Security and Compliance with MySQL

Mike Frank, MySQL Product Management Director | Oracle
Agenda

• Compliance Overview
• How to Examples
• Architectural Review
• Latest Enhancements
• Security Guidelines
  • MySQL Secure Deployment Guide
  • CIS Benchmark for MySQL 8.0 EE
  • DISA STIG
• Resources
• Tell us what you need
Security is Job #1
Data is the Most Valuable Asset

“Keep the organization safe (cybersecurity/cyber resilience/GDPR compliance/data protection compliance) “

Almost all breaches - preventable.
Data Security & Privacy Regulations are Proliferating

- NCUA
- FS-ISAC
- CCPA
- FFIEC
- Patriot Act
- Dodd Frank
- 50 State Data Privacy Laws

Countries and Regulations:
- EU GDPR
- Ru DPA
- Cn GDPL
- Jp APPI
- Ru DPA
- In PDPB
- Hk PDPO
- Th PDA
- Si PDPA
- Au APP
- Nz PA

- PCI
- Co DPL
- Br GDPL
- Sa ECTA
- CI PPL
- Ar PDPL
- Mx PDPL
- GLBA
- NY DFS500
- HIPAA
- CIP
- PIPEDA
- FOIPPA
- GLBA
- PCI
- NY DFS500
- HIPAA
- PIPEDA
- FOIPPA

- BASEL III
- Dodd Frank
Data Breaches – keep increasing

2021 a record year for data breaches

Manufacturing & utilities 48 compromises and a total of 48,294,629 victims.

1,291 breaches in 2021 compared to 1,108 breaches in 2020

Healthcare sector 78 compromises and over 7 million victims.

Regulations require these Security Steps

- **Assess**
  - Locate Risks and Vulnerabilities, Ensure that necessary security controls are

- **Prevent**
  - Using Cryptography, User Controls, Access Controls, etc

- **Detect**
  - Still a possibility of a breach – so Audit, Monitor, Alert

- **Recover**
  - Ensure service is not interrupted as a result of a security incident
  - Even through the outage of a primary database
  - Forensics – postmortem – fix vulnerability
MySQL Security Overview

- Authentication
- Authorization
- Encryption
- Firewall
- Auditing
- Masking/De-Identification

https://www.mysql.com/products/enterprise/
Enterprise Security Architecture

- **Enterprise Authentication**
  - SSO - LDAP, AD, PAM, Native Kerberos, FIDO2, WebAuthn, MFA

- **Enterprise Monitor**

- **Enterprise Masking & De-Identification**
  - Masking
  - Substitute/Subset
  - Random Formatted Data
  - Blacklisted Data

- **Access Controls**

- **Firewall**

- **Gateway**
  - Attack minimization

- **Network Encryption**
- **Strong Authentication**

- **Enterprise Audit**
  - Powerful Rules Engine

- **Audit Vault**

- **Enterprise Backup**
  - Encrypted

- **HA**
  - InnoDB Cluster

- **Data Encryption**
  - TDE
  - Encryption
  - PKI

- **Key Vault**
  - Protect Keys (KMIP, Rest APIs)

- **Thread Pool**
  - Attack minimization

- **Users**

- **Enterprise Masking & De-Identification**

- **Assess**
- **Prevent**
- **Detect**
- **Recover**
How to Examples
Much can be done using SQL and ...

1. DBA does not need to SSH/Login to the OS where mysql is running
   This is common.

2. All DBA actions must be audited
   MySQL Auditing can capture all statements performed by DBAs via SQL.

3. OS Admins don't need to be touching MySQL
   OS Auditing should show little past the initial installation
   Commands not exposed


5. Great for repeatable assessment and fix automation.
First thing
Secure the Root password

Did you type in the initial root password
  Depends on the installation package
  Windows Installer and DEB Packages prompt
  RPM does not

In not immediately reset the root password
Read the Post-Installation Instructions

A random password has been set for the MySQL root user! You will find that password in `/root/.mysql_secret`

You must change that password on your first connect. No other statement but 'SET PASSWORD' will be accepted. See the manual for the semantics of the 'password expired' flag.

Also, the account for the anonymous user has been removed.

In addition, you can run:

```
/usr/bin/mysql_secure_installation
```

which will also give you the option of removing the test database. This is strongly recommended for production servers. See the manual for more instructions.

Please report any problems at http://bugs.mysql.com/

The latest information about MySQL is available on the web at

```
http://www.mysql.com
```

Support MySQL by buying support/licenses at http://shop.mysql.com

A new default config file was created as /usr/my.cnf and will be used by default by the server when you start it. You may edit this file to change server settings:

```
[mysqld]
```

(host@localhost:19560)
Reset the “root” Password

```bash
$ mysql -u root --password=`sudo cat /root/.mysql_secret | cut -c 87-`
SQL> SELECT 1
Must fail with "you must set password"
SQL> ALTER USER root@localhost IDENTIFIED BY '<auth_string>';
SQL> EXIT;
$ mysql -u root --password
  enter your new password at the prompt
SQL> EXIT;
```
SELECT user, host FROM mysql.user where user='root' and host<>'localhost';

Multiple root accounts!

Is the host name constrained or is it global – '%'

Remove and ”global” host roots. Limit access if remote is necessary.

Note that only root@localhost is with a changed password!
Password Policies In Place?

IS THE COMPONENT INSTALLED?

