بسمه تعالی

رویدادنگاری، ممیزی و جرم‌شناسی در پایگاه داده‌ی MySQL

فهرست مطالب

۱ مقدمه

در بحث جرم‌شناسی پایگاه‌های داده، جمع‌آوری شواهد و اطلاعات برای تجزیه‌وتحلیل از اهمیت زیادی برخوردار است. شواهد در پایگاه داده شامل رخدادهای ثبت‌شده و ممیزی‌ها است. ازاین‌رو، ثبت رویدادها و تهیه‌ی ممیزی در تمامی پایگاه‌های داده موردتوجه قرار گرفته است.

با توجه به اهمیت ثبت رویدادها و تهیه‌ی ممیزی در پایگاه‌های داده، در این گزارش پایگاه داده‌ی MySQL از این دو جهت موردبررسی قرار گرفته است.

در بخش ۳ انواع فایل‌های رویدادنگاری[1] موجود در پایگاه داده‌ی MySQL موردبررسی قرارگرفته‌اند. ممیزی در پایگاه داده، شامل مشاهده و نظارت بر پایگاه داده به منظور آگاهی از اقدامات کاربران پایگاه داده است. در بخش ۴، افزونه‌ی audit_log به منظور تهیه‌ی ممیزی موردبررسی قرار گرفته است. برای پایگاه داده‌ی MySQL ابزار جرم‌شناسی اختصاصی مفیدی، تابه‌حال یافت نشده است. در بخش ۵، ابزار DB Audit and security 360 که از پایگاه‌های داده مختلف ازجمله MySQL پشتیبانی می‌کند و برای جرم‌شناسی می‌تواند مفید باشد، توضیح داده می‌شود.

۲ نحوه‌ی ثبت وقایع در پایگاه داده MySQL

در این بخش انواع فایل‌های رویدادنگاری موجود در پایگاه داده‌ی MySQL موردبررسی قرارگرفته‌اند. لازم به ذکر است که بررسی‌های انجام‌شده در این گزارش بر روی سیستم عامل ویندوز ۷ و سیستم مدیریت پایگاه داده‌ی MySQL 5.7 (نسخه‌ی Community) صورت گرفته است.

۲-۱ انواع فایل‌های رویدادنگاری

کارگزار MySQL دارای فایل‌های رویدادنگاری گوناگونی است که با کمک آن‌ها می‌توان وقوع فعالیت‌های مختلف را پیگیری کرد. فایل‌های رویدادنگاری به‌صورت پیش‌فرض در مسیر خروجی دستور SHOW VARIABLES LIKE '%datadir%'; نوشته می‌شوند (شکل ۱).

[1] Log files

```
mysql> show variables like '%datadir%';
+---------------+-----------------------------------------+
| Variable_name | Value                                   |
+---------------+-----------------------------------------+
| datadir       | C:\ProgramData\MySQL\MySQL Server 5.7\Data\ |
+---------------+-----------------------------------------+
1 row in set (0.02 sec)
```

شکل ۱ مسیر دایرکتوری Data

در جدول زیر انواع فایل‌های رویدادنگاری موجود در پایگاه داده‌ی MySQL، به‌اختصار توضیح داده‌شده‌اند [۱].

جدول ۱ انواع فایل‌های رویدادنگاری

نوع فایل رویدادنگاری	اطلاعات ثبت‌شده در فایل
Error log	مشکلات رخ داده شده در حین راه‌اندازی، توقف یا در طول اجرای کارگزار MySQL
General Query log	اتصالات ایجادشده‌ی سرویس‌گیرنده[2] و عبارات[3] دریافت شده از سرویس‌گیرندگان
Binary log	عباراتی که باعث تغییر داده‌ها می‌شوند.
Relay log	تغییرات داده که از کارگزار ارشد تکرار[4] دریافت می‌شوند در relay log ثبت می‌گردند. این نوع فایل‌های رویدادنگاری تنها بر روی کارگزاران slave هستند و تغییرات داده‌ها از جانب کارگزار ارشد را در خود نگه می‌دارند.
Slow query log	پرسمان‌هایی که زمان اجرای آن‌ها بیش از زمان تعیین‌شده توسط پارامتر long_query_time به‌طول می‌انجامد.
DDL log (metadata log)	عملیات اجراشده توسط عبارات DDL

[2] Established client connections

[3] Statements

[4] Replication master server

در ادامه هر یک از این فایل‌ها به‌اختصار توضیح داده‌شده‌اند.

