Thank you to our sponsors - Readyset, ProxySQL, PDB Monitor, Percona, Alura, FIAP

Improving your MySQL Security Posture

Ronald Bradford
Oct 2025

https://ronaldbradford.com/presentations/

me@ronaldbradford.com

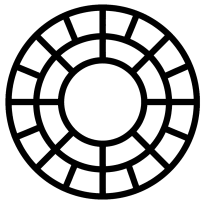
Agenda

- Why is security important
- Rank your MySQL security now
- Five proactive steps to improve your security posture
- What happens when your exposed

About Me

Author | Speaker | Contributor

- 1999 - Started using MySQL (pre 3.23) (26/30)
- 2006 - First MySQL Conference Presentation (19/30)
- 2006 - 2008 Worked at MySQL Inc
- 2010 - 2017 Oracle ACE Director (Alumni)

- 2025 - Database Architecture Director  **VSCO**

What is a Security Posture?

An organization's overall cybersecurity health and readiness to prevent, detect, respond to, and recover from cyber threats.

"Your database security is the final barrier"

Why is this important?

It is not a shiny new product.

It is not a glamorous job.

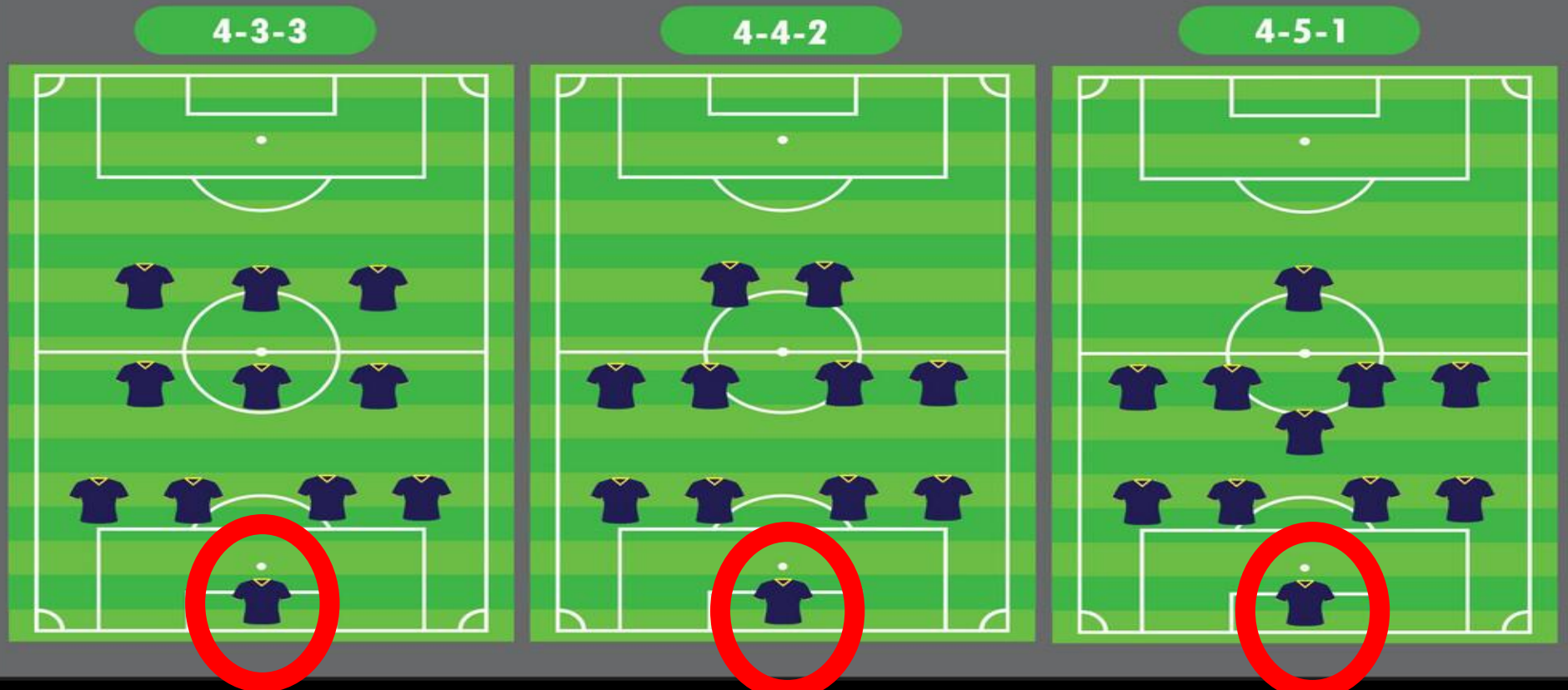
But in reality.