SELECT component_urn, 'PASSWORD Policy Component Installed?' as Note,
    if(count(component_urn) > 0, 'YES', 'NO') as Answer
FROM mysql.component
where component_urn='file://component_validate_password'
group by component_urn;

<table>
<thead>
<tr>
<th>component_urn</th>
<th>Note</th>
<th>Answer</th>
</tr>
</thead>
<tbody>
<tr>
<td>file://component_validate_password</td>
<td>PASSWORD Policy Component Installed?</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Password Policies

```
SELECT VARIABLE_NAME, VARIABLE_VALUE
FROM performance_schema.global_variables
where VARIABLE_NAME like 'valid%password%'
OR VARIABLE_NAME='default_password_lifetime';
```
Change my password policy

If needed

INSTALL COMPONENT
'file://component_validate_password';

Set Password Policies

set persist
validate_password.check_user_name='ON';

set persist
validate_password.dictionary_file='<FILENAME OF DICTIONARY FILE';

set persist validate_password.length=15;

set persist
validate_password.mixed_case_count=1;

set persist
validate_password.special_char_count=2;

set persist validate_password.number_count=2;
set persist validate_password.policy='STRONG';
set persist password_history = 5;
set persist password_reuse_interval = 365;
Set global default_password_lifetime = 180;

Additionally maybe for password reset
set persist password_require_current=YES

Note some things can be set per account.
ALTER USER 'jeffrey'@'localhost'
  PASSWORD HISTORY 5
  PASSWORD REUSE INTERVAL 365 DAY;
ALTER USER 'jeffrey'@'localhost' PASSWORD EXPIRE INTERVAL 90 DAY;
MySQL Connection Controls

Dealing with Failed Login Attempts related to Brute Force Attacks

Are the Connection Controls Plugins in place?

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

<table>
<thead>
<tr>
<th>PLUGIN_NAME</th>
<th>PLUGIN_STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONNECTION_CONTROL</td>
<td>ACTIVE</td>
</tr>
<tr>
<td>CONNECTION_CONTROL_FAILED_LOGIN_ATTEMPTS</td>
<td>ACTIVE</td>
</tr>
</tbody>
</table>

Check Settings

```sql
select @@connection_control_failed_connections_threshold, @@connection_control_min_connection_delay, @@connection_control_max_connection_delay, @@connection_control_failed_connections_threshold;
```
Installing and Setting Connection Controls

Install and Set

INSTALL PLUGIN CONNECTION_CONTROL SONAME 'connection_control.so';
INSTALL PLUGIN CONNECTION_CONTROL_FAILED_LOGIN_ATTEMPTS SONAME 'connection_control.so';

For example

SET PERSIST connection_control_failed_connections_threshold = 4;
SET PERSIST connection_control_min_connection_delay = 1500;

Use your CA
MySQL Installers create self signed keys
Better if you generate and replace from your Certificate Authority

```sql
select 'ALL SSL VARIABLES Listing' as NOTE, @@ssl_ca, @@ssl_capath, @@ssl_cert, @@ssl_cipher, @@ssl_crl, @@ssl_crlpath, @@ssl_fips_mode, @@ssl_key;
```

Note: MySQL 8.0.16 now allows you to change SSL options without a restart.
Prepares a new SSL context for the listening socket and then replaces the old ones.
Generate your new pem files – put them in place - then

```
ALTER INSTANCE RELOAD TLS;
```

FIPs Required
See if it's on or not

```
SELECT VARIABLE_NAME, VARIABLE_VALUE, FIPS Mode' as Note, IF(VARIABLE_VALUE = 'ON' OR VARIABLE_VALUE = 'STRICT', 'Yes', 'No') FROM performance_schema.global_variables where variable_name = 'ssl_fips_mode';
```
SSL Required?

Force it globally

```
SELECT VARIABLE_NAME, VARIABLE_VALUE, 'ONLY ALLOW SSL ' as Note,
IF(VARIABLE_VALUE = 'ON', 'PASS', 'FAIL') AS CHECK_VAL
FROM performance_schema.global_variables
WHERE VARIABLE_NAME IN ('require_secure_transport');
```

<table>
<thead>
<tr>
<th>VARIABLE_NAME</th>
<th>VARIABLE_VALUE</th>
<th>Note</th>
<th>CHECK_VAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>require_secure_transport</td>
<td>ON</td>
<td>MUST ONLY ALLOW SSL CONNECTIONS</td>
<td>PASS</td>
</tr>
</tbody>
</table>

FORCE encrypted connections globally

set persist require_secure_transport=ON;
In MySQL 8.0 DBAs can set system variables from SQL

The value of SET PERSIST is written to mysqld-auto.cnf

SET PERSIST ONLY – stores to mysqld-auto.cnf without setting the runtime value.
   Use for configuring read-only system variables that can be set only at server startup.

A few system variables can't be set using this command

Need to be even more Secure –
   Install a MySQL Keyring then
   persist sensitive variables in plaintext=ON

   When set the server encrypts the values of any sensitive system variables

This file is in the datadir
Less accessible than my.cnf
Added security

Epoch timestamped
Track change times
1564600430679850
Mon, 26 Aug 2019 17:57:47 GMT

```json
{
    "Version": 1,
    "mysql_server": {
        "require_secure_transport": {
            "Value": "ON",
            "Metadata": {
                "Timestamp": 1564600430679850,
                "User": "root",
                "Host": "localhost"
            }
        },
        "validate_password.dictionary_file": {
            "Value": "<FILENAME OF DICTIONARY FILE",
            "Metadata": {
                "Timestamp": 1564598898444506,
                "User": "root",
                "Host": "localhost"
            }
        },
        "authentication_ldap_sasl_server_host": {
            "Value": "127.0.0.1",
            "Metadata": {
                "Timestamp": 1564695043687370,
                "User": "root",
                "Host": "localhost"
            }
        },
        "authentication ldap sasl bind base dn": {
```
Permitting import and export operations
Turn off what you are using – reduce the attack surface

```
SELECT VARIABLE_NAME, VARIABLE_VALUE, 'Secure File Check' as Note,
IF(length(VARIABLE_VALUE) > 0 and VARIABLE_VALUE!='NULL', 'FAIL', 'PASS') as SecFileCheck
FROM performance_schema.global_variables
where variable_name = 'secure_file_priv';
```
LOCAL Load Data INFILE
Secure by default - OFF