۱-۱-۲ Error log

فایل رویدادنگار خطا[5] شامل رکوردهای مربوط به زمان‌های راه‌اندازی و توقف کارگزار MySQL است. همچنین در این فایل، پیغام‌های خطا، هشدار و یادداشت‌هایی[6] که در حین راه‌اندازی و توقف کارگزار و در طول اجرای آن ممکن است رخ دهند نیز مشاهده می‌شود. به‌عنوان‌مثال، درصورتی‌که کارگزار MySQL تشخیص دهد که جدولی نیاز به بررسی خودکار یا اصلاح[7] دارد، یک پیغام در فایل رویدادنگار خطا می‌نویسد[۱].

پیغام‌های مربوط به ثبت خطا یا در کنسول نمایش داده می‌شوند یا در فایلی با نام [host_name].err نوشته می‌شوند. همچنین می‌توان محل و نام فایل دلخواهی را برای ذخیره‌سازی پیغام‌ها مشخص کرد.

دستور زیر نام فایل فعلی برای ذخیره‌سازی پیغام‌های خطا را نمایش می‌دهد (شکل ۲):

```
SHOW VARIABLES LIKE '%log_error%';
```

```
mysql> show variables like '%log_error%';
+----------------------+------------------+
| Variable_name        | Value            |
+----------------------+------------------+
| binlog_error_action  | ABORT_SERVER     |
| log_error            | .\SEPIDEH-PC.err |
| log_error_verbosity  | 3                |
+----------------------+------------------+
3 rows in set (0.88 sec)
```

شکل ۲ مسیر فایل رویدادنگار خطا

همان‌طور که ذکر شد، می‌توان پیغام‌های مربوط به ثبت خطا را در کنسول نمایش داد؛ تغییر پیکربندی برای نمایش پیغام‌های خطا در کنسول، به‌صورت زیر انجام می‌شود:

- ابتدا سرویس MySQL متوقف می‌شود.
- سپس فایل my.ini از محل زیر به‌منظور ویرایش باز می‌شود.

---

[5] Error log

[6] Note

[7] Repair

C:\ProgramData\MySQL\MySQL Server 5.7

- مطابق شکل ۲، در زیر بخش Error Logging، پارامتر console جایگزین پارامتر log-error می‌شود تا پیغام‌ها در کنسول نمایش داده شوند.

```
# Error Logging.
console
#log-error="SEPIDEH-PC.err"

# Server Id.
server-id=1

# Secure File Priv.
secure-file-priv="C:/ProgramData/MySQL/MySQL Server 5.7/Uploads"

# The maximum amount of concurrent sessions the MySQL server will
```

شکل ۲ تنظیمات ثبت خطا در فایل my.ini

- حال کارگزار MySQL راه‌اندازی می‌شود.