If your business was hacked,

and your data was stolen, **would you have a job?**

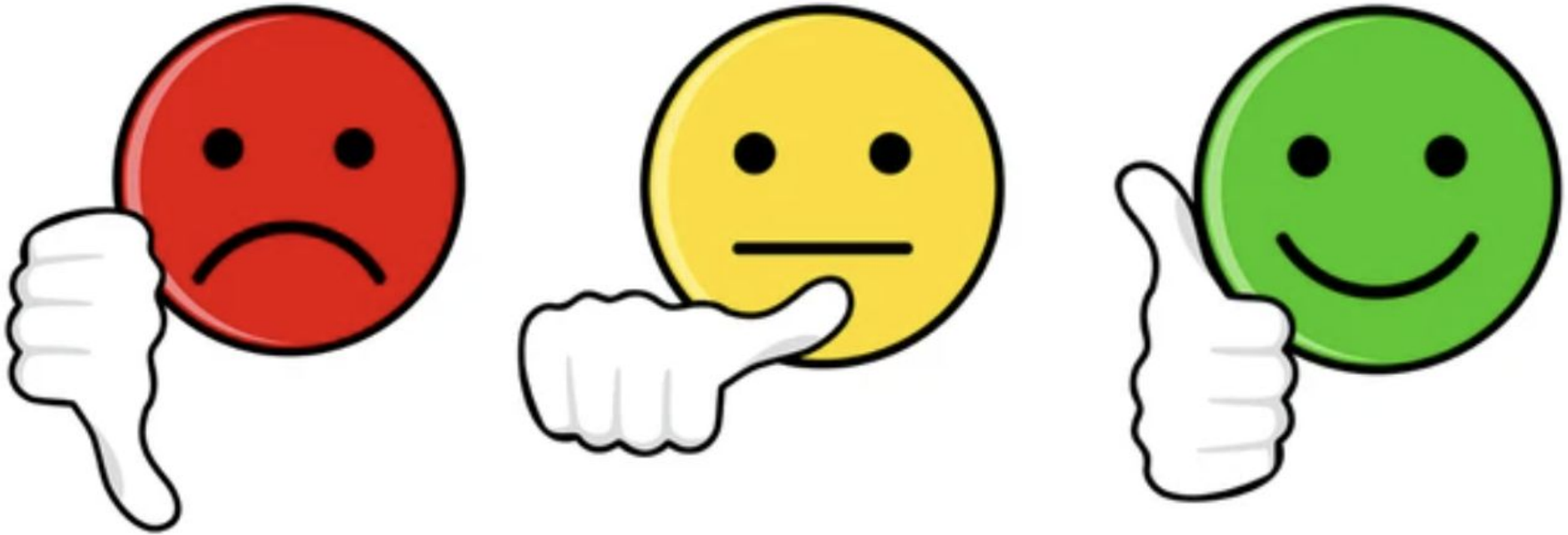
TACTICAL FORMATION

TACTICAL FORMATION




Database security is the last defense

Rate your database security health?



Self Assessment Checks

- Accounts with no password
- Accounts with easy password (rainbow)
- Accounts with insecure authentication
- Accounts with SUPER privilege
- Accounts with to many privileges
- No password rotation
- Unused accounts
- Access to mysql.user
- Using MySQL 'root' user
- Run as OS 'root' user

Level 100

- Excessive `data_dir` privileges
- External server access
- Restrict SUPER access (server)
- Limit resource access (SUPER)
- No TLS in transit
- No encryption in rest
- Plain text passwords (filesystem)
- Insecure backups (unencrypted)
- Searchable default passwords
- Exposed credentials (.env)

Level 200

Five (5) Proactive Processes

1. Rotating Passwords

- It was hard before 8.0
- It is easy for 8.0+
  - `RETAIN CURRENT PASSWORD`
  - `DISCARD OLD PASSWORD`