Check local_infile

```
select if(@@local_infile, 'ON', 'OFF') as LOCAL_LOAD_DATA_ALLOWED;
```

By Default in 8.0 this is off

```
set persist local_infile=OFF;
```
USERS
Who, What Kind, Where/How do they authenticate

Internal Users
Internal using X.509
Externally Authenticating Users
Proxy Users
Internal Users
Authenticated internally

```
select host, user, plugin,
if(plugin = 'mysql_native_password', 'WEAK SHA1', 'STRONG SHA2') AS HASHTYPE
FROM mysql.user
WHERE user not in ('mysql.infoschema', 'mysql.session')
and (plugin not like 'auth%' and plugin <> 'mysql_no_login')
and length(authentication_string) > 0
order by plugin;
```

<table>
<thead>
<tr>
<th>host</th>
<th>user</th>
<th>plugin</th>
<th>HASHTYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>%</td>
<td>mysql.yso</td>
<td>caching_sha2_password</td>
<td>STRONG SHA2</td>
</tr>
<tr>
<td>%</td>
<td>newusarq</td>
<td>caching_sha2_password</td>
<td>STRONG SHA2</td>
</tr>
<tr>
<td>%</td>
<td>newusuwn</td>
<td>caching_sha2_password</td>
<td>STRONG SHA2</td>
</tr>
<tr>
<td>localhost</td>
<td>dev1</td>
<td>caching_sha2_password</td>
<td>STRONG SHA2</td>
</tr>
<tr>
<td>localhost</td>
<td>mysql.sys</td>
<td>caching_sha2_password</td>
<td>STRONG SHA2</td>
</tr>
<tr>
<td>localhost</td>
<td>read_user1</td>
<td>caching_sha2_password</td>
<td>STRONG SHA2</td>
</tr>
<tr>
<td>localhost</td>
<td>read_user2</td>
<td>caching_sha2_password</td>
<td>STRONG SHA2</td>
</tr>
<tr>
<td>localhost</td>
<td>root1</td>
<td>caching_sha2_password</td>
<td>STRONG SHA2</td>
</tr>
<tr>
<td>localhost</td>
<td>nr_norole_user</td>
<td>caching_sha2_password</td>
<td>STRONG SHA2</td>
</tr>
<tr>
<td>localhost</td>
<td>nw_user1</td>
<td>caching_sha2_password</td>
<td>STRONG SHA2</td>
</tr>
<tr>
<td>%</td>
<td>newusar</td>
<td>mysql_native_password</td>
<td>WEAK SHA1</td>
</tr>
<tr>
<td>%</td>
<td>newusuarr</td>
<td>mysql_native_password</td>
<td>WEAK SHA1</td>
</tr>
<tr>
<td>%</td>
<td>root2</td>
<td>mysql_native_password</td>
<td>WEAK SHA1</td>
</tr>
<tr>
<td>%</td>
<td>root3</td>
<td>mysql_native_password</td>
<td>WEAK SHA1</td>
</tr>
</tbody>
</table>

To Do's
- Lock Accounts that are unknown – then drop once sure
- Drop and create new user accounts with stricter host specification
- Users with native typically are from MySQL 5.7 upgrade to 8.0

https://mysqlserverteam.com/mysql-8-0-4-new-default-authentication-plugin-caching_sha2_password/
Internal Users

REQUIRING X509 CERTIFICATE

SELECT `user`.`Host`, `user`.`User`, `user`.`ssl_type`,
CAST(`user`.`x509_issuer` as CHAR) as Issuer,
CAST(`user`.`x509_subject` as CHAR) as Subject
FROM `mysql`.`user` where (user not like 'mysql.%') AND ssl_type='X509';
External Authentication
Globally manage – map to Enterprise, Use stronger Options
LDAP, Windows AD SSPI, Kerberos, FIDO2 – Many Options

SELECT `user`.`Host`, `user`.`User`, `user`.`plugin`, `user`.`authentication_string` from mysql.user where plugin like 'auth%';

<table>
<thead>
<tr>
<th>Host</th>
<th>User</th>
<th>plugin</th>
<th>authentication_string</th>
</tr>
</thead>
<tbody>
<tr>
<td>localhost</td>
<td>betsy</td>
<td>authentication_ldap_simple</td>
<td>uid=betsy_ldap,ou=People,dc=example,dc=com</td>
</tr>
<tr>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
<td>NULL</td>
</tr>
</tbody>
</table>

Many companies are going to external authentication – especially for internal users – DBAs and Developers
Map and manage in LDAP, Actual User in Audit Trail
Make sure users or mapped organizations should have MySQL Access.
Multi-Factor Authentication

Up to 3 – various regulations requiring MFA, 2FA, ...

