                    Setting a value of system initialization variable 'log_output'
```

```
+---------------+-------+                                  A value of system initialization variable log_output'
| Variable_name | Value |
+---------------+-------+
| log_output    | TABLE |
+---------------+-------+
```

- General query query log is recorded in a relational table `mysql.general_log`

```
DESCRIBE mysql.general_log;                                       Listing a structure of 'general_log' table
```

```
+--------------+---------------------+------+-----+----------------------+------------------------------+     'general_log' table
| Field        | Type                | Null | Key | Default              | Extra                        |
+--------------+---------------------+------+-----+----------------------+------------------------------+
| event_time   | timestamp(6)        | NO   |     | CURRENT_TIMESTAMP(6) | on update CURRENT_TIMESTAMP(6) |
| user_host    | mediumtext          | NO   |     | NULL                 |                              |
| thread_id    | bigint(21) unsigned | NO   |     | NULL                 |                              |
| server_id    | int(10) unsigned    | NO   |     | NULL                 |                              |
| command_type | varchar(64)         | NO   |     | NULL                 |                              |
| argument     | mediumblob          | NO   |     | NULL                 |                              |
+--------------+---------------------+------+-----+----------------------+------------------------------+
```

Database Auditing in MySQL

General query log

To remove old contents of a general query log execute the following `TRUNCATE TABLE` statement

TRUNCATE TABLE statement

```
TRUNCATE TABLE mysql.general_log;
```

`TRUNCATE TABLE` statement is surprisingly categorized as DDL statement and it removes the contents of a relational table forever

It means that `TRUNCATE TABLE` statement cannot be reversed with `ROLLBACK` statement

In HTML view press 'p' to see the lecture notes

Database Auditing in MySQL

To find the contents of a general query log we use the following `SELECT` statement

Listing the contents of 'general_log' table

```
SELECT * FROM mysql.general_log;
```

The sample contents of 'general_log' table

```
+----------------------------+---------------------------+-----------+-----------+...
| event_time                 | user_host                 | thread_id | server_id |
+----------------------------+---------------------------+-----------+-----------+...
| 2017-05-08 14:14:45.546765 | root[root] @ localhost [] |         3 |         0 |
+----------------------------+---------------------------+-----------+-----------+...


                                                   ...--------------+---------------------------------+
                                                   ... command_type |  argument                       |
                                                   ...--------------+---------------------------------+
                                                   ... Query        | select * from mysql.general_log |
                                                   ...--------------+---------------------------------+
```

Database Auditing in MySQL

General query log

- Note, that when MySQL server is shutdown and restarted all values of system variables return to its default values or values set up in the system configuration file
- It means that after shutdown and restart the values of variables like `general_log, log_output, general_log_file` return to their original values
- To stop writing into a general query log execute a statement

Setting a value of system initialization variable 'general_log'

```
set global general_log='OFF'
```

Listing a value of system initialization variable 'general_log'

```
show variables like 'general_log';
```

A value of system initialization variable 'general_log'

```
+-------------+-------+
| Variable_name | Value |
+-------------+-------+
| general_log   | OFF   |
+-------------+-------+
```

In HTML view press 'p' to see the lecture notes

Database Auditing in MySQL

Slow query log

- The slow query log consists of SQL statements that took more than `long_query_time` seconds to execute and required at least `min_examined_row_limit` rows to be examined
- The minimum and default values of `long_query_time` are 0 and 10, respectively

Listing a value of system initialization variable 'long_query_time'

```
show variables like long_query_time;
```

A value of system initialization variable 'long_query_time'

```
+-----------------+-----------+
| Variable_name   | Value     |
+-----------------+-----------+
| long_query_time | 10.000000 |
+-----------------+-----------+
```

Database Auditing in MySQL

Slow query log

Listing a value of system initialization variable 'min_examined_row_limit'

```
show varfiables like min_examined_row_limit;
```

A value of system initialization variable 'min_examined_row_limit'

```
+------------------------+-------+
| Variable_name          | Value |
+------------------------+-------+
| min_examined_row_limit | 0     |
+------------------------+-------+
```

The value can be specified to a resolution of microseconds

For logging to a file, times are written including the microseconds part

For logging to tables, only integer times are written; the microseconds part is ignored

Database Auditing in MySQL

Slow query log

- To change a value of long query time use the following statement

Listing and setting a value of system initialization variable 'long_query_time'

```
SET long_query_time = 0
SHOW VARIABLES LIKE 'long_query_time';
```

A value of system initialization variable 'long_query_time'

```
+-----------------+----------+
| Variable_name   | Value    |
+-----------------+----------+
| long_query_time | 0.000000 |
+-----------------+----------+
```

In HTML view press 'p' to see the lecture notes

Database Auditing in MySQL

Slow query log

- A system variable `slow_query_log` controls logging to a slow query log

Listing a value of system initialization variable 'slow_query_log'

```
show variables like 'slow_query_log';
```

A value of system initialization variable 'slow_query_log'

```
+----------------+-------+
| Variable_name  | Value |
+----------------+-------+
| slow_query_log | OFF   |
+----------------+-------+
```

Database Auditing in MySQL

Slow query log

- To start logging execute the following statement

Setting a value of system initialization variable 'slow_query_log'

```
set global slow_query_log='ON'
```

Listing a value of system initialization variable 'slow_query_log'

```
show variables like 'slow_query_log';
```

A value of system initialization variable 'slow_query_log'

```
+----------------+-------+
| Variable_name  | Value |
+----------------+-------+
| slow_query_log | ON    |
+----------------+-------+
```

Database Auditing in MySQL

Slow query log

- Slow query log is written either to a file or to a relational table
- A variable `log_output` determines whether general a query log is written either to a file or to a relational table

Listing a value of system initialization variable 'log_output'