به منظور مشاهده‌ی نمونه‌ای از فایل‌های رویدادنگار خطا، تنظیمات به حالت پیش‌فرض برگردانده می‌شوند. در ادامه، سعی به تولید پیغام خطا می‌شود. بدین منظور، ابتدا کارگزار MySQL را متوقف و در فایل my.ini پارامتر ناشناخته‌ی error-log=0، وارد می‌شود. پس‌ازآن، کارگزار راه‌اندازی می‌شود. شکل ۳ پیغام خطای ذخیره‌شده در فایل host_name.err را نمایش می‌دهد.

```
2017-08-29T17:22:20.0520952 0 [Note] InnoDB: File '.\ibtmp1' size is now 12 MB.
2017-08-29T17:22:20.0833782 0 [Note] InnoDB: 96 redo rollback segment(s) found. 96 redo rollback segment(s)
2017-08-29T17:22:20.0833782 0 [Note] InnoDB: 32 non-redo rollback segment(s) are active.
2017-08-29T17:22:20.0833782 0 [Note] InnoDB: Waiting for purge to start
2017-08-29T17:22:20.1458822 0 [Note] InnoDB: 5.7.19 started; log sequence number 2540964
2017-08-29T17:22:20.1458822 0 [Note] InnoDB: Loading buffer pool(s) from C:\ProgramData\MySQL\MySQL Server
2017-08-29T17:22:20.1458822 0 [Note] Plugin 'FEDERATED' is disabled.
2017-08-29T17:22:20.1927362 0 [ERROR] unknown variable 'error-log=0'
2017-08-29T17:22:20.1927362 0 [ERROR] Aborting

2017-08-29T17:22:20.1927362 0 [Note] Binlog end
2017-08-29T17:22:20.1927362 0 [Note] Shutting down plugin 'ngram'
2017-08-29T17:22:20.1927362 0 [Note] Shutting down plugin 'partition'
2017-08-29T17:22:20.1927362 0 [Note] Shutting down plugin 'BLACKHOLE'
2017-08-29T17:22:20.1927362 0 [Note] Shutting down plugin 'ARCHIVE'
2017-08-29T17:22:20.1927362 0 [Note] Shutting down plugin 'PERFORMANCE_SCHEMA'
2017-08-29T17:22:20.1927362 0 [Note] Shutting down plugin 'MRG_MYISAM'
2017-08-29T17:22:20.1927362 0 [Note] Shutting down plugin 'MyISAM'
2017-08-29T17:22:20.1927362 0 [Note] Shutting down plugin 'INNODB SYS VIRTUAL'
```

شکل ۳ نمونه‌ای از فایل رویدادنگار خطا

با استفاده از متغیر سیستمی log_error_verbosity می‌توان سطح پیغام‌های error یا note یا warning را کنترل کرد. مقادیر مجاز شامل موارد زیر است:

- مقدار ۱: فقط خطاها
- مقدار ۲: خطاها و هشدارها
- مقدار ۳: خطاها، هشدارها، یادداشت‌ها (مقدار پیش‌فرض)

به‌منظور تغییر مقدار پیش‌فرض و مشاهده‌ی مقدار جاری این متغیر از دستورات زیر استفاده می‌شود.

```
SET GLOBAL log_error_verbosity=2;
SELECT @@log_error_verbosity;
```

همچنین می‌توان در فایل my.ini پارامتر log_error_verbosity را با مقدار دلخواه وارد کرد.

```
log_error_verbosity = 2
```

۲-۱-۲ General query log

در General query log دو نوع اطلاعات ثبت می‌شوند:

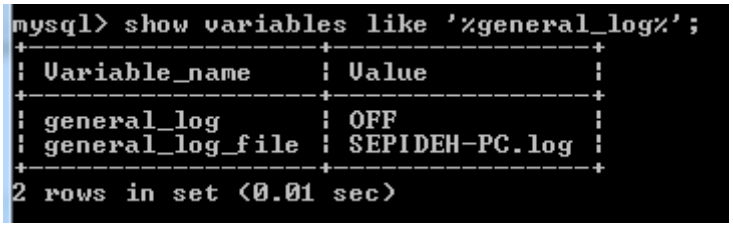
۱- اطلاعات مربوط به اتصال و قطع اتصال کارخواه‌ها[8]

۲- عبارات SQL دریافت شده از کارخواه‌ها

رکوردهایی که نشان‌دهنده‌ی اتصال کارخواه هستند، شامل قسمتی برای مشخص کردن نوع اتصال، یعنی پروتکلی که برای برقراری اتصال استفاده‌شده است (همچون TCP/IP، SSL/TLS، socket، Named pipe یا Shared memory)، نیز هستند. عبارات SQL نیز به ترتیب دریافت آن‌ها توسط کارگزار ثبت می‌شوند؛ این ترتیب ممکن است نسبت به ترتیب اجرا متفاوت باشد.