PCI DSS 8.3 for example

Create with 2

CREATE USER 'alice'@'localhost' IDENTIFIED WITH caching_sha2_password BY 'sha2_password' AND IDENTIFIED WITH authentication_ldap_sasl AS 'uid=u1_ldap,ou=People,dc=example,dc=com';

Can add a second or here a third factor later with ALTER

ALTER USER 'alice'@'localhost' ADD 3 FACTOR IDENTIFIED WITH authentication_fido;

“Assure that strong multi-factor authentication is pervasive to protect against common attacks against the credentials of consumers, merchants, and service providers”

“The PCI DSS requires multi-factor authentication (MFA) mechanism for remote access to the Cardholder Data Environment (CDE).”

https://www.pcisecuritystandards.org/documents/PCIDSS_QRGv3_1.pdf
Roles and Proxy Users

SELECT * FROM mysql.proxies_priv where grantor<>'boot@';

To inspect specific user, role or user using role
User or for a role
SHOW GRANTS FOR 'app_developer'@'%';

User with Role
SHOW GRANTS FOR 'u1'@'localhost' USING 'r1';
User Rights
Max Connections

For example if your company policy is MAX 210

```sql
SELECT VARIABLE_NAME, VARIABLE_VALUE, 'MUST be 210 or less' as Note,
IF(VARIABLE_VALUE < 211, 'PASS', 'FAIL')
FROM performance_schema.global_variables WHERE VARIABLE_NAME LIKE 'max_connections';

If the result is FAIL – then FIX

SET PERSIST max_connections = 210;
```
User Rights

Granted Permissions

MySQL Schema

db
tables_priv;
columns_priv;
procs_priv;
roles;
users

Information Schema (VIEWS)

user_privileges
table_privileges
schema_privileges
column_privileges
## MySQL Workbench Schema Inspector - Grants

<table>
<thead>
<tr>
<th>Host</th>
<th>User</th>
<th>Scope</th>
<th>Select</th>
<th>Insert</th>
<th>Update</th>
<th>Delete</th>
<th>Create</th>
<th>Drop</th>
<th>Grant</th>
<th>Refer...</th>
<th>Index</th>
<th>Alter</th>
<th>Create...</th>
<th>Lock Ta...</th>
<th>Create...</th>
<th>Create...</th>
<th>After R...</th>
<th>Execu</th>
</tr>
</thead>
<tbody>
<tr>
<td>%</td>
<td>root2</td>
<td>&lt;global&gt;</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>localhost</td>
<td>mysql.infosch...</td>
<td>&lt;global&gt;</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>localhost</td>
<td>root</td>
<td>&lt;global&gt;</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>%</td>
<td>app_read</td>
<td>sakila</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>%</td>
<td>app_write</td>
<td>sakila</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
<tr>
<td>localhost</td>
<td>rw_norola_user</td>
<td>sakila</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>
MySQL Workbench Table Inspector - Grants
Permissions Reporting – Direct Grants

WITH

tableprivs AS (SELECT user, host, 'mysql.tables_priv' as PRIV_SOURCE, DB as _db, Table_Name as _obj, '' as _col
FROM mysql.tables_priv where Table_name like '%'),
colprivs AS (SELECT User, Host, 'mysql.columns_priv' as PRIV_SOURCE, DB as _db, table_name as _obj, column_name as _col
FROM mysql.columns_priv WHERE Table_name like '%')

SELECT user,host, PRIV_SOURCE, _db as _db, _obj, _col FROM
( SELECT user,host, PRIV_SOURCE, _db, _obj, _col FROM colprivs UNION
SELECT user,host, PRIV_SOURCE, _db, _obj, _col FROM tableprivs) as tt group by user, host, PRIV_SOURCE, _db, _obj, _col;

<table>
<thead>
<tr>
<th>user</th>
<th>host</th>
<th>PRIV_SOURCE</th>
<th>_db</th>
<th>_obj</th>
<th>_col</th>
</tr>
</thead>
<tbody>
<tr>
<td>newuserw</td>
<td>%</td>
<td>mysql.columns_priv</td>
<td>mysql</td>
<td>plugin</td>
<td>name</td>
</tr>
<tr>
<td>mysql.session</td>
<td>localhost</td>
<td>mysql.tables_priv</td>
<td>mysql</td>
<td>user</td>
<td></td>
</tr>
<tr>
<td>newuserq</td>
<td>%</td>
<td>mysql.tables_priv</td>
<td>mysql</td>
<td>plugin</td>
<td></td>
</tr>
<tr>
<td>newuserw</td>
<td>%</td>
<td>mysql.tables_priv</td>
<td>mysql</td>
<td>plugin</td>
<td></td>
</tr>
<tr>
<td>betsy</td>
<td>localhost</td>
<td>mysql.tables_priv</td>
<td>sakila</td>
<td>actor</td>
<td></td>
</tr>
<tr>
<td>mysql.sys</td>
<td>localhost</td>
<td>mysql.tables_priv</td>
<td>sys</td>
<td>sys_config</td>
<td></td>
</tr>
</tbody>
</table>
Which users / roles have access to actor

use mysql;
WITH
  globalprivs AS (SELECT user, host FROM mysql.user WHERE 'Y' IN
                 (Select_priv, Insert_priv, Update_priv, Delete_priv, Create_priv,
                  Drop_priv, Reload_priv, Shutdown_priv, Process_priv, File_priv,
                  Grant_priv, References_priv, Index_priv, Alter_priv, Show_db_priv,
                  Super_priv, Create_tmp_table_priv, Lock_tables_priv, Execute_priv,
                  Repl_slave_priv, Repl_client_priv, Create_view_priv, Show_view_priv,
                  Create_routine_priv, Alter_routine_priv, Create_user_priv,
                  Event_priv, Trigger_priv, Create_tablespace_priv, Create_role_priv,
                  Drop_role_priv)
                 ),
  dbprivs AS (SELECT user, host FROM mysql.db WHERE 'Y' IN
              (Select_priv, Insert_priv, Update_priv, Delete_priv, Create_priv, Drop_priv,
               Grant_priv, References_priv, Index_priv, Alter_priv, Create_tmp_table_priv,
               Lock_tables_priv, Create_view_priv, Show_view_priv, Create_routine_priv,
               Alter_routine_priv, Execute_priv, Event_priv, Trigger_priv)
              ),
  tableprivs AS (SELECT user, host FROM tables_priv WHERE Table_name='actor' ),
  colprivs AS (SELECT User, Host FROM mysql.columns_priv WHERE Table_name='actor' )
SELECT user, host FROM (SELECT user, host FROM globalprivs UNION
SELECT user, host FROM dbprivs UNION
SELECT user, host FROM tableprivs) as tt group by user, host;

