Enhancing MySQL Security

Vinicius M. Grippa
Senior Support Engineer for MySQL/MongoDB
vinicius.grippa@percona.com
About Me

• Support Engineer at Percona since 2017.
• Working with MySQL for over six years.
• Working with databases for over nine years.
• Speaker at PL 2018 and meetups about MySQL/MongoDB.
Basic Principles

• Minimum access
• Isolate
• Audit
• Avoid spying
• Default firewall
Agenda

• SO/Cloud security
• SSL
• Password management
• Audit plugin
• Percona Server encryption features
• MySQL 8 features (undo, redo encryption)
• TDE
• New caching_sha2_password
• FIPS mode
• Roles
OS/Cloud Security
OS/Cloud Security

• Uninstall services that are not used
• Do not run compilers
• Firewalls
• Block internet access
• Disable remote root login
• Use of SSH Key
OS/Cloud Security

- Use of Amazon Virtual Private Cloud (VPC)
- Use AWS Identity and Access Management (IAM) policies
- Use security groups
Step 1: Select a VPC Configuration

**VPC with a Single Public Subnet**

**VPC with Public and Private Subnets**

In addition to containing a public subnet, this configuration adds a private subnet whose instances are not addressable from the Internet. Instances in the private subnet can establish outbound connections to the Internet via the public subnet using Network Address Translation (NAT).

**Creates:**

A /16 network with two /24 subnets. Public subnet instances use Elastic IPs to access the Internet. Private subnet instances access the Internet via Network Address Translation (NAT). (Hourly charges for NAT devices apply.)

**Select**
# OS/Cloud Security

## IAM Resources

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Users</td>
<td>0</td>
</tr>
<tr>
<td>Groups</td>
<td>0</td>
</tr>
<tr>
<td>Roles</td>
<td>4</td>
</tr>
<tr>
<td>Identity Providers</td>
<td>0</td>
</tr>
<tr>
<td>Customer Managed Policies</td>
<td>0</td>
</tr>
</tbody>
</table>

## Security Status

- [x] Delete your root access keys
- [x] Activate MFA on your root account
- [ ] Create individual IAM users
- [ ] Use groups to assign permissions
- [ ] Apply an IAM password policy

2 out of 5 complete.
## OS/Cloud Security

Security Group: sg-0039ef80773d62cd9

<table>
<thead>
<tr>
<th>Type</th>
<th>Protocol</th>
<th>Port Range</th>
<th>Source</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MYSQL/Aurora</td>
<td>TCP</td>
<td>3306</td>
<td>0.0.0.0/0</td>
<td></td>
</tr>
<tr>
<td>MYSQL/Aurora</td>
<td>TCP</td>
<td>3306</td>
<td>::/0</td>
<td></td>
</tr>
</tbody>
</table>
SSL

- Move information over a network in a secure fashion
- SSL provides a way to cryptograph the data
- Default for MySQL 5.7 or higher
- Certificates
  - MySQL 5.7
    - mysql_ssl_rsa_setup
  - MySQL 5.6
    - openssl
SSL

mysql > show global variables like '%ssl%';
+---------------------------------+-----------------+
| Variable_name            | Value           |
+---------------------------------+-----------------+
| have_openssl             | YES             |
| have_ssl                 | YES             |
| ssl_ca                   | ca.pem          |
| ssl_capath               |                 |
| ssl_cert                 | server-cert.pem |
| ssl_cipher               |                 |
| ssl_crl                  |                 |
| ssl_crlpath              |                 |
| ssl_key                  | server-key.pem  |
+---------------------------------+-----------------+
9 rows in set (0.03 sec)
mysql: root@localhost ((none)) GRANT ALL PRIVILEGES ON *.* TO 'ssluser'@'%' IDENTIFIED BY 'sekret' REQUIRE SSL;

Query OK, 0 rows affected, 1 warning (0.00 sec) Query OK, 0 rows affected (0.01 sec)

[root@node1 ~]# mysql -ussluser -psekret
--ssl-cert=/var/lib/mysql/client-cert.pem
--ssl-key=/var/lib/mysql/client-key.pem --ssl-ca=/var/lib/mysql/ca.pem -h 127.0.0.1 -P 3306 -e "\s"| grep SSL

mysql: [Warning] Using a password on the command line interface can be insecure.

SSL: Cipher in use is ECDHE-RSA-AES128-GCM-SHA256
It is also possible to set `ssl-mode` to ensure that all connections use SSL. This option is available only for client programs, not the server.