به‌صورت پیش‌فرض general query log در MySQL غیرفعال است [۱]. به‌منظور شناخت وضعیت general query log باید دستوری مطابق شکل ۴ اجرا شود.

[8] Clients



شکل ۴ بررسی وضعیت general query log

در فایل my.ini با سه پارامتر می‌توان general query log را کنترل کرد [۲]:

- با استفاده از پارامتر general-log می‌توان general query log را فعال یا غیرفعال کرد:

```
general-log[={0,1}]
```

- پارامتر general_log_file نام فایل رویدادنگاری را نشان می‌دهد که به‌صورت پیش‌فرض host_name.log است.
- پارامتر log_output، مقصد خروجی general query log را تعریف می‌کند که می‌تواند یکی از مقادیر FILE، TABLE یا NONE باشد. درصورتی‌که FILE انتخاب شود، رویدادها در فایل مشخص‌شده در پارامتر general_log_file ثبت می‌شوند. با انتخاب TABLE، رویدادها در جدول ثبت‌شده و NONE باعث غیرفعال شدن general query log می‌شود.

همچنین می‌توان با دستورات زیر general query log را تنظیم کرد:

```
SET global general_log = 1;
SET global log_output = TABLE;
```

در صورت انتخاب TABLE برای پارامتر log_output می‌توان رویدادها را با پرسمان زیر مشاهده کرد (شکل):

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

```
mysql> select * from mysql.general_log;
+----------------------------+-----------------------------------+-----------+-------
----+--------------+-------------------------------------------------------------------
-------------------------------------------+
| event_time                 | user_host                         | thread_id | server
_id | command_type | argument
                                        |
+----------------------------+-----------------------------------+-----------+-------
----+--------------+-------------------------------------------------------------------
-------------------------------------------+
| 2017-09-11 22:43:17.487029 | [boot] @  []                      |         2 |
  1 | Query        | SELECT TABLE_SCHEMA, TABLE_NAME FROM INFORMATION_SCHEMA.TAB
LES WHERE CREATE_OPTIONS LIKE '%partitioned%'; |
| 2017-09-11 22:43:49.343306 | [root] @ localhost [::1]          |         3 |
  1 | Connect      | root@localhost on  using TCP/IP
                                        |
| 2017-09-11 22:43:49.358906 | root[root] @ localhost [::1]      |         3 |
  1 | Query        | show variables like '%general_log%'
                                        |
| 2017-09-11 22:44:02.022749 | root[root] @ localhost [::1]      |         3 |
  1 | Query        | select * from mysql.general_log
                                        |
+----------------------------+-----------------------------------+-----------+-------
----+--------------+-------------------------------------------------------------------
-------------------------------------------+
4 rows in set (0.00 sec)
```

شکل ۶ general query log در جدول

همچنین می‌توان log_output را برای نوشتن اطلاعات بر روی فایل تنظیم کرد:

```
SET global general_log_file='/tmp/mysql.log';
SET global log_output = FILE;
SET global general_log = ۱;
```

نمونه‌ای از این فایل در شکل ۵ نشان داده شده است:

```
2017-09-11T17:51:07.028674Z       3 Query show variables like
'%general_log%'
2017-09-11T17:52:29.995727Z       3 Query select * from test.test
```

شکل ۵ general query log در فایل

۳-۱-۲ Binary log

Binary log شامل رویدادهایی است که تغییرات پایگاه داده را توصیف می‌کنند (همچون عملیات ایجاد جدول یا تغییر در داده‌های جدول). این نوع رویداد همچنین شامل عباراتی است که پتانسیل ایجاد تغییر را دارند همچون یک عبارت DELETE که با هیچ سطری منطبق نشود.