Note:
There are various mysql.* users used by internal components
mysql.informationschema
mysql.session, mysql.sys
WITH
globalprivs AS (SELECT user,host FROM mysql.user WHERE
  Select_priv = 'Y'
),
dbprivs AS (SELECT user,host FROM mysql.db WHERE
  Select_priv = 'Y'
),
colprivs AS (SELECT user, host FROM mysql.columns_priv WHERE Table_name='actor'
  AND FIND_IN_SET('Select',Column_priv)),
tableprivs AS (SELECT User, Host FROM mysql.tables_priv WHERE Table_name='actor'
  AND FIND_IN_SET('Select',Table_priv))
SELECT user,host FROM (SELECT user,host FROM globalprivs UNION
SELECT user,host FROM dbprivs UNION
SELECT user,host FROM colprivs UNION
SELECT user,host FROM tableprivs) as tt group by user, host;
FOR table actor – roles who can modify data

WITH
globalpr"\vvs AS (SELECT user,host FROM mysql.user WHERE 'Y' IN (Insert_priv, Update_priv, Delete_priv, Drop_priv, Alter_priv)),
dbpr"\vvs AS (SELECT user,host FROM mysql.db WHERE 'Y' IN (Insert_priv, Update_priv, Delete_priv, Drop_priv, Alter_priv)),
tablepr"\vvs AS (SELECT user, host FROM tables_priv WHERE table_name='actor'),
colpr"\vvs AS (SELECT User, Host FROM mysql.columns_priv WHERE table_name='actor')
SELECT from_user, from_host FROM (SELECT user,host FROM globalpr"\vvs UNION SELECT user,host FROM dbpr"\vvs UNION SELECT user,host FROM colpr"\vvs UNION SELECT user,host FROM tablepr"\vvs) as tt
RIGHT JOIN
mysql.role_edges as tr
ON tr.to_user=tt.user AND tr.to_host= tt.host GROUP BY from_user, from_host;
Users with administrative/global permissions

```
SELECT user, host, 'Global Priv', Select_priv, Insert_priv, Update_priv, Delete_priv,
Create_priv, Drop_priv, Reload_priv, Shutdown_priv, Process_priv, File_priv,
Grant_priv, References_priv, Index_priv, Alter_priv, Show_db_priv,
Super_priv, Create_tmp_table_priv, Lock_tables_priv, Execute_priv,
Repl_slave_priv, Repl_client_priv, Create_view_priv, Show_view_priv,
Create_routine_priv, Alter_routine_priv, Create_user_priv,
Event_priv, Trigger_priv, Create_tablespace_priv, Create_role_priv,
Drop_role_priv FROM mysql.user
WHERE ('Y' IN
(SELECT_priv, Insert_priv, Update_priv, Delete_priv, Create_priv,
Drop_priv, Reload_priv, Shutdown_priv, Process_priv, File_priv,
Grant_priv, References_priv, Index_priv, Alter_priv, Show_db_priv,
Super_priv, Create_tmp_table_priv, Lock_tables_priv, Execute_priv,
Repl_slave_priv, Repl_client_priv, Create_view_priv, Show_view_priv,
Create_routine_priv, Alter_routine_priv, Create_user_priv,
Event_priv, Trigger_priv, Create_tablespace_priv, Create_role_priv,
Drop_role_priv)) and (user.user not like 'mysql.%');
```
Review MySQL Plugins – Install if missing or uninstall if unused

```
SELECT `PLUGINS`.'PLUGIN_NAME', `PLUGINS`.'PLUGIN_VERSION',
`PLUGINS`.'PLUGIN_STATUS', `PLUGINS`.'PLUGIN_TYPE',
`PLUGINS`.'PLUGIN_TYPE_VERSION', `PLUGINS`.'PLUGIN_LIBRARY',
`PLUGINS`.'PLUGIN_LIBRARY_VERSION', `PLUGINS`.'PLUGIN_DESCRIPTION',
`PLUGINS`.'PLUGIN_LICENSE', `PLUGINS`.'LOAD_OPTION'
FROM `information_schema`.'PLUGINS' where plugin_library is Not null;
```
Review User Ports

SELECT VARIABLE_NAME, VARIABLE_VALUE, 'If the defined port is deemed prohibited, this is a FAIL.' as Note
FROM performance_schema.global_variables
WHERE VARIABLE_NAME in ('port', 'mysqlx_port', 'admin_port');

MySQL Port Reference Tables

Check on where your files are stored

SELECT VARIABLE_NAME, VARIABLE_VALUE
FROM performance_schema.global_variables
WHERE (VARIABLE_NAME LIKE '%dir' or VARIABLE_NAME LIKE '%file')
and (VARIABLE_NAME NOT LIKE '%core%' AND VARIABLE_NAME <> 'local_infile'
AND VARIABLE_NAME <> 'relay_log_info_file') order by VARIABLE_NAME;