[client]
ssl-mode=required
SSL

Sysbench Read-Only - Throughput

Queries Per Second

Threads

No SSL
1024-bit SSL
2048-bit SSL
4096-bit SSL
Password Management
Password Management

- Password expiration
- validate_password plugin
Password Expiration

• MySQL enables database administrators to expire account passwords manually, and to establish a policy for automatic password expiration. Expiration policy can be established globally, and individual accounts can be set to either defer to the global policy or override the global policy with specific per-account behavior.
Password Expiration

Individual Accounts

mysql> create user test_expired_user@localhost identified by 'Sekr$K1et' PASSWORD EXPIRE INTERVAL 1 day;
Query OK, 0 rows affected (0.01 sec)

Globally

mysql> SET GLOBAL default_password_lifetime = 1;
Password Expiration

mysql: test_expired_user@localhost ((none)) > show databases;
ERROR 1820 (HY000): You must reset your password using ALTER USER statement before executing this statement.
validate_plugin

Its main purpose is to test passwords and improve security. It is possible to ensure the strength, length and required characters of the password.
validate_plugin - Installing

# Runtime
mysql: root@localhost ((none)) > INSTALL PLUGIN validate_password SONAME 'validate_password.so';
Query OK, 0 rows affected (0.07 sec)

# my.cnf
[mysqld]
plugin-load-add=validate_password.so
validate_plugin - Validate

mysql: root@localhost ((none)) > show global variables like '%plugin%';
+-------------------------------------------------+
| Variable_name         | Value                        |
+-------------------------------------------------+
| default_authentication_plugin | mysql_native_password |
| plugin_dir               | /usr/lib64/mysql/plugin/ |
+-------------------------------------------------+
2 rows in set (0.00 sec)

mysql: root@localhost ((none)) > SELECT PLUGIN_NAME, PLUGIN_STATUS
    FROM INFORMATION_SCHEMA.PLUGINS
    WHERE PLUGIN_NAME LIKE 'validate%';
+-------------------------------+---------------+
| PLUGIN_NAME       | PLUGIN_STATUS |
+-------------------------------+---------------+
| validate_password | ACTIVE        |
+-------------------------------+---------------+
1 row in set (0.00 sec)
validate_plugin - Example

mysql: root@localhost ((none)) > set global validate_password_length = 6;
Query OK, 0 rows affected (0.00 sec)

mysql: root@localhost ((none)) > set global validate_password_policy=2;
Query OK, 0 rows affected (0.00 sec)
validate_plugin - Example

```
mysql: root@localhost ((none)) > create user
test_password@localhost identified by 'PasSw0Rd';
ERROR 1819 (HY000): Your password does not satisfy the
current policy requirements

mysql: root@localhost ((none)) > create user
test_password@localhost identified by 'PasSw0Rd12@';
Query OK, 0 rows affected (0.00 sec)
```
Audit Plugin
Audit Plugin

- MySQL Enterprise – Paid
- Percona Server (works with community version) – Free
- It is different from general log
- Filter by command / user / database
Audit Plugin - Installing

mysql > INSTALL PLUGIN audit_log SONAME 'audit_log.so';
Query OK, 0 rows affected (0.05 sec)

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

+-------------+---------------+
| PLUGIN_NAME | PLUGIN_STATUS |
+-------------+---------------+
| audit_log   | ACTIVE        |
+-------------+---------------+
1 row in set (0.00 sec)
Audit Plugin

[mysqld]
## Audit Logging ##
audit_log_policy=ALL
audit_log_format=JSON
audit_log_file=/var/log/mysql/audit.log
audit_log_rotate_on_size=1024M
audit_log_rotations=10
## Audit Plugin

```
mysql: root@localhost ((none)) > show global variables like 'audit%';
+----------------------------------------------------------+
| Variable_name                                | Value           |
+----------------------------------------------------------+
| audit_log_buffer_size                        | 1048576         |
| audit_log_exclude_accounts                   |                 |
| audit_log_exclude_commands                   |                 |
| audit_log_exclude_databases                  |                 |
| audit_log_file                               | /var/log/mysql/audit.log |
| audit_log_flush                             | OFF             |
| audit_log_format                            | JSON            |
| audit_log_handler                           | FILE            |
| audit_log_include_accounts                   |                 |
| audit_log_include_commands                   |                 |
| audit_log_include_databases                  |                 |
```
## Audit Plugin

Here are the global variables related to audit configuration:

```sql
global variables like 'audit%';
```