Binary log دو هدف کلی و مهم دارد:

۱- **تکرار**[۹]. بدین منظور، binary log در کارگزار ارشد تکرار تغییرات داده‌ها را برای ارسال به کارگزاران slave فراهم می‌کند.

۲- **بازیابی**: برخی از عملیات بازیابی داده‌ها نیاز به استفاده از binary log دارد. پس‌از آنکه، پشتیبان بازیابی می‌شود، رویدادهای درون binary log که پس از ایجاد پشتیبان ثبت‌شده‌اند، مجدداً اجرا می‌شوند. اجرای این رویدادها، پایگاه داده را از نقطه پشتیبان، به‌روز می‌کند.

Binary log برای عباراتی همچون SELECT یا SHOW که داده‌ها را تغییر نمی‌دهند، کاربرد ندارد. در صورت فعال کردن Binary log بر روی کارگزار، عملکرد کمی کند می‌شود؛ با این وجود فراهم آوردن امکان تکرار و بازیابی عملیات، مهم‌تر از کاهش ناچیز عملکرد است [۱].

فایل‌های binary log دارای نامی با ساختار host_name-bin.nnnnnn و پسوند عددی هستند. این فایل‌ها در کنار خود فایل شاخصی با ساختار host_name-bin.index دارند. فایل شاخص، نام تمامی فایل‌های binary log مورداستفاده را شامل می‌شود[۳]:

ثبت در Binary log بلافاصله بعد از کامل شدن عبارت یا تراکنش و پیش از آزاد شدن قفل‌ها یا اجرای commit، انجام می‌شود. بدین ترتیب می‌توان مطمئن بود که ثبت در فایل binary log به ترتیب commit انجام می‌شود [۲].

با استفاده از دستور زیر وضعیت فعال یا غیرفعال بودن binary log چک می‌شود (شکل):

```
SELECT @@log_bin;
```

[9] Replication

شکل۸ بررسی وضعیت binary log

به‌منظور فعال‌سازی binary log در فایل my.ini پارامتر log-bin به همراه مسیر ذخیره‌سازی فایل‌های رویدادنگاری، مقداردهی می‌شود:

```
log-bin=C:\mySqlbinlog
```

همچنین درصورتی‌که برای پارامتر log-bin مقدار تعیین نشود، فایل‌های Binary log به‌صورت پیش‌فرض در دایرکتوری data نوشته می‌شوند.

۲-۱-٤ Relay log

تغییرات داده که از کارگزار ارشد تکرار دریافت می‌شوند در relay log ثبت می‌گردند. فایل‌های رویدادنگاری، نام‌هایی با ساختار host_name-relay-bin.nnnnnn دارند. این فایل‌ها با فایل شاخصی با ساختار نام host_name-relay-bin.index همراه هستند. این نوع فایل‌های رویدادنگاری تنها بر روی کارگزاران slave هستند و تغییرات داده‌ها از جانب کارگزار ارشد را در خود نگه می‌دارند. ساختار این فایل‌های رویدادنگاری شبیه فایل‌های binary log هستند [۲].

۲-۱-۵ Slow query log

شامل عبارات SQL است که بیش از long_query_time ثانیه، زمان برای اجرا و حداقل min_examined_row_limit سطر برای ارزیابی نیاز دارند. حداقل و مقدار پیش‌فرض برای long_query_time به ترتیب صفر و ۱۰ ثانیه هستند. مقدار تعیین‌شده می‌تواند دقت میکروثانیه نیز داشته باشد. در حقیقت با استفاده از این نوع فایل‌های رویدادنگاری می‌توان پرسمان‌هایی که زمان زیادی برای اجرا نیاز دارند را شناسایی و بهینه کرد. زمانی که به‌منظور دریافت قفل‌های اولیه صرف می‌شود، به‌عنوان