<table>
<thead>
<tr>
<th>VARIABLE_NAME</th>
<th>VARIABLE_VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>audit_log_file</td>
<td>audit.log</td>
</tr>
<tr>
<td>basedir</td>
<td>/usr/local/mysql-commercial-8.0.17-macos10.14-x86_64/</td>
</tr>
<tr>
<td>character_sets_dir</td>
<td>/usr/local/mysql-commercial-8.0.17-macos10.14-x86_64/share/charsets/</td>
</tr>
<tr>
<td>datadir</td>
<td>/usr/local/mysql/data/</td>
</tr>
<tr>
<td>ft_stopword_file</td>
<td>(built-in)</td>
</tr>
<tr>
<td>general_log_file</td>
<td>/usr/local/mysql/data/dhcp-10-154-137-61.log</td>
</tr>
<tr>
<td>init_file</td>
<td></td>
</tr>
<tr>
<td>innodb_data_home_dir</td>
<td>.</td>
</tr>
<tr>
<td>innodb_log_group_home_dir</td>
<td>#innodb_log_dir</td>
</tr>
<tr>
<td>innodb_tmpdir</td>
<td></td>
</tr>
<tr>
<td>lc_messages_dir</td>
<td>/usr/local/mysql-commercial-8.0.17-macos10.14-x86_64/share/</td>
</tr>
<tr>
<td>pid_file</td>
<td>/usr/local/mysql/data/mysql.sock/.pid</td>
</tr>
<tr>
<td>plugin_dir</td>
<td>/usr/local/mysql/plugin/</td>
</tr>
<tr>
<td>slave_log_file</td>
<td>/var/log/mysql/</td>
</tr>
<tr>
<td>slow_query_log_file</td>
<td>/usr/local/mysql/data/dhcp-10-154-137-61-slow.log</td>
</tr>
<tr>
<td>tmpdir</td>
<td>/var/log/</td>
</tr>
</tbody>
</table>
| validate_password.dictionary_file   | $FILENAME OF DICTIONARY FILE
Are your keys safe? Is keyring installed? Key manager?

```
SELECT `PLUGIN_NAME`, `PLUGIN_STATUS`, `PLUGIN_TYPE`, `PLUGIN_LIBRARY`, `PLUGIN_DESCRIPTION`, `LOAD_OPTION`
FROM `information_schema`.`PLUGINS` where PLUGIN_NAME LIKE 'keyring_file' and plugin_status='ACTIVE';
```

NOTE: keyring_file – is not for production. (Dev/QA only – its in a Plain text file)

KMIP, Encrypted Keyring, OCI Vault, Hashicorp, AWS KMS, etc. should be used in

<table>
<thead>
<tr>
<th>PLUGIN_NAME</th>
<th>PLUGIN_STATUS</th>
<th>PLUGIN_TYPE</th>
<th>PLUGIN_LIBRARY</th>
<th>PLUGIN_DESCRIPTION</th>
<th>LOAD_OPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>keyring_file</td>
<td>ACTIVE</td>
<td>KEYRING</td>
<td>keyring_file.so</td>
<td>store/fetch authentication data to/from a flat file</td>
<td>ON</td>
</tr>
</tbody>
</table>

NOTE: Keyring installation is key manager specific. See https://dev.mysql.com/doc/refman/8.0/en/keyring.html
AT REST Encryption Checks

InnoDB Tablespace Checks

```sql
SELECT `INNODB_TABLESPACES`.`NAME`, `INNODB_TABLESPACES`.`ENCRYPTION`, IF(ENCRYPTION = 'Y', 'PASS', 'FAIL') as CHECK_VAL
FROM `information_schema`.`INNODB_TABLESPACES` where ENCRYPTION='N';
```

REQUIRE INNODB TDE (Are tables required to be encrypted?)

```sql
SELECT VARIABLE_NAME, VARIABLE_VALUE, 'table_encryption_privilege_check - TABLE REQUIRE AT REST ENCRYPTION' as Note,
IF(VARIABLE_VALUE = 'ON', 'PASS', 'FAIL') as CHECK_VAL
FROM performance_schema.global_variables where variable_name = 'table_encryption_privilege_check';
```
InnoDB REDO, UNDO, Binlog, Audit log Encrypted?

SELECT VARIABLE_NAME, VARIABLE_VALUE, 'innodb_redo_log AT REST ENCRYPTION' as Note,
   IF(VARIABLE_VALUE = 'ON', 'PASS', 'FAIL')  as CHECK_VAL
FROM performance_schema.global_variables where variable_name = 'innodb_redo_log_encrypt';

SELECT VARIABLE_NAME, VARIABLE_VALUE, 'innodb_undo_log AT REST ENCRYPTION' as Note,
   IF(VARIABLE_VALUE = 'ON', 'PASS', 'FAIL')  as CHECK_VAL
FROM performance_schema.global_variables where variable_name = 'innodb_undo_log_encrypt';

SELECT VARIABLE_NAME, VARIABLE_VALUE, 'BINLOG - AT REST ENCRYPTION' as Note,
   IF(VARIABLE_VALUE = 'ON', 'PASS', 'FAIL')  as CHECK_VAL
FROM performance_schema.global_variables where variable_name = 'binlog_encryption';

SELECT VARIABLE_NAME, VARIABLE_VALUE, 'AUDIT LOG - AT REST ENCRYPTION' as Note,
   IF(VARIABLE_VALUE = 'AES', 'PASS', 'FAIL')
FROM performance_schema.global_variables where variable_name = 'audit_log_encryption';
Auditing Enabled?

Is the audit plugin loaded

SELECT `PLUGIN_NAME`, `PLUGIN_STATUS`, `PLUGIN_TYPE`, `PLUGIN_LIBRARY`,
`PLUGIN_DESCRIPTION`, `LOAD_OPTION` FROM `information_schema`.`PLUGINS` where PLUGIN_NAME LIKE 'audit_log' and plugin_status='ACTIVE';

If not loaded then run the installations script it will add the plugin and meta tables

# shell> mysql -u root -p < audit_log_filter_linux_install.sql;

# Edit the mysql config file my.cnf (or my.ini on windows)
set --audit-log to ON, FORCE, or FORCE_PLUS_PERMANENT.
Audit Rules, Auditing Who?

Rules in place (Log everything)

```
SELECT `audit_log_filter`.'NAME', `audit_log_filter`.'FILTER' FROM `mysql`.'audit_log_filter';
```

Adding a rule.

```
audit_log_filter_set_filter('log_all', '{ "filter": { "log": true } }');
```

NOTE: We have many rule templates. (20+) – which cover most needs. Simple rules may fill your disk or under audit. Rules let define selectivity.

