PERCONA

Databases run better with Percona

MySQL Belgian Days 2024 - MySQL Password Complexity

Date: 2024-02-02

Time: 13:00–13:30

MySQL Password Complexity

Best practices dictate that you rotate passwords regularly, require a certain amount of complexity in the password itself, and not be something obvious (no 'password' or 's3cr3t').

So how to you set up your instance to do all that? You will learn the options, the restrictions, the best practices to have secure passwords that meet your requirements, and how to monitor their status.

MySQL Password Complexity

Dave Stokes
@Stoker
David.Stokes@Percona.com

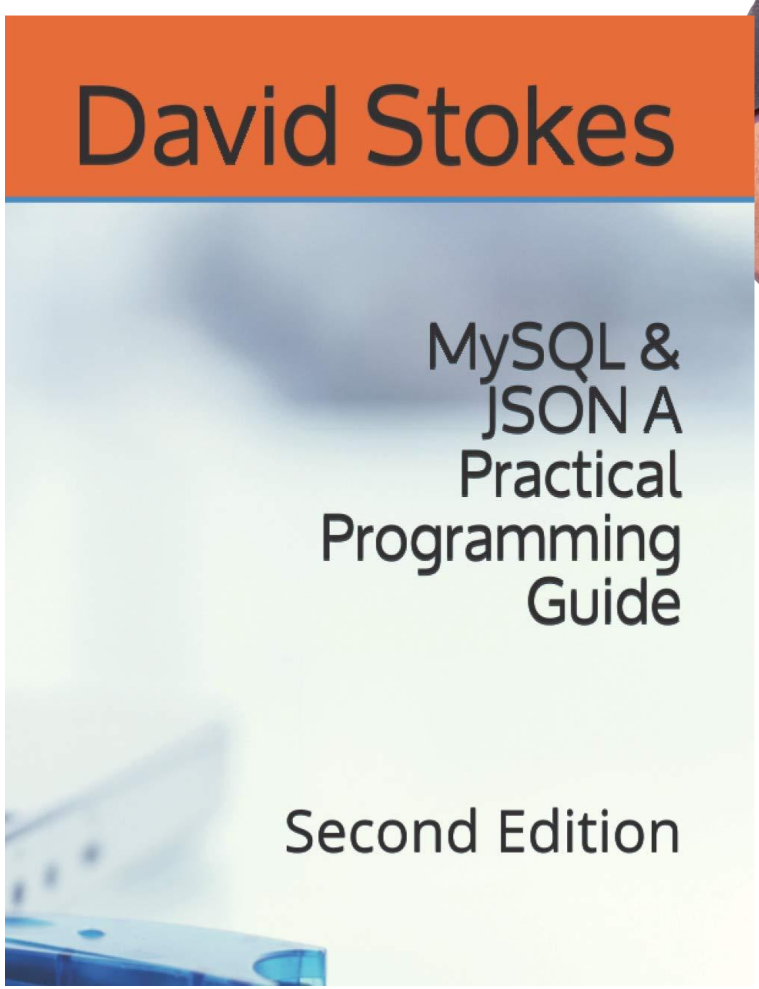
About Me!

Dave Stokes

Technology Evangelist

David.Stokes@Percona.com

@Stoker



David Stokes

MySQL & JSON A Practical Programming Guide

Second Edition

Passwords

General Advice

Maastricht University
Treat your passwords like your underwear
• Never share them with anyone
• Change them regularly
• Keep them off your desk
Be aware of the new European law on privacy and personal data, General Data Protection Regulation – effective 25 May 2018
www.maastrichtuniversity.nl/privacy

Treat your passwords like your underwear

- Never share then with anyone
- Change them regularly
- Keep them off your desk

Where are password stored?

```
SQL > select User, Host, authentication_string
from user
where User='root'\g
+------+-----------+-------------------------------------------+
| User | Host      | authentication_string                     |
+------+-----------+-------------------------------------------+
| root | localhost | *C22B6ED4C01FFB958C87E92A2B5A7CA61FF1AA10 |
+------+-----------+-------------------------------------------+
1 row in set (0.0011 sec)
```

https://dev.mysql.com/doc/refman/8.0/en/security-guidelines.html

- ***Do not ever give anyone (except MySQL `root` accounts) access to the `user` table in the `mysql` system database!*** This is critical.

- Try `mysql -u root`. If you are able to connect successfully to the server without being asked for a password, anyone can connect to your MySQL server as the MySQL `root` user with full privileges!.

- Assume that all passwords will be subject to automated cracking attempts using lists of known passwords, and also to targeted guessing using publicly available information about you, such as social media posts.

- Passwords can be written as plain text in SQL statements such as `CREATE USER`, `GRANT` and `SET PASSWORD`. If such statements are logged by the MySQL server as written, passwords in them become visible to anyone with access to the logs.

- Require all MySQL accounts to have a password.