زمان اجرا در نظر گرفته نمی‌شود. کارگزار MySQL عبارتی را پس از اجرا و آزاد شدن تمامی قفل‌ها در Slow query log می‌نویسد؛ بنابراین ترتیب ثبت در فایل رویدادنگاری با ترتیب اجرا می‌تواند متفاوت باشد[۱]. درصورتی‌که نام فایل برای ثبت مشخص نشود، نام فایل رویدادنگاری به‌صورت پیش‌فرض host_name-slow.log خواهد بود. وضعیت این نوع ثبت رخدادها مطابق شکل ۶ بررسی می‌شود.

```
mysql> show variables like '%slow_query_log%';
+---------------------+----------------------+
| Variable_name       | Value                |
+---------------------+----------------------+
| slow_query_log      | ON                   |
| slow_query_log_file | SEPIDEH-PC-slow.log  |
+---------------------+----------------------+
2 rows in set (0.02 sec)
```

شکل ۶ بررسی وضعیت slow query log

به‌منظور تنظیم Slow query log از پارامترهای زیر در فایل my.ini استفاده می‌شوند:

```
log-output=FILE
slow-query-log=1
slow_query_log_file="SEPIDEH-PC-slow.log"
long_query_time=10
```

۲-۱-۶ DLL log

DLL log یا metadata log عملیات فراداده‌ی مربوط به عبارات تعریف داده[10] همچون DROP TABLE و ALTER TABLE را ثبت می‌کند. درصورتی‌که در حین عملیات فراداده، MySQL دچار مشکل شود، می‌تواند از این نوع ثبت رخداد برای ترمیم شرایط، استفاده کند.

رکوردی از عملیات فراداده در فایل ddl_log.log ثبت می‌شود. این فایل دودویی است و نباید محتوای آن توسط افراد تغییر داده شود [۱].

[10] Data definition statements

۳ نحوه‌ی انجام ممیزیدر پایگاه داده‌ی MySQL

ممیزی[11] در پایگاه داده شامل مشاهده و نظارت بر پایگاه داده به منظور آگاهی از اقدامات کاربران پایگاه داده است.

این بخش بر روی ویندوز ۷ و MySQL 5.7 (نسخه‌ی Enterprise) تهیه شده است. نسخه MySQL Enterprise شامل MySQL Enterprise Audit است که به‌صورت پلاگینی با نام audit_log پیاده‌سازی شده است. با استفاده از MySQL Enterprise Audit می‌توان بر اتصالات و پرسمان‌هایی که بر روی کارگزار MySQL اجرا می‌شوند، نظارت کرد و رخدادها را ثبت کرد. زمانی که پلاگین نصب شود، پلاگین این امکان را برای کارگزار MySQL فراهم می‌کند که فایل رویدادنگاری حاوی رکوردهای فعالیت‌های کارگزار تولید شود. محتوای فایل رویدادنگاری شامل اتصال و قطع اتصال کارخواه از کارگزار و فعالیت‌هایی است که در حین اتصال، کارخواه انجام می‌دهد. فایل رویدادنگاری پس از نصب پلاگین در دایرکتوری Data با نام audit.log ایجاد می‌شود. این فایل دارای فرمت XML است و محتوای آن، رمز شده نیست. این فایل می‌تواند شامل اطلاعات حساس همچون متن عبارات SQL باشد. برای امنیت بیشتر این فایل باید در دایرکتوری نوشته شود که تنها برای کارگزار MySQL و کاربران مجاز قابل‌دسترسی باشد [۱].

به‌منظور نصب پلاگین ابتدا باید از موجود بودن نسخه‌ی MySQL Enterprise اطمینان حاصل شود (مطابق با شکل ۷).