Applied to Who

```
SELECT `audit_log_user`.'USER', `audit_log_user`.'HOST', `audit_log_user`.'FILTERNAME' FROM `mysql`.'audit_log_user';
```
With MySQL Shell you can bring checks together in a script. For example

```bash
dhcp-10-154-179-209:stig mfrank$ mysqlsh --py -uroot -p < securityexample.py
Check to see if you have installed MySQL Connection Controls - which deter attacks user passwords

<table>
<thead>
<tr>
<th>PLUGIN_NAME</th>
<th>PLUGIN_STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONNECTION_CONTROL</td>
<td>ACTIVE</td>
</tr>
<tr>
<td>CONNECTION_CONTROL_FAILED_LOGIN_ATTEMPTS</td>
<td>ACTIVE</td>
</tr>
</tbody>
</table>

GOOD - Connection controls are in place.
Showing top 50 failed login counts

<table>
<thead>
<tr>
<th>USERHOST</th>
<th>FAILED_ATTEMPTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>'auditme'@'localhost'</td>
<td>2</td>
</tr>
<tr>
<td>'mybackuser'@'localhost'</td>
<td>2</td>
</tr>
<tr>
<td>'app_developer'@'%'</td>
<td>1</td>
</tr>
</tbody>
</table>
```
The code for this is simple python

dhcp-10-154-179-209:stig mfrank$ cat securityexample.py

shell.connect("mysql://root2:Oracle22@localhost")

session = shell.get_session()

print("Check to see if you have installed MySQL Connection Controls - which deter attacks on user passwords")
r = session.run_sql("SELECT PLUGIN_NAME, PLUGIN_STATUS FROM INFORMATION_SCHEMA.PLUGINS"
                     " WHERE PLUGIN_NAME LIKE 'connection%' ")

if shell.dump_rows(r) > 0:
    print('GOOD - Connection controls are in place.')
    print("Showing top 50 failed login counts")
r = session.run_sql("select * from information_schema.CONNECTION_CONTROL_FAILED_LOGIN_ATTEMPTS"
                    " order by failed_attempts desc limit 50")
if shell.dump_rows(r) > 0:
    print(' ')  
else:
    print("No failed logins found")
else:
    print("RECOMMENDATION - ENABLE CONNECTION CONTROL PLUGINS")
print("INSTALL PLUGIN CONNECTION_CONTROL SONAME 'connection_control.so';")
print("INSTALL PLUGIN CONNECTION_CONTROL_FAILED_LOGIN_ATTEMPTS SONAME 'connection_control.so';")
print("OR add to my.cnf to force at startup")
print("[mysqld]"
     "plugin-load-add=connection_control.so "
     "connection-control=FORCE_PLUS_PERMANENT "
     "connection-control-failed-login-attempts=FORCE_PLUS_PERMANENT")
print("Change from defaults values if desired - for example")
print("SET PERSIST connection_control_failed_connections_threshold = 4;")
print("SET PERSIST connection_control_min_connection_delay = 1500;")
Upgrade, Upgrade, Upgrade

Stay up to date

New LTS model makes upgrades far simpler.
MySQL Security Architecture
MySQL Enterprise Edition - SECURITY

MySQL Enterprise **TDE**
- Data-at-Rest Encryption
- Key Management/Security
- KMIP, Hashicorp, OCI Vault

MySQL Enterprise **Authentication**
- External Authentication Modules
  - Microsoft AD, Linux PAMs, LDAP, Native Kerberos

MySQL Enterprise **Encryption**
- Public/Private Key Cryptography
- Asymmetric Encryption
- Digital Signatures, Data Validation
- User Activity Auditing, Regulatory Compliance

MySQL **Data Masking**

- **MySQL Enterprise Firewall**
  - Block SQL Injection Attacks
  - Intrusion Detection

- **MySQL Enterprise Audit**
  - User Activity Auditing, Regulatory Compliance

- **MySQL Enterprise Monitor**
  - Changes in Database Configurations, Users Permissions, Database Schema, Passwords

- **MySQL Enterprise Backup**
  - Securing Backups, AES 256 encryption

- **MySQL Enterprise Thread pool**
  - Attack Hardening
Enterprise Security Architecture

- **Network Encryption**
  - Strong Authentication

- **Firewall**

- **Access Controls**

- **Thread Pool**
  - Attack minimization

- **Audit Vault**

- **Enterprise Audit**
  - Powerful Rules Engine

- **Enterprise Backup**
  - Encrypted

- **Key Vault**
  - Protect Keys (KMIP, Rest APIs)

- **Data Encryption**
  - TDE
  - Encryption
  - PKI

- **Enterprise Authentication**
  - SSO - LDAP, AD, PAM, Native Kerberos, FIDO2, WebAuthn, MFA

- **Enterprise Masking & De-Identification**
  - Masking
  - Substitute/Subset
  - Random Formatted Data
  - Blacklisted Data

- **Assess**
- **Prevent**
- **Detect**
- **Recover**

- **HA**
  - Innodb Cluster
Authentication

Plenty of options
- LDAP SASL/Kerberos
- Native Kerberos (8.0.26)
- MDS - OCI IAM / Identity Domains (8.0.26)
- Passwordless (FIDO U2F)
  - For example Yubikey’s
- MFA (up to 3 factors) (8.0.28)
- FIDO2 WebAuthn (8.2)

Additional Authentication methods are in the works.
MySQL Enterprise Authentication Options

- **Auth Methods (1 to 3)**
  - Challenge/Response
  - LDAP/Active Directory
  - X.509
  - Fido2/WebAuthn
  - Kerberos

- **Multi-Factor – up to 3**
  - Username/Password
  - Certificate
  - FIDO2/U2F, WebAuthn
  - Ticket
  - kinit – request

- **Multi-Factor**
  - User/Pass
  - SASL
  - GSSAPI / Kerberos
  - SSPI / Kerberos

- **KDC (Key Distribution Center)**

- **LDAP or Active Directory**

- **CA Certificates**
At Rest Encryption / TDE / Keyring

Already

- Complete at-rest
  - InnoDB, Redo/Undo, Binlogs, Audit Data

- Secure storage for sensitive system variables (8.0.29)
  - Extension to existing server configuration settings - determines how the SET PERSIST code will handle the backend storage of these settings.
  - If a server variable is marked as sensitive, instead of going an OS file, it will be stored in a keyring using the keyring API.