1. How To - Rotating Passwords

```
CREATE USER brazil25 IDENTIFIED WITH caching_sha2_password BY 'MySQL80#password';
ALTER  USER brazil25 IDENTIFIED BY 'MySQL80#newpassword'RETAIN CURRENT PASSWORD;
```

```
$ mysql -ubrazil25 -p MySQL80#password  -e "SELECT USER(),
NOW()"
+--------------------+---------------------+
| USER()             | NOW()               |
+--------------------+---------------------+
| brazil25@localhost | 2025-09-27 16:20:04 |
+--------------------+---------------------+
```

```
$ mysql -ubrazil25 -p MySQL80#newpassword  -e "SELECT USER(),
NOW()"
+------------------------+---------------------+
| USER()                 | NOW()               |
+------------------------+---------------------+
| brazil25@192.168.4.208 | 2025-09-27 16:20:12 |
+------------------------+---------------------+
```

```
ALTER  USER brazil25 DISCARD OLD PASSWORD;
```

2. Use `caching_sha2_password`

- `mysql_native_password` plugin is insecure

```
mysql> SELECT plugin, COUNT(*) FROM mysql.user GROUP BY plugin;
+-----------------------+----------+
| plugin                | COUNT(*) |
+-----------------------+----------+
| caching_sha2_password |       25 |
| mysql_native_password |        6 |
+-----------------------+----------+
```

- See presentation "RIP mysql_native_password"

  https://speakerdeck.com/ronaldbradford/rip-mysql-native-password-2025-04

2. How To - Using `caching_sha2_password`

Pre MySQL 8.0 (GA in 2018) application

- Upgrade to MySQL 8.0 Database
  - Keep using `mysql_native_password`
- Upgrade client connectors to 8.0+ compatible versions
  - i.e. Both plugins are available
- Switch users from `mysql_native_password` to `caching_sha2_password`
  - Ideally create new users (Auditability)
  - Use PASSWORD EXPIRE for old user

3. Using A Stronger Password Policy

- Remove the ability to use easy passwords
- Doesn't have to be random characters
  - e.g. *Mercury-Venus-Earth-Mars-2025*

https://dev.mysql.com/doc/refman/8.0/en/validate-password.html

3. How To - Using A Stronger Password Policy

```
mysql> SHOW GLOBAL VARIABLES LIKE 'validate%';
+--------------------------------------------------+-------+
| Variable_name                                    | Value |
+--------------------------------------------------+-------+
| validate_password.changed_characters_percentage  | 0     |
| validate_password.check_user_name                | ON    |
| validate_password.dictionary_file                |       |
| validate_password.length                         | 8     |
| validate_password.mixed_case_count               | 1     |
| validate_password.number_count                   | 1     |
| validate_password.policy                         | LOW   |
| validate_password.special_char_count             | 1     |
+--------------------------------------------------+-------+

mysql> SHOW STATUS LIKE 'validate_password.%';
+-----------------------------------------------+---------------------+
| Variable_name                                 | Value               |
+-----------------------------------------------+---------------------+
| validate_password.dictionary_file_last_parsed | 2025-08-14 06:21:08 |
| validate_password.dictionary_file_words_count | 0                   |
+-----------------------------------------------+---------------------+
```

3. How To - Using A Stronger Password Policy


```
mysql> SET GLOBAL validate_password.policy=STRONG;
mysql> SET GLOBAL validate_password.length=16;

mysql> SELECT VALIDATE_PASSWORD_STRENGTH('qwerty123');
+-----------------------------------------+
| VALIDATE_PASSWORD_STRENGTH('qwerty123') |
+-----------------------------------------+
|                                      25 |
+-----------------------------------------+

mysql> SELECT VALIDATE_PASSWORD_STRENGTH('Brazil-World-Cup-2026');
+-------------------------------------------------------+
| VALIDATE_PASSWORD_STRENGTH('Brazil-World-Cup-2026') |
+-------------------------------------------------------+
|                                                   100 |
+-------------------------------------------------------+
```

4. Limit SUPER and GRANT OPTION Access

- No generic users
- Limit hosts they can connect from
- Limit concurrent connections (`MAX_USER_CONNECTIONS`)
- Limit queries per hour (`MAX_QUERIES_PER_HOUR`)
- Change the default from 'root' to another user
- Lock on invalid passwords

