

Securing your MySQL/MariaDB Data

Ronald Bradford, Colin Charles
Percona Live Europe Amsterdam 2016

About: Colin Charles

- Chief Evangelist (in the CTO office), Percona Inc
- Founding team of MariaDB Server (2009-2016), previously at Monty Program Ab, merged with SkySQL Ab, now MariaDB Corporation
- Formerly MySQL AB (exit: Sun Microsystems)
- Past lives include Fedora Project (FESCO), OpenOffice.org
- MySQL Community Contributor of the Year Award winner 2014
- <http://bytebot.net/blog/>

About: Ronald Bradford


- Experienced MySQL database guy
- Author/Blogger/Speaker
- Looking for my next great opportunity

- <http://ronaldbradford.com/presentations/>
- <http://effectivemysql.com>

Agenda

- Observed insecure practices
- Securing communications
- Securing connections
- Securing data
- Securing user accounts
- Securing server access

Signs of Poor Security




Found in
any version

- old_passwords
- Users without passwords
- Anonymous users
- GRANT privilege users
- ALL privilege users
- '%' host user accounts
- 'root' MySQL user without password
- 'root' MySQL user
- Generic OS DBA user e.g. 'dba'
- Disabled OS Firewall/SELinux/Apparmor
- Open data directory privileges
- Default test database

Easy Fixes

```
$ mysql_secure_installation
```

Current Insecure Practices



Very easy to
fix practices

- Using password on command line
 - Command history
 - MySQL shell history
- Using simple passwords
 - It's just a test environment
- Using excessive permissions
 - GRANT, ALL, *.* , %

Why being SUPER is bad (GRANT ALL ON *.*)

- Bypasses read_only
- Bypasses init_connect
- Can disable binary logging
- Can change dynamic configuration
- Takes the reserved connection

<http://ronaldbradford.com/blog/why-grant-all-is-bad-2010-08-06/>

<http://effectivemysql.com/presentation/mysql-idiosyncrasies-that-bite/>

Secure Communications

- SSL for replication
- SSL for client connections
- SSL for admin connections
- Encryption on the wire

<https://dev.mysql.com/doc/refman/5.6/en/secure-connections.html>

<https://dev.mysql.com/doc/refman/5.7/en/secure-connections.html>

Secure Communications

```
[mysqld]
```

```
ssl-ca=ca.pem
```

```
ssl-cert=server-cert.pem
```

```
ssl-key=server-key.pem
```

SSL Protocols and Ciphers

```
mysql> SHOW SESSION STATUS LIKE 'Ssl_version';
```

```
+-----+-----+
| Variable_name | Value |
+-----+-----+
| Ssl_version   | TLSv1 |
+-----+-----+
```

```
mysql> SHOW SESSION STATUS LIKE 'Ssl_cipher';
```

```
+-----+-----+
| Variable_name | Value |
+-----+-----+
| Ssl_cipher    | DHE-RSA-AES128-GCM-SHA256 |
+-----+-----+
```

SSL Client Connections

<https://dev.mysql.com/doc/connector-python/en/connector-python-connectargs.html>

```
import mysql.connector
from mysql.connector.constants import ClientFlag

config = {
    'user': 'ssluser',
    'password': 'asecret',
    'host': '127.0.0.1',
    'client_flags': [ClientFlag.SSL],
    'ssl_ca': '/opt/mysql/ssl/ca.pem',
    'ssl_cert': '/opt/mysql/ssl/client-cert.pem',
    'ssl_key': '/opt/mysql/ssl/client-key.pem',
}
```

<https://dev.mysql.com/doc/connectors/en/connector-net-tutorials-ssl.html>

Secure Connections

- `mysql_ssl_rsa_setup` in MySQL 5.7
 - This program creates the SSL certificate and key files and RSA key-pair files required to support secure connections using SSL and secure password exchange using RSA over unencrypted connections, if those files are missing.
- uses the `openssl` command

Secure Storage

- Encryption of data at rest
 - Data (table vs tablespace)
 - Binary Logs
 - Other Logs
- Key management

Encryption in MariaDB Server

- Encryption: **tablespace** OR **table** level encryption with support for rolling keys using the AES algorithm
 - table encryption — `PAGE_ENCRYPTION=1`
 - tablespace encryption — encrypts everything including log files
- `file_key_management_filename`, `file_key_management_filekey`, `file_key_management_encryption_algorithm`
- Well documented —
<https://mariadb.com/kb/en/mariadb/data-at-rest-encryption/>
- Tablespace/logs scrubbing: background process that regularly scans through the tables and upgrades the encryption keys
- `--encrypt-tmp-files` & `--encrypt-binlog`