Create an account with a password

```
CREATE USER 'local_user'@'localhost' IDENTIFIED BY 'password';
```

MySQL Password Validation Component

6.4.3 The Password Validation Component

The validate_password component serves to improve security by requiring account passwords and enabling strength testing of potential passwords.

This component exposes system variables that enable you to configure password policy, and status variables for component monitoring.

Complexity

Make versus Buy decision

6.4.3.3 Transitioning to the Password Validation Component

```
mysql> select @@plugin_dir;
+------------------------+
| @@plugin_dir           |
+------------------------+
| /usr/lib/mysql/plugin/ |
+------------------------+
1 row in set (0.01 sec)

mysql> install component 'file://component_validate_password';
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> show variables like 'validate_password.%';
+-------------------------------------------------+--------+
| 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                        | MEDIUM |
| validate_password.special_char_count            | 1      |
+-------------------------------------------------+--------+
8 rows in set (0.01 sec)
```

Policies

```
mysql> show variables like 'validate_password.%';
+-------------------------------------------------+--------+
| 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                        | MEDIUM |
| validate_password.special_char_count            | 1      |
+-------------------------------------------------+--------+
8 rows in set (0.01 sec)
```

LOW

The LOW policy tests password length only.

Passwords must be at least 8 characters long.

To change this length, modify validate_password.length.

MEDIUM

The MEDIUM policy adds the conditions that passwords must contain at leas1

- 1 numeric character
- 1 lowercase character
- 1 uppercase character
- 1 special (nonalphanumeric) character.

To change these values, modify validate_password.number_count, validate_password.mixed_case_count, and validate_password.special_char_count.

STRONG

The STRONG policy **adds** the condition that password substrings of length 4 or longer must not match words in the dictionary file, if one has been specified.

To specify the dictionary file, modify validate_password.dictionary_file.

Two Passwords?

Yes, you can have two passwords!

```
SQL>ALTER USER 'dualtest'@'192.168.4.%' IDENTIFIED BY 'password2' RETAIN CURRENT PASSWORD;
```

```
SQL>select user,host, plugin, authentication_string, password_last_changed,User_attributes
from mysql.user where user ='dualtest' order by 1,2G
*************************** 1. row ***************************
                 user: dualtest
                 host: 192.168.4.%
            plugin: mysql_native_password
authentication_string: *DC52755F3C09F5923046BD42AFA76BD1D80DF2E9
password_last_changed: 2022-11-17 08:46:28
      User_attributes: {"additional_password": "*668425423DB5193AF921380129F465A6425216D0"}
1 row in set (0.00 sec)
```

```
SQL>ALTER USER 'dualtest'@'192.168.4.%' DISCARD OLD PASSWORD;
```

Proper Passwords

Not ‘password’ or ‘thebossisajerk’

Create a list of forbidden passwords

```
root@test1:/etc/mysql/mysql.conf.d# cat ../badpasswords
password
passwd
thebossisajerk
secret
s3cr3t
notlongenough
```

Modify the config file before restarting

```
root@test1:/etc/mysql/mysql.conf.d# cat mysqld.cnf
#
# The Percona Server 8.0 configuration file.
#
# For explanations see
# http://dev.mysql.com/doc/mysql/en/server-system-variables.html

[mysqld]
pid-file     = /var/run/mysqld/mysqld.pid
socket            = /var/run/mysqld/mysqld.sock
datadir           = /var/lib/mysql
log-error    = /var/log/mysql/error.log
validate_password.dictionary_file = /etc/mysql/badpasswords
```

And test

```
mysql> create user 'baspass'@'localhost' identified by 'password';
ERROR 1819 (HY000): Your password does not satisfy the current policy requirements
mysql> create user 'badpass'@'localhost' identified by 'abc123';
ERROR 1819 (HY000): Your password does not satisfy the current policy requirements
mysql> create user 'badpass'@'localhost' identified by 's3cret#';
ERROR 1819 (HY000): Your password does not satisfy the current policy requirements
mysql>
```

Other Options

That may be helpful

```
mysql> show variables like 'validate_password.%';
+-------------------------------------------------+--------+
| 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                        | MEDIUM |
| validate_password.special_char_count            | 1      |
+-------------------------------------------------+--------+
8 rows in set (0.01 sec)
```

Checking user name

```
mysql> create user 'foobar'@'localhost' IDENTIFIED BY 'foobar';
ERROR 1819 (HY000): Your password does not satisfy the current policy
requirements
mysql> create user 'foobar'@'localhost' IDENTIFIED BY '@foobar123';
ERROR 1819 (HY000): Your password does not satisfy the current policy
requirements
mysql>
mysql> show variables like 'validate_password.%';
+-------------------------------------------------+--------+
| 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                        | MEDIUM |
| validate_password.special_char_count            | 1      |
+-------------------------------------------------+--------+
8 rows in set (0.01 sec)
```

Changed Character Percentage

Indicates the minimum number of characters, as a percentage of all characters, in a password that a user must change before validate_password accepts a new password for the user's own account.

Has anyone got this to work??