<table>
<thead>
<tr>
<th>Variable Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>audit_log_policy</td>
<td>ALL</td>
</tr>
<tr>
<td>audit_log_rotate_on_size</td>
<td>1073741824</td>
</tr>
<tr>
<td>audit_log_rotations</td>
<td>10</td>
</tr>
<tr>
<td>audit_log_strategy</td>
<td>ASYNCHRONOUS</td>
</tr>
<tr>
<td>audit_log_syslog_facility</td>
<td>LOG_USER</td>
</tr>
<tr>
<td>audit_log_syslog_ident</td>
<td>percona-audit</td>
</tr>
<tr>
<td>audit_log_syslog_priority</td>
<td>LOG_INFO</td>
</tr>
</tbody>
</table>

There are 18 rows in the set, processed in 0.02 seconds.
Percona Server Encryption Features
Percona Server Encryption

Percona server provides extra encryption:

• encrypt_binlog
• encrypt_tmp_files
• innodb_encrypt_online_alter_logs
• innodb_encrypt_tables – BETA quality
• innodb_parallel_dblwr_encrypt – ALPHA quality
• innodb_sys_tablespace_encrypt – ALPHA quality
• innodb_temp_tablespace_encrypt – BETA quality
Percona Server Encryption

[mysqld]
# Binary Log Encryption
encrypt_binlog
master_verify_checksum = 1
binlog_checksum = 1

mysql: root@localhost ((none)) > show global variables like '%encrypt_binlog%';
+----------------+-------+
| Variable_name   | Value |
+----------------+-------+
| encrypt_binlog  | ON    |
+----------------+-------+
1 row in set (0.00 sec)
### Percona Server Encryption

```sql
mysql: root@localhost ((none)) > show global variables like '%encrypt%';
```

<table>
<thead>
<tr>
<th>Variable_name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>block_encryption_mode</td>
<td>aes-128-ecb</td>
</tr>
<tr>
<td>encrypt_binlog</td>
<td>ON</td>
</tr>
<tr>
<td>encrypt_tmp_files</td>
<td>OFF</td>
</tr>
<tr>
<td>innodb_encrypt_onlineAlterLogs</td>
<td>OFF</td>
</tr>
<tr>
<td>innodb_encrypt_tables</td>
<td>OFF</td>
</tr>
<tr>
<td>innodb_parallel_dblwr_encrypt</td>
<td>OFF</td>
</tr>
<tr>
<td>innodb_sys_tablespace_encrypt</td>
<td>OFF</td>
</tr>
<tr>
<td>innodb_temp_tablespace_encrypt</td>
<td>OFF</td>
</tr>
</tbody>
</table>

8 rows in set (0.00 sec)
MySQL 8 Features
(Undo, Redo Encryption)
MySQL 8 - (Undo, Redo Encryption)

- MySQL 8 extends tablespace encryption feature to redo log and undo log.
- It is necessary using one of the Keyring plugins.
MySQL 8 - (Undo, Redo Encryption)

The process is very straightforward, to enable the encryption on the redo log and the undo log:

```sql
mysql> set global innodb_undo_log_encrypt = 1;
Query OK, 0 rows affected (0.00 sec)

mysql> set global innodb_redo_log_encrypt = 1;
Query OK, 0 rows affected (0.00 sec)

mysql> show global variables like '%log_encrypt%';
+-------------------------+-------+
| Variable_name           | Value |
+-------------------------+-------+
| innodb_redo_log_encrypt | ON    |
| innodb_undo_log_encrypt | ON    |
+-------------------------+-------+
2 rows in set (0.00 sec)
```
Transparent Data Encryption (TDE)
Transparent Data Encryption (TDE)

- Enables data-at-rest encryption in the database.
- Encryption and decryption occurs without any additional coding, data type or schema modifications.
Transparent Data Encryption (TDE)

```
[mysqld]
# TDE
early-plugin-load=keyring_file.so
keyring-file-data=/var/lib/mysql-keyring/keyring

mysql: root@localhost ((none)) > INSTALL PLUGIN keyring_udf SONAME 'keyring_udf.so';
Query OK, 0 rows affected (0.00 sec)

mysql: root@localhost ((none)) > SELECT PLUGIN_NAME, PLUGIN_STATUS
    FROM INFORMATION_SCHEMA.PLUGINS
    WHERE PLUGIN_NAME LIKE 'keyring%';
```