Encryption in MariaDB Server II

[mysqld]

plugin-load-add=file_key_management.so

file-key-management

file-key-management-filename = /home/mdb/keys.enc

innodb-encrypt-tables

innodb-encrypt-log

innodb-encryption-threads=4

aria-encrypt-tables=1 # PAGE row format

encrypt-tmp-disk-tables=1 # this is for Aria

```
CREATE TABLE customer (
```

```
customer_id bigint not null primary key,
```

```
customer_name varchar(80),
```

```
customer_creditcard varchar(20))
```

```
ENGINE=InnoDB
```

```
page_encryption=1
```

```
page_encryption_key=1;
```


Encryption in MariaDB Server III

- Use the preset! - /etc/my.cnf.d/enable_encryption.preset
- MariaDB Enterprise has a plugin for Amazon Key Management Server (KMS)
 - The reality is you can just compile this for MariaDB Server
- mysqlbinlog has no way to read (i.e. decrypt) an encrypted binlog
- This does not work with MariaDB Galera Cluster yet (gcache is not encrypted yet), and also xtrabackup needs additional work (i.e. if you encrypt the redo log)

Encryption in MySQL

- MySQL 5.7.11 introduces InnoDB tablespace encryption
- `early-plugin-load=keyring_file.so` in `my.cnf`
- Must use `innodb_file_per_table`
- Convert via `ALTER TABLE table ENCRYPTION='Y'`
- Data is not encrypted in the redo/undo/binary logs
- Has external key management (Oracle Key Vault)

Secure Accounts

- Privileges
- Passwords
- Password filesystem storage

MySQL 5.6 improvements

- Password expiry - `ALTER USER 'foo'@'localhost' PASSWORD EXPIRE;`
- Password validation plugin - `VALIDATE_PASSWORD_STRENGTH()`
- `mysql_config_editor` - store authentication credentials in an encrypted login path file named `.mylogin.cnf`
 - <http://dev.mysql.com/doc/refman/5.6/en/mysql-config-editor.html>
- Random 'root' password on install
 - `mysql_install_db --random-passwords` stored in `$HOME/.mysql_secret`

MySQL 5.7 improvements

- Improved password expiry — automatic password expiration available, so set `default_password_lifetime` in `my.cnf`
- You can also require password to be changed every n-days
 - `ALTER USER 'foo'@'localhost' PASSWORD EXPIRE INTERVAL n DAY;`
- There is also account locking/unlocking now
 - `ACCOUNT LOCK/ACCOUNT UNLOCK`

MariaDB Server passwords

- Password validation plugin (finally) exists now
 - <https://mariadb.com/kb/en/mariadb/development/mariadb-internals-documentation/password-validation/>
- `simple_password_check` password validation plugin
 - can enforce a minimum password length and guarantee that a password contains at least a specified number of uppercase and lowercase letters, digits, and punctuation characters.
- `cracklib_password_check` password validation plugin
 - Allows passwords that are strong enough to pass CrackLib test. This is the same test that `pam_cracklib.so` does

Authentication in MySQL / MariaDB Server

- `Auth_socket` - Authenticates against the Unix socket file, using `so_peercred`
- `Sha256_password` - `default-authentication-plugin=sha256_password`, passwords never exposed as cleartext when connecting; SSL or RSA auth
- Kerberos/GSSAPI/SSPI - User principals: `<username>@<KERBEROS REALM>`
- Active Directory (Enterprise only)
- `Mysql_no_login` (MySQL 5.7) - prevents all client connections to an account that uses it

PAM authentication

Percona Server

```
INSTALL PLUGIN auth_pam SONAME 'auth_pam.so'; INSTALL SONAME 'auth_pam';
```

```
CREATE USER byte IDENTIFIED WITH auth_pam;
```

In `/etc/pam.d/mysqld`:

```
auth      required      pam_warn.so
auth      required      pam_unix.so audit
account   required      pam_unix.so audit
```

MariaDB Server

```
CREATE USER byte IDENTIFIED via pam USING
'mariadb';
```

Edit `/etc/pam.d/mariadb`:

```
auth      required      pam_unix.so
account   required      pam_unix.so
```

Just use `—pam-use-clear-text-plugin` for MySQL to use `mysql_clear_text_password` instead of dialog plugin