- Support for more many KMIP failover server IPs (8.0.29)
  - Currently 2. Expanding for up to 64. (9 was requested)
Firewall

What’s added (8.0.26)

• Named allow list sets
  • Turns the 1-to-1 between user accounts and Allow List rules into many-to-many
    • Named group profiles can be created.
    • A group profile can include multiple accounts as members
    • An account can be a member of multiple group profiles.
  • Define named Allow lists and then assign them to user accounts
Audit

What’s Added

**ANALYZE TABLE** statements now produce read audit events

Audit log connect events include any connection attributes passed by the client.

**audit_api_message_emit** component - enables applications to add their own message events to the audit log

- **audit_api_message_emit_udf()** user-defined function.
- See **The Audit Message Component**.
- Audit Log event function reading starting from specified date/time
- Remove and groom audit data by time and size.

https://mysqlserverteam.com/auditing-changes-to-classified-data-stored-in-mysql-8-0/
Audit (cont.)

What’s Added

- **Scrub sensitive data in the audit log**
- **Epoch time format** – (like linux) for simplification of audit data consolation
- **Audit log grooming** – by age and/or size
  - DBAs that can’t get on the OS to remove audit data
- Global Stop/Start (8.0.28)
- Add performance metrics to audit logs
- Custom Schema – allows user to define – use for Replication Filters
  - You can replicate audit filters or now.
  - Can switch filters using schemas as templates - change and flush.
Audit Log Performance Statistics

Within the filter rule you can add metrics

For example

```sql
SELECT audit_log_filter_set_filter('QueryStatistics',
    '{ "filter": { "class": { "name": "general", "event": { "name": "status", "print": 
        { "service": { "implementation": "mysql_server", "tag": "query_statistics", "element": [ 
            { "name": "query_time", "type": "double" },
            { "name": "bytes_sent", "type": "longlong" },
            { "name": "bytes_received", "type": "longlong" },
            { "name": "rows_sent", "type": "longlong" },
            { "name": "rows_examined", "type": "longlong" } ] } } } } }');
```

What’s New

• Reload TLS certificate online

• Support for TLS 1.3 - `tls_ciphersuites` system variable enables explicitly specifying which TLSv1.3 ciphersuites the server permits.

• TLSv1 and TLSv1.1 connection protocols now are deprecated and support for them is subject to removal in a future MySQL version.

• On platforms on which OpenSSL libraries are bundled
  • The linked OpenSSL library for MySQL Server has been updated to version 1.1.1k.
  • Issues fixed in the new OpenSSL version are described at
    • https://www.openssl.org/news/cl111.txt and
    • https://www.openssl.org/news/vulnerabilities.html
What’s New

Router

• Connection multiplexing and TLS Endpoint
  • Moves connection creation and TLS/SSL overhead from the Server to the Router
• Accept connections only if destinations are available

What’s next

• Support for OpenSSL and FIPS Object Model
MySQL Enterprise **Masking and De-Identification**

Data De-identification helps database customers improve security

Accelerates compliance for

- Government – GDPR, CHHS
- Financial - PCI
- Healthcare – HIPAA, Clinic Trials Data

Reduce IT costs by simplifying sanitizing production data

- Transforming sensitive data for use in analytics, testing, development, and more
MySQL Enterprise **Masking and De-Identification**

Data Masking and Random Data Generation

Data Masking
- String masking
- Dictionary based replacement
- Specific masking
  - SSN
  - Payment card: Strict/Relaxed

Random Data Generators
- Random number within a range
- Email
- Payment card (Luhn check compliant)
- SSN
- Dictionary based generation
MySQL Security Guidelines

Recommendations from us

Department of Defense (DoD) approved and published Security Technical Implementation Guide (STIG)

- DISA STIG for MySQL 8.0 EE

https://www.mysql.com/products/enterprise/stig.html

https://public.cyber.mil/stigs/
Center For Internet Security Benchmark

CIS Benchmark for MySQL 8.0 EE

• https://www.cisecurity.org/benchmark/oracle_mysql/
Resources

MySQL Secure Deployment Guide


60+ blogs to dive into specific topics and features

• https://blogs.oracle.com/mysql/search.html?contentType=Blog-Post&default=security*
• https://dev.mysql.com/blog-archive/?cat=Security

Whitepapers

• https://www.mysql.com/why-mysql/white-papers/#en-22-40

On Demand Webinars


Forums

• https://forums.mysql.com/
Tell us – with emails, requirements documents

New features you want

Where are your pains

What strategies do you want to see longer term

If I can get requests in emails – mike.frank@oracle.com
   - requirement, use case, time frame, etc.
MySQL Summit 2024

Wednesday, May 1st
Oracle Conference Center, Redwood Shores, California

- Generative AI and Vector Store
- Machine Learning
- Lakehouse and Analytics
- Performance Tuning Tips and Tricks
- High Availability and Disaster Recovery
- And many more popular topics

Register for this free event
https://www.oracle.com/events/mysql-summit/redwood-shores/