+--------------+---------------+
<table>
<thead>
<tr>
<th>PLUGIN_NAME</th>
<th>PLUGIN_STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>keyring_file</td>
<td>ACTIVE</td>
</tr>
<tr>
<td>keyring_udf</td>
<td>ACTIVE</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------</td>
</tr>
</tbody>
</table>
Transparent Data Encryption (TDE)

mysql: root@localhost ((none)) > SELECT keyring_key_generate('MyKey', 'AES', 32);
+------------------------------------------+
| keyring_key_generate('MyKey', 'AES', 32) |
+------------------------------------------+
| 1 |
+------------------------------------------+
1 row in set (0.00 sec)

mysql> CREATE TABLESPACE `amer_meeting1` ADD DATAFILE 'amer_meeting1.ibd' ENCRYPTION = 'Y' Engine=InnoDB;
Query OK, 0 rows affected (0.01 sec)

mysql: root@localhost (test) > CREATE TABLE t1 (a INT, b TEXT) TABLESPACE vgrippa ENCRYPTION='N';
ERROR 1478 (HY000): InnoDB: Tablespace `vgrippa` can contain only an ENCRYPTED tables.

mysql: root@localhost (test) > CREATE TABLE t1 (a INT, b TEXT) TABLESPACE vgrippa ENCRYPTION='Y';
Query OK, 0 rows affected (0.02 sec)
Transparent Data Encryption (TDE)

A flag field in the INFORMATION_SCHEMA.INNODB_SYS_TABLESPACES has bit number 13 set if tablespace is encrypted.

```sql
mysql: root@localhost (test) > SELECT space, name, flag, 
(flag & 8192) != 0 AS encrypted FROM 
INFORMATION_SCHEMA.INNODB_SYS_TABLESPACES WHERE name in ('vgrippa');
```

<table>
<thead>
<tr>
<th>space</th>
<th>name</th>
<th>flag</th>
<th>encrypted</th>
</tr>
</thead>
<tbody>
<tr>
<td>156</td>
<td>vgrippa</td>
<td>10240</td>
<td>1</td>
</tr>
</tbody>
</table>

1 row in set (0.00 sec)
caching_sha2_password
caching_sha2_password

MySQL provides two authentication plugins that implement SHA-256 hashing for user account passwords:

- **sha256_password**: Implements basic SHA-256 authentication
- **caching_sha2_password**: Implements SHA-256 authentication (like sha256_password), but uses caching on the server side for better performance and has additional features for wider applicability. (In MySQL 5.7, caching_sha2_password is implemented only on the client)

Note: In MySQL 8.0, caching_sha2_password is the default authentication plugin rather than mysql_native_password.
mysql: root@localhost ((none)) > grant all privileges on *.* to vgrippa@localhost identified by 'teste';
Query OK, 0 rows affected, 1 warning (0.00 sec)

mysql: root@localhost ((none)) > grant all privileges on *.* to vgrippa1@localhost identified by 'teste';
Query OK, 0 rows affected, 1 warning (0.00 sec)
### caching_sha2_password

```sql
mysql: root@localhost ((none)) > select user, host, plugin, authentication_string from mysql.user where user like 'vgrippa%';
```
```
+----------+-----------+-----------------------+--------------------------+
| user     | host      | plugin                | authentication_string    |
|----------|-----------+-----------------------+--------------------------+
| vgrippa  | localhost | mysql_native_password | *A00D6EEF76EC509DB66358D2E6685F8FF7A4C3DD |
| vgrippa1 | localhost | mysql_native_password | *A00D6EEF76EC509DB66358D2E6685F8FF7A4C3DD |
+----------+-----------+-----------------------+--------------------------+
2 rows in set (0.00 sec)
```
Example

# MySQL 8
[mysqld]
default_authentication_plugin=caching_sha2_password

mysql> CREATE USER 'sha2user'@'localhost' IDENTIFIED WITH caching_sha2_password BY 'password';
Query OK, 0 rows affected (0.06 sec)

mysql> select user, host, plugin from mysql.user where user like 'sha2user%';

+----------+-----------+-----------------------+
| user     | host      | plugin                |
|----------+-----------+-----------------------|
| sha2user | localhost | caching_sha2_password |
+----------+-----------+-----------------------+
1 row in set (0.00 sec)
Example

mysql: root@localhost ((none)) > create user vgrippa@localhost identified by 'teste';
Query OK, 0 rows affected (0.01 sec)

mysql: root@localhost ((none)) > create user vgrippa1@localhost identified by 'teste';
Query OK, 0 rows affected (0.01 sec)
Example

mysql: root@localhost ((none)) > select user, host, plugin, authentication_string from mysql.user where user like 'vgrippa%';

+----------+-----------+-----------------------+-----------------------------------+
| user     | host      | plugin                | authentication_string             |
+----------+-----------+-----------------------+-----------------------------------+
| vgrippa  | localhost | caching_sha2_password | $A$005$)8?=V_"J75FFqjUVMUZmnZ1t8aSybB4AISoj1MXdlse1UrQay6bGG1ne8 |  
| vgrippa1 | localhost | caching_sha2_password | $A$005$zEZ;bEmj[hq1T"!LftqZzAB0hacxgwNfhM/gl6gBFHqY1wuozW2N04Gj9958 |  
+----------+-----------+-----------------------+-----------------------------------+
2 rows in set (0.01 sec)
FIPS Mode
FIPS

- MySQL supports FIPS mode, if compiled using OpenSSL, and an OpenSSL library and FIPS Object Module are available at runtime.