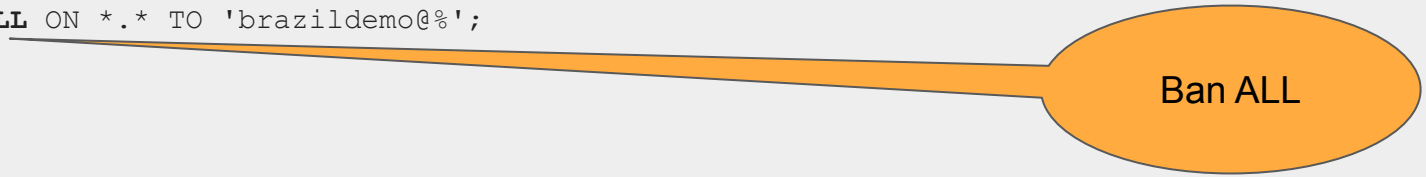
4. How To - Find SUPER access

```
mysql> SELECT host,user FROM mysql.user WHERE super_priv='Y';
+-----------+------------------+
| host      | user             |
+-----------+------------------+
| %         | dba              |
| %         | readyset         |
| 172.%     | hammerdb         |
| localhost | mysql.session    |
| localhost | root             |
+-----------+------------------+
```

```
mysql> SELECT host,user FROM mysql.user WHERE Grant_priv='Y';
+-----------+------+
| host      | user |
+-----------+------+
| %         | root |
| localhost | root |
+-----------+------+
```

4. How To - Limit SUPER access

```
mysql> CREATE USER 'brazildemo@%' IDENTIFIED BY 'Brazil-World-Cup-2026';
mysql> GRANT ALL ON *.* TO 'brazildemo@%';
```




```
ALL = GRANT
APPLICATION_PASSWORD_ADMIN,AUDIT_ABORT_EXEMPT,AUDIT_ADMIN,AUTHENTICATION_POLICY_ADMIN,BACKU
P_ADMIN,BINLOG_ADMIN,BINLOG_ENCRYPTION_ADMIN,CLONE_ADMIN,CONNECTION_ADMIN,ENCRYPTION_KEY_AD
MIN,FIREWALL_EXEMPT,FLUSH_OPTIMIZER_COSTS,FLUSH_STATUS,FLUSH_TABLES,FLUSH_USER_RESOURCES,GR
OUP_REPLICATION_ADMIN,GROUP_REPLICATION_STREAM,INNODB_REDO_LOG_ARCHIVE,INNODB_REDO_LOG_ENAB
LE,PASSWORDLESS_USER_ADMIN,PERSIST_RO_VARIABLES_ADMIN,REPLICATION_APPLIER,REPLICATION_SLAVE
_ADMIN,RESOURCE_GROUP_ADMIN,RESOURCE_GROUP_USER,ROLE_ADMIN,SENSITIVE_VARIABLES_OBSERVER,SER
VICE_CONNECTION_ADMIN,SESSION_VARIABLES_ADMIN,SET_USER_ID,SHOW_ROUTINE,SYSTEM_USER,SYSTEM_V
ARIABLES_ADMIN,TABLE_ENCRYPTION_ADMIN,TELEMETRY_LOG_ADMIN,XA_RECOVER_ADMIN ON *.* TO
`brazildemo@%`@`%`
```