Rotation

Lifetime, Expire, & Reuse

```
default_password_lifetime=180  # Measured in Days

default_password_lifetime=0     # Does not expire

SET PERSIST default_password_lifetime = 180;  # Theses setting can be set at runtime too.
```

```
CREATE USER 'jeffrey'@'localhost' PASSWORD EXPIRE INTERVAL 90 DAY;
ALTER USER 'jeffrey'@'localhost' PASSWORD EXPIRE INTERVAL 90 DAY;

CREATE USER 'jeffrey'@'localhost' PASSWORD EXPIRE DEFAULT;
```

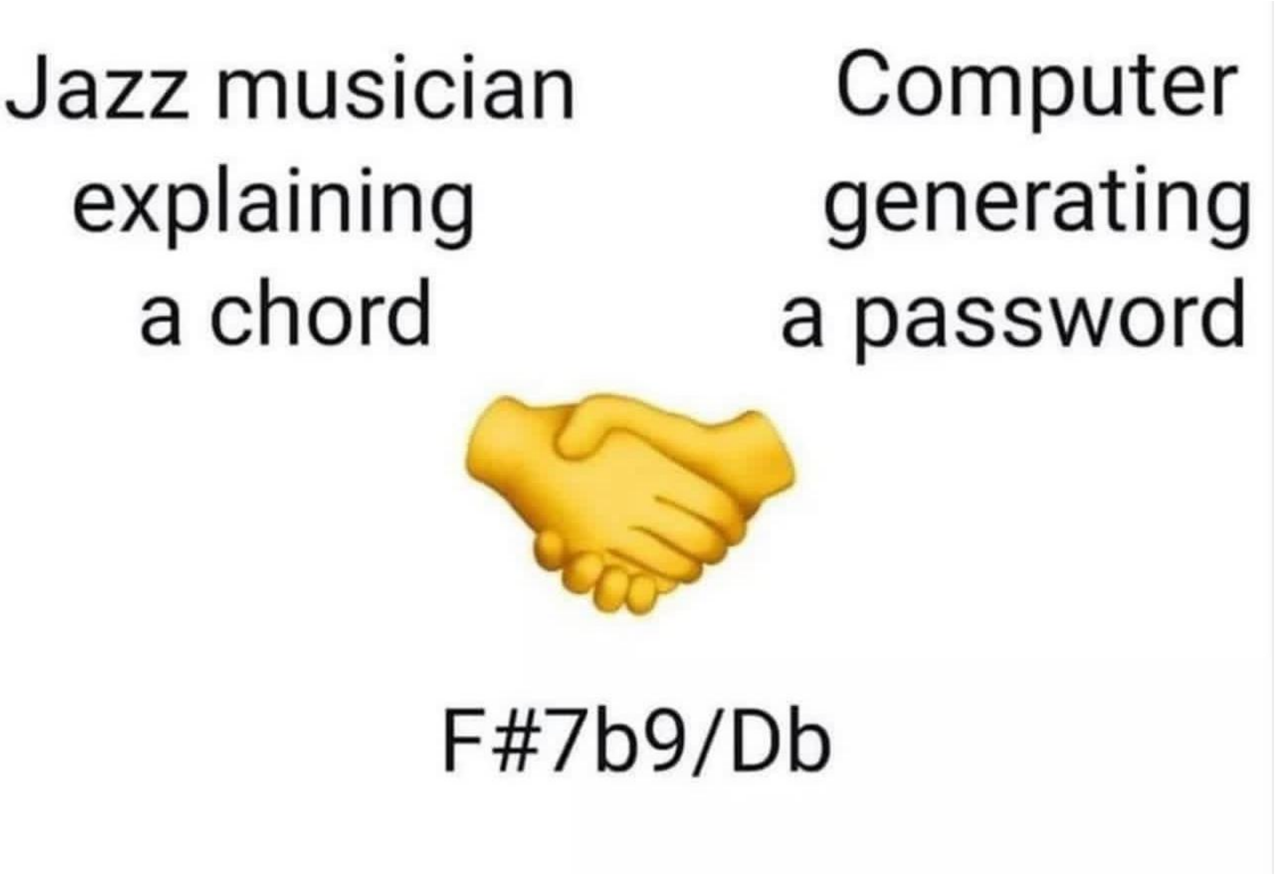
```
password_history=6;    # Have to use six passwords before repeatings
password_reuse_interval=365 EXPIRE DEFAULT;

CREATE USER 'jeffrey'@'localhost' PASSWORD REUSE INTERVAL 365 DAY;
ALTER USER 'jeffrey'@'localhost' PASSWORD REUSE INTERVAL 365 DAY;
```

Wrap up

Use good passwords

- Make them complex
- Rotate them on a regular bases
- Do not use the same password over and over
- Use roles
- Double check with your corporate policy
- Paranoia is not necessarily a bad thing

Use good passwords


Jazz musician explaining a chord
Computer generating a password
F#7b9/Db

Thank You!

David.Stokes@Percona.com
@Stoker
Speakerdeck.com/Stoker