[11] Auditing

```
mysql> show variables like 'version%';
+-------------------------+-----------------------------------------------------
----+
| Variable_name           | Value
    |
+-------------------------+-----------------------------------------------------
----+
| version                 | 5.7.19-enterprise-commercial-advanced-log
    |
| version_comment         | MySQL Enterprise Server - Advanced Edition (Commerci
al) |
| version_compile_machine | AMD64
    |
| version_compile_os      | Win32
    |
+-------------------------+-----------------------------------------------------
----+
4 rows in set (0.01 sec)
```

شکل ۷ بررسی نسخه‌ی MySQL

به‌منظور بارگذاری پلاگین و ثبت آن در جدول سیستمی mysql.plugins با هدف بارگذاری پلاگین در راه‌اندازی‌های بعدی کارگزار، دستور زیر اجرا می‌شود:

```
INSTALL PLUGIN audit_log SONAME 'audit_log.dll';
```

حال با دستور زیر می‌توان وضعیت نصب پلاگین را بررسی کرد (شکل ۸):

```
SELECT PLUGIN_NAME, PLUGIN_STATUS
    FROM INFORMATION_SCHEMA.PLUGINS
     WHERE PLUGIN_NAME LIKE 'audit%';
```

```
mysql> select plugin_name,plugin_status from information_schema.plugins where pl
ugin_name like 'audit%';
+-------------+---------------+
| plugin_name | plugin_status |
+-------------+---------------+
| audit_log   | ACTIVE        |
+-------------+---------------+
1 row in set (0.00 sec)
```

شکل ۸ بررسی وضعیت نصب پلاگین audit_log

پس‌ازآن فایل audit.log در دایرکتوری Data ایجاد می‌شود. نمونه‌ای از این فایل در شکل ۹ نشان داده شده است.

```
<AUDIT_RECORD>
 <TIMESTAMP>2017-09-12T17:04:21 UTC</TIMESTAMP>
 <RECORD_ID>12_2017-09-12T16:47:37</RECORD_ID>
 <NAME>Query</NAME>
 <CONNECTION_ID>5</CONNECTION_ID>
 <STATUS>0</STATUS>
 <STATUS_CODE>0</STATUS_CODE>
 <USER>root[root] @ localhost [::1]</USER>
 <OS_LOGIN/>
 <HOST>localhost</HOST>
 <IP>::1</IP>
 <COMMAND_CLASS>create_table</COMMAND_CLASS>
 <SQLTEXT>create table mysql.test(a varchar(10))</SQLTEXT>
</AUDIT_RECORD>
<AUDIT_RECORD>
 <TIMESTAMP>2017-09-12T17:04:58 UTC</TIMESTAMP>
 <RECORD_ID>13_2017-09-12T16:47:37</RECORD_ID>
 <NAME>Query</NAME>
 <CONNECTION_ID>5</CONNECTION_ID>
 <STATUS>0</STATUS>
 <STATUS_CODE>0</STATUS_CODE>
 <USER>root[root] @ localhost [::1]</USER>
 <OS_LOGIN/>
 <HOST>localhost</HOST>
 <IP>::1</IP>
 <COMMAND_CLASS>insert</COMMAND_CLASS>
 <SQLTEXT>insert into mysql.test values('test')</SQLTEXT>
</AUDIT_RECORD>
<AUDIT_RECORD>
 <TIMESTAMP>2017-09-12T17:05:29 UTC</TIMESTAMP>
 <RECORD_ID>14_2017-09-12T16:47:37</RECORD_ID>
 <NAME>Query</NAME>
 <CONNECTION_ID>5</CONNECTION_ID>
 <STATUS>0</STATUS>
 <STATUS_CODE>0</STATUS_CODE>
 <USER>root[root] @ localhost [::1]</USER>
 <OS_LOGIN/>
 <HOST>localhost</HOST>
 <IP>::1</IP>
 <COMMAND_CLASS>select</COMMAND_CLASS>
 <SQLTEXT>select * from mysql.test</SQLTEXT>
</AUDIT_RECORD>
```

شکل ۹ نمونه‌ای از فایل audit.log

٤ ابزارهای جرم‌شناسی

برای پایگاه داده MySQL ابزار جرم‌شناسی اختصاصی مفیدی، تابه‌حال یافت نشده است. در این بخش ابزار DB Audit and security 360 که از پایگاه‌های داده مختلف ازجمله MySQL پشتیبانی می‌کند، توضیح داده می‌شود.