```
show variables like 'log_output'
```

A value of system initialization variable 'log_output'

```
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| log_output    | FILE  |
+---------------+-------+
```

Database Auditing in MySQL

Slow query log

- A name of file for a slow_query_log is determined by a variable `slow_query_log_file`

Listing a value of system initialization variable 'slow_query_log_file'

```
show variables like 'slow_query_log_file'
```

A value of system initialization variable 'slow_query_log_file'

```
+---------------------+------------------------------------------+
| Variable_name       | Value                                    |
+---------------------+------------------------------------------+
| slow_query_log_file | /var/lib/mysql/csit115-VirtualBox-slow.log |
+---------------------+------------------------------------------+
```

Database Auditing in MySQL

Slow query log

- To redirect slow query log to a relational table we change a value of the variable `log_output` in the following way:

Listing a value of system initialization variable 'log_output'

```
set global log_output='TABLE'
```

A value of system initialization variable 'log_output

```
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| log_output    | TABLE |
+---------------+-------+
```

In HTML view press 'p' to see the lecture notes

Database Auditing in MySQL

Slow query log

- A slow query log is recorded in a relational table `mysql.slow_log`

Listing a structure of 'slow_log' table

```
DESCRIBE mysql.slow_log
```

A structure of 'slow_log' table

```
+----------------+---------------------+------+-----+----------------------+------------------------------+
| Field          | Type                | Null | Key | Default              | Extra                        |
+----------------+---------------------+------+-----+----------------------+------------------------------+
| start_time     | timestamp(6)        | NO   |     | CURRENT_TIMESTAMP(6) | on update CURRENT_TIMESTAMP(6) |
| user_host      | mediumtext          | NO   |     | NULL                 |                              |
| query_time     | time(6)             | NO   |     | NULL                 |                              |
| lock_time      | time(6)             | NO   |     | NULL                 |                              |
| rows_sent      | int(11)             | NO   |     | NULL                 |                              |
| rows_examined  | int(11)             | NO   |     | NULL                 |                              |
| db             | varchar(512)        | NO   |     | NULL                 |                              |
| last_insert_id | int(11)             | NO   |     | NULL                 |                              |
| insert_id      | int(11)             | NO   |     | NULL                 |                              |
| server_id      | int(10) unsigned    | NO   |     | NULL                 |                              |
| sql_text       | mediumblob          | NO   |     | NULL                 |                              |
| thread_id      | bigint(21) unsigned | NO   |     | NULL                 |                              |
+----------------+---------------------+------+-----+----------------------+------------------------------+
```

Database Auditing in MySQL

Slow query log

- To remove the old contents of a slow query log we use the following `TRUNCATE TABLE` statement

TRUNCATE TABLE statement

```
TRUNCATE TABLE mysql.slow_log;
```

- `TRUNCATE TABLE` statement is surprisingly categorized as a DDL statement and it removes the contents of a relational table forever
- It means that `TRUNCATE TABLE` statement cannot be reversed with `ROLLBACK` statement
- To find the contents of some columns from a slow log execute `SELECT` statement

Listing the contents of 'slow_log' table

```
SELECT query_time, sql_text FROM mysql.slow_log
```

The sample contents of 'slow_log' table

```
+-----------------+-------------------------------------------------+
| query_time      | sql_text                                        |
+-----------------+-------------------------------------------------+
| 00:00:00.034088 | truncate table mysql.slow_log                   |
| 00:00:00.000870 | select * from mysql.slow_log                    |
| 00:00:00.000914 | select query_time, sql_text from mysql.slow_log |
+-----------------+-------------------------------------------------+
```

Database Auditing in MySQL

Slow query log

- Note, that when MySQL server is shutdown and restarted all values of the system variables return to their default values or values set up in the system configuration file
- It means that after shutdown and restart the values of variables like `long_query_time`, `min_examined_row_limit`, `slow_query_log`, `log_output`, `slow_query_log_file` return to their original values
- To stop writing into slow query log execute a statement

Setting a value of system initialization variable 'slow_query_log'

```
set global slow_query_log='OFF'
```

Listing a value of system initialization variable 'slow_query_log'

```
show variables like 'slow_query_log';
```

A value of system initialization variable 'slow_query_log'

```
+----------------+-------+
| Variable_name  | Value |
+----------------+-------+
| slow_query_log | OFF   |
+----------------+-------+
```

Database Auditing in MySQL

Binary log

- A binary log contains “events” that describe database changes such as table creation operations or changes to table data
- It also contains events for statements that potentially could have made changes (for example, a `DELETE` which matched no rows)
- Binary log also contains information about how long took the computations of `UPDATE` statements
- Binary log has two important purposes:
  - it is used for replication; the binary log on a master replication server provides a record of the data changes to be sent to slave servers
  - it is used for data recovery operations
- Binary log is not used for statements such as `SELECT` or `SHOW` that do not modify data
- To log all statements (for example, to identify a problem query), we use a general query log

In HTML view press 'p' to see the lecture notes

Database Auditing in MySQL

DDL log

- A DDL log, or metadata log, records metadata operations generated by data definition statements such as `CREATE TABLE`,`DROP TABLE`, and `ALTER TABLE`
- MySQL uses this log to recover from crashes occurring in the middle of a metadata operation
- When executing the statement `DROP TABLE t1, t2`, we need to ensure that both `t1` and `t2` are dropped, and that each table drop is complete

In HTML view press 'p' to see the lecture notes

References

C. Coronel, S. Morris, A. Basta, M. Zgola, Data Management and Security, Chapter 12, Cengage Compose eBook, 2018, eBook: Data Management and Security, 1st Edition

MySQL 5.7 Reference Manual, MySQL Server Administration, MySQL Server logs