SQL standard Roles

- Bundles users together, with similar privileges - follows the SQL standard
- MariaDB Server 10.0 (10.1 adds that each user can have a DEFAULT ROLE)
- MySQL 8.0 DMR

```
CREATE ROLE audit_bean_counters;
```

```
GRANT SELECT ON accounts.* to audit_bean_counters;
```

```
GRANT audit_bean_counters to ceo;
```

Auditing

MySQL

- Logging account access
- Logging SQL statements
- Logging uncommon SQL patterns

OS

Logging account logins

Logging sudo commands

#PerconaLive @bytebot @RonaldBradford

Auditing Implementation

- MariaDB Server
 - User filtering as an additional feature via audit API extensions
 - Query cache enabled? No table records
- Percona
 - Multiple output formats: OLD, NEW, JSON, CSV
 - Filter by user, SQL command type, database,
 - Auditing can be expensive, so asynchronous/performance/semisynchronous/synchronous modes for logging - e.g. log using memory buffers, drop messages if buffers are full, or log directly to file, flush and sync at events, etc.
- McAfee Audit plugin
 - Uses offsets
- MySQL Enterprise Audit Plugin (utility: `mysqldauditgrep`)

Firewall Implementation

MySQL - MySQL Enterprise Firewall

MariaDB - MariaDB MaxScale dbfirewallfilter

Percona - use MariaDB MaxScale / ProxySQL

SQL Injection

- Always using bind variables
- Escape input content
- Restricted "least" privileges
 - Do not have GRANT ALL

Database Server Access

- Restricting user access to your database server (login accounts)
 - Every physical person has a dedicated login
 - Separate OS & Database accounts
 - sudo restrictions (e.g. `sudo su -`)
 - Setup sudo group
 - Grant only specific commands to execute
 - Never share account details
- MFA

Database Server Access

- Restricting traffic to your database server (open ports)
- Run a software firewall
 - iptables, ufw
- You should use OS software meant to benefit security
 - SELinux / Apparmor

Database Server Access

- If you can login, and stop MySQL, you can bypass security
 - `--skip-grant-tables`
- If you can edit `/etc/my.cnf` you can set
 - `--init-file=/path/to/my.sql`
- If you use `--init-file`, can you modify content of file

Database Server Access

- Restrict access to datadir
- Restrict access to view mysql.user table on filesystem
- Check out the examples of how to Hack MySQL

<http://effectivemysql.com/downloads/MySQLSecurityEssentialsPerconaLive2015.pdf>

Installation

- Using your Linux distribution... mostly gets you MariaDB when you ask for `mysql-server`
 - Except on Debian/Ubuntu
 - However, when you get `mariadb-server`, you get an authentication plugin — `auth_socket` for “automatic logins”
 - You are asked by `debhelper` to enter a password
- You can use the APT/YUM repositories from Oracle MySQL, Percona or MariaDB
- Don't disable SELinux: `system_u:system_r:mysqld_t:s0`

Update cadence

A security patch is so named because it improves security and generally addresses a means of attack of your system

- OS
- Database
- Application Stack

Why are you still running MySQL 5.5 or older?

Deployment Security

Who has control over running deployments?

i.e. deploying code that manipulates your data or structure

An application user SHOULD NOT have CREATE, ALTER, DROP privileges

- User to write data
- User to read data
- DBA to administer data (restricted to localhost)

Use of Docker Containers

Docker shows a disregard for security with 'root' OS logins by default

- MySQL server installation approach via exposed passwords
 - See Giuseppe's MySQL Docker operations tutorial
- Configuration is contained with container
 - Can you access the container via SSH
 - Can you copy and use container

Reference Material

<https://www.mysql.com/why-mysql/presentations/mysql-security-best-practices/>

- MySQL Authentication and Password Policies
- MySQL Authorization and Privilege Management
- MySQL Encryption to secure sensitive data
- MySQL Enterprise Firewall to block database attacks such as an SQL Injection
- MySQL Enterprise Audit to implement policy

MySQL Manual Security Practices

<http://dev.mysql.com/doc/refman/5.5/en/security.html>

<http://dev.mysql.com/doc/refman/5.6/en/security.html>

<http://dev.mysql.com/doc/refman/5.7/en/security.html>

Rate us! Thanks/Q&A

- Don't forget to rate us in the app!
- Ronald Bradford: @RonaldBradford / <http://effectivemysql.com>
- Colin Charles: @bytebot / <http://www.bytebot.net/blog/>