- FIPS mode on the server side applies to cryptographic operations performed by the server. This includes replication (master/slave and Group Replication) and X Plugin, which run within the server. FIPS mode also applies to attempts by clients to connect to the server.
Example

mysql> show global variables like '%fips%';
+---------------+-------+
| Variable_name | Value |
+---------------+-------+
| ssl_fips_mode | ON    |
+---------------+-------+
1 row in set (0.01 sec)

mysql> set global ssl_fips_mode=1;
Query OK, 0 rows affected (0.06 sec)
Example

mysql> select md5('a');
+----------------------------------+
| md5('a')                         |
+----------------------------------+
| 00000000000000000000000000000000 |
+----------------------------------+
1 row in set, 1 warning (0.00 sec)
mysql> show warnings;
+--------------------------------------------------------+
| Level   | Code  | Message                     |
|---------+-------+------------------------------------------------------------------------|
+---------+-------+------------------------------------------------------------------------|
| Warning | 11272 | SSL fips mode error: FIPS mode ON/STRICT: MD5 digest is not supported. |
+---------+-------+------------------------------------------------------------------------|
1 row in set (0.00 sec)
Example

```sql
mysql> select sha2('a', 256);
+------------------------------------------------------------------+
| sha2('a', 256)                                                   |
| ca978112ca1bbdcafac231b39a23dc4da786eff8147c4e72b9807785afee48bb |
+------------------------------------------------------------------+
1 row in set (0.00 sec)
```
Roles
Roles

- MySQL 8 comes with Roles feature. A role is a named collection of privileges. Like user accounts, roles can have privileges granted to and revoked from them.
Roles

mysql> create role app_read;
Query OK, 0 rows affected (0.03 sec)

mysql> grant select on *.* to app_read;
Query OK, 0 rows affected (0.04 sec)
Roles

mysql> select * from app_db.joinit;
ERROR 1142 (42000): SELECT command denied to user 'test_role'@'localhost' for table 'joinit'

mysql> SELECT CURRENT_ROLE();
+----------------+
| CURRENT_ROLE() |
+----------------+
| NONE           |
+----------------+
1 row in set (0.00 sec)
Roles

mysql> SET ROLE all;
Query OK, 0 rows affected (0.00 sec)

mysql> SELECT CURRENT_ROLE();
+-----------------------------------------------+
| CURRENT_ROLE()                                |
+-----------------------------------------------+
| `app_read`@`%`, `app_write`@`%`, `app_read`@`localhost` |
+-----------------------------------------------+
1 row in set (0.00 sec)

mysql> select * from app_db.joinit;
Roles

It is possible to use `activate_all_roles_on_login` to activate all roles granted to each account at login time.
# SO/Cloud security

# Audit log
https://www.percona.com/blog/2015/09/10/percona-server-audit-log-plugin-best-practices/

#caching_sha2_password

# SSL
https://www.percona.com/blog/2013/06/22/setting-up-mysql-ssl-and-secure-connections/#setup
https://www.percona.com/blog/2013/10/10/mysql-ssl-performance-overhead/

# TDE
https://www.percona.com/doc/percona-server/LATEST/management/data_at_rest_encryption.html#usage

# Roles
https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html#sysvar_activate_all_roles_on_login

# Password management

# FIPS

# Percona Server 8.0 Alpha release

# MySQL 8 redo and undo encryption
Any Questions?
Rate My Session

Accelerating Application Development with Amazon Aurora

Description
In this hands-on tutorial, you'll learn how to leverage the unique features of Amazon Aurora to build faster, more scalable database applications optimized for the cloud. We discuss architectural best practices and features designed to help you develop applications faster and reach the widest possible audience, including Aurora Serverless, read replica auto scaling, cross-region replicas, backtracking, fast database cloning, and Performance Insights. You'll understand how to best take advantage of the Aurora platform's capabilities to effectively accelerate application development.
We’re Hiring!

Percona’s open source database experts are true superheroes, improving database performance for customers across the globe.

Our staff live in nearly 30 different countries around the world, and most work remotely from home.

Discover what it means to have a Percona career with the smartest people in the database performance industries, solving the most challenging problems our customers come across.