https://dev.mysql.com/doc/refman/8.0/en/grant.html

5. Use the new features

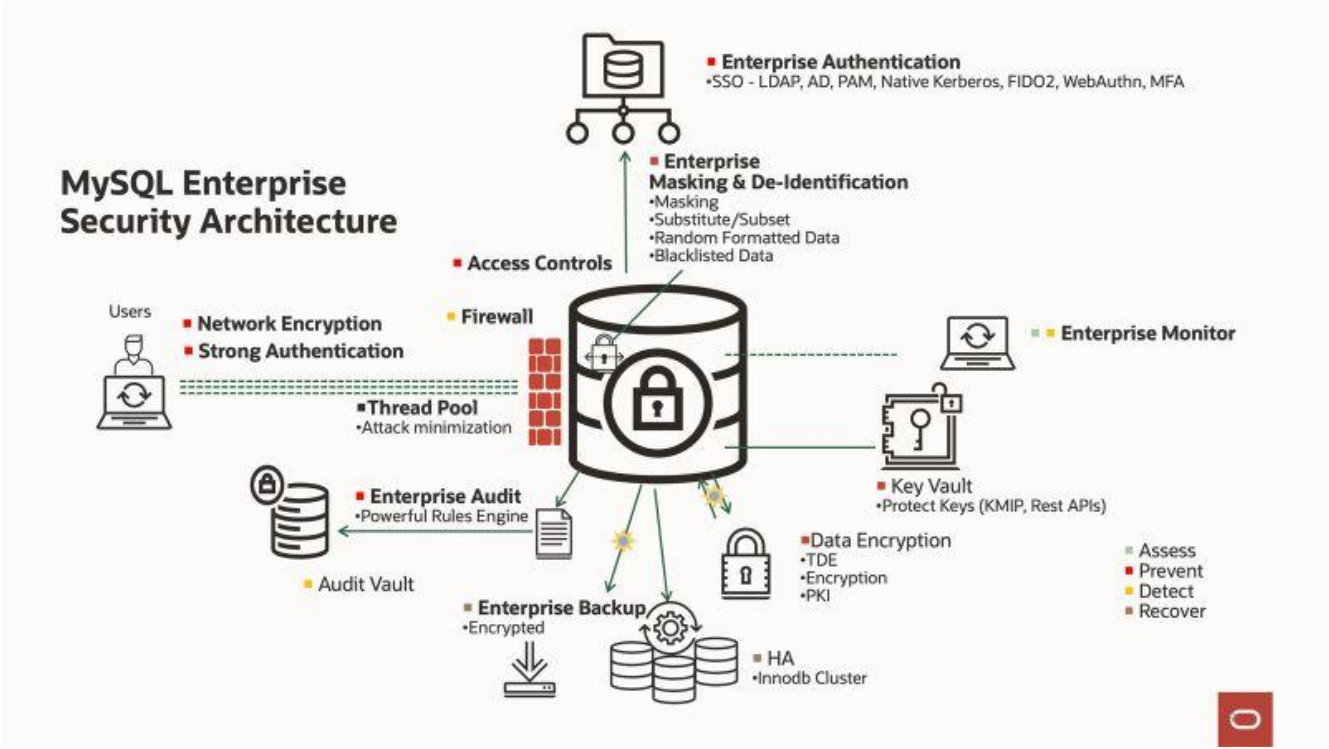
- Expire important passwords (See `EXPIRE INTERVAL 'n' DAY`)
- TLS/SSL (See `REQUIRE SSL`)
- Disable use of old password (See `PASSWORD HISTORY 'n'`)
- Lock accounts on failed (See `FAILED_LOGIN_ATTEMPTS 'n'`)
- MFA - Complicated

https://dev.mysql.com/doc/refman/9.3/en/create-user.html

DO NOT DO THIS - Days not Minutes


```
mysql> ALTER USER root@localhost FAILED_LOGIN_ATTEMPTS 3 PASSWORD_LOCK_TIME 60;

$ mysql -uroot -py -e "SELECT NOW()"

ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)

$ mysql -uroot -py -e "SELECT NOW()"

ERROR 1045 (28000): Access denied for user 'root'@'localhost' (using password: YES)

$ mysql -uroot -py -e "SELECT NOW()"

ERROR 3955 (HY000): Access denied for user 'root'@'localhost'.

 Account is blocked for 60 day(s) (60 day(s) remaining) due to 3 consecutive failed logins.
```

MySQL Reference Architectures for Security



https://www.mysql.com/why-mysql/white-papers/mysql-reference-architectures-for-security/

Password Security 101

- Use stronger authentication rules
- Use least-privileged authorization
- Individual user per service, product, tool, monitoring
- Move to centralized, token based user management
  - Weakest link - Plain text credentials (in files)

https://dev.mysql.com/doc/refman/9.3/en/create-user.html

Conclusion

- EASY - Do the minimum for applications
- LESS EASY - Upgrading the database users
  - New admin process
- HARD - Replacing all client connectors
- HARD - Proactive controls

What is more important?

HARD work or being **HACKED**

Thank You

https://ronaldbradford.com