٤-١ ابزار DB Audit and security 360

ابزار DB Audit and security 360، راه‌حل حرفه‌ای برای امنیت، ممیزی و ارزیابی امنیتی پایگاه داده است. این ابزار از پایگاه‌های داده اوراکل، Sybase، DB2، MySQL و MSSQL Server پشتیبانی می‌کند. DB Audit به مدیران پایگاه داده و سیستم، مدیران امنیتی، حسابرسان[12] و اپراتورها این امکان را می‌دهد که به‌درستی از سیستم‌های پایگاه داده حفاظت و فعالیت‌های پایگاه داده شامل دسترسی به پایگاه داده، ایجاد داده، تغییر و حذف داده‌ها را به صورت به‌هنگام[13] ردیابی و تجزیه‌وتحلیل کنند[۴]. این ابزار متن‌باز نیست و دارای نسخه‌ی آزمایشی ۳۰ روزه با کلیه‌ی قابلیت‌ها و ویژگی‌ها است.

ویژگی‌های کلیدی این ابزار شامل موارد زیر است:

- بهبود امنیت سیستم و اطمینان از پاسخگویی سیستم
- کنترل متمرکز ممیزی و امنیت چندین سیستم پایگاه داده از یک مکان واحد و فراهم آوردن مدیریت آسان
- فراهم آوردن گزارش‌های تحلیلی و تبدیل حجم زیادی از داده‌های ممیزی به گزارش‌های خلاصه و جامع و قابلیت شناسایی نقض‌های امنیتی پایگاه داده
- فراهم آوردن گزارش‌های تحلیلی که به شناسایی فرایندها و کاربران مصرف‌کننده‌ی منابع سیستم منجر می‌شوند

[12] Auditor

[13] Real time

- فراهم آوردن ممیزی با جزئیات. ممیزی‌های فراهم‌شده نسبت به ممیزی‌های تهیه شده توسط ابزار ممیزی خود پایگاه داده، جزئیات بیشتری دارد.
- امکان ارسال ایمیل هشدار به افراد کلیدی در صورت وقوع تغییری در داده‌های حساس
- عدم نیاز DBA به ایجاد و مدیریت تریگرها برای تهیه ممیزی از تغییر داده‌ها
- پشتیبانی از تنظیمات ممیزی منعطف
- فراهم آوردن ممیزی از تغییر داده و ممیزی در سطح سیستم به صورت نهان[14] از دید برنامه‌های کاربردی، یعنی بدون نیاز به تغییر در برنامه‌های کاربردی

۵ جمع‌بندی

کارگزار MySQL دارای فایل‌های رویدادنگاری گوناگونی است که با کمک آن‌ها می‌توان وقوع فعالیت‌های مختلف را پیگیری کرد. همچنین نسخه MySQL Enterprise شامل MySQL Enterprise Audit است که به‌صورت پلاگینی با نام audit_log پیاده‌سازی شده است. با استفاده از MySQL Enterprise Audit می‌توان بر اتصالات و پرسمان‌هایی که بر روی کارگزار MySQL اجرا می‌شوند، نظارت کرد و رخدادها را ثبت کرد. برای پایگاه داده MySQL ابزار جرم‌شناسی اختصاصی مفیدی، تابه‌حال یافت نشده است و تنها ابزار DB Audit and security 360 که از پایگاه‌های داده مختلف ازجمله MySQL پشتیبانی می‌کند، به این منظور می‌تواند مورداستفاده قرار گیرد.

٦ منابع

[1]. https://dev.mysql.com/doc/refman/5.7/en/

[2]. http://howtolamp.com/lamp/mysql/5.6/log-files/

[3]. https://www.safaribooksonline.com/library/view/mysql-reference-manual/0596002653/ch04s09.html

[4]. DB Audit and Security 360, version 5.0, user's guide, SoftTree technologies, Inc. 2015

[14] Transparent