PCI/DSS Compliance with MySQL: 2019 Edition

Room A, 4:00 PM

Carlos Tutte, MySQL Support Engineer
Percona

PERCONA
LIVE EUROPE
AMSTERDAM

Agenda

- Introduction
- What is PCI DSS?
- PCI DSS requirements list
- How to implement PCI DSS with MySQL
- Conclusions
- References
- Questions

What is PCI DSS?

What is PCI DSS?

- Payment Card Industry Data Security Standard
- Set of 12 requirements for businesses handling cardholder data
- Created by the major payment brand cards
- Aims to reduce CC fraud

PCI DSS Requirement List

PCI DSS Requirement List

Goals	PCI DSS Requirements
Build and Maintain a Secure Network	1. Install and maintain a firewall configuration to protect cardholder data 2. Do not use vendor-supplied defaults for system passwords and other security parameters
Protect Cardholder Data	3. Protect stored data 4. Encrypt transmission of cardholder data across open, public networks
Maintain a Vulnerability Management Program	5. Use and regularly update anti-virus software or programs 6. Develop and maintain secure systems and applications
Implement Strong Access Control Measures	7. Restrict access to cardholder data by business need-to-know 8. Assign a unique ID to each person with computer access 9. Restrict physical access to cardholder data
Regularly Monitor and Test Networks	10. Track and monitor all access to network resources and cardholder data 11. Regularly test security systems and processes
Maintain an Information Security Policy	12. Maintain a policy that addresses information security for all personnel

PCI DSS requirements NOT tackled by MySQL

- Requirement 1: Install and maintain a firewall configuration to protect cardholder data
- Requirement 5: Protect all systems against malware and regularly update anti-virus software or programs
- Requirement 9: Restrict physical access to cardholder data
- Requirement 11: Regularly test security systems and processes
- Requirement 12: Maintain a policy that addresses information security for all personal

They are non-database related!

PCI DSS requirement list tackled by MySQL

- Requirement 2: Not Using Vendor Default Passwords and Security Settings
- Requirement 3: Protect Stored Cardholder Data
- Requirement 4: Encrypt transmission of cardholder data across open, public networks
- Requirement 6: Develop and Maintain Secure Systems and Applications
- Requirement 7: Restrict Access to Cardholder Data by Business Need to Know
- Requirement 8: Identify and Authenticate Access to System Components
- Requirement 10: Track and Monitor Access to Cardholder Data

Requirement 2:
Not Using Vendor Default Passwords and Security Settings

Requirement 2:
Not Using Vendor Default Passwords and Security Settings

1. Implement one primary function per server, i.e: dedicated MySQL Server
2. Enable only necessary services, protocols, daemons, ports... etc
3. Change password for root accounts when DBAs leave the company or change position
4. **Disable anonymous accounts and default passwords**
5. **Configure system security parameters to prevent misuse**

Requirement 2.4
Disable anonymous accounts and defaults

- Use mysql_secure_installation binary, which covers:
  - Setup of VALIDATE PASSWORD plugin
  - Level of password validation policy
  - Setting password for root
  - Removal of anonymous users
  - Revoking remote root logins
  - Removal of default schemas and grants for it

Requirement 2.5
Configure system security parameters to prevent misuse

- In MySQL some variables can be tuned to improve security:
  - Disable local_infile
  - Set secure_file_priv to NULL (not dynamic)
  - Disable old_passwords (removed in 8.0.11)
  - Enable secure_auth (removed in 8.0.3)
  - Set read_only and super_read_only = 1 on Slave

Requirement 3:
Protect Stored Cardholder Data

Requirement 3:
Protect Stored Cardholder Data

1. Do not store sensitive data after payment authorization and do not store full content of any track
2. **Mask PAN (only display first 6 and last 4 digits) and protect with strong encryption**
3. **Secure cryptographic key storage**
4. **Cryptographic key change after a defined period of time**

Requirement 3.2
Strong encryption

- For encryption, 3 methods are possible {full disk, db, app level}
- Since MySQL 5.7.11, data at rest encryption + keyring_file plugin available
  - PCI-DSS 3.5.2 requirement data and key must be stored separately
  - Keyring_file makes this harder
- Since Percona Server 5.7.20, Hashicorp vault plugin can also be used
- InnoDB uses a two tier encryption key architecture

Requirement 3.2
Strong encryption

- After enabling encryption, creating encrypted tables:

```
mysql> CREATE TABLE ... ENCRYPTION='Y';
mysql> ALTER TABLE ... ENCRYPTION='Y';
```

- Overhead of encryption is < 10%;
  - Some tests on Galera clusters shows ~20%

Requirement 3.2
Strong encryption

- Binlog encryption PS: 5.7.20; upstream 8.0.14

binlog_encryption=TRUE

- Encrypt redo log and undo log PS 5.7.23; upstream 8.0

innodb_redo_log_encryp=TRUE

innodb_undo_log_encrypt=TRUE

Requirement 3.2
Strong encryption

PS:

- encrypt_binlog
- encrypt_tmp_files
- innodb_parallel_dblwr_encrypt
- innodb_encrypt_online_alter_logs
- innodb_encrypt_tables
- innodb_sys_tablespace_encrypt
- innodb_temp_tablespace_encrypt
- keyring_file_data
- keyring_operations
- keyring_vault_config
- keyring_vault_timeout

MySQL enterprise:

- keyring_aws_cmk_id
- keyring_aws_conf_file
- keyring_aws_data_file
- keyring_aws_region
- keyring_encrypted_file_data
- keyring_encrypted_file_password
- keyring_okv_conf_dir

Requirement 3.2
Strong encryption with backups

- PXB allows taking backups of encrypted databases:

```
xtrabackup --backup --stream=xbstream --target-dir=./  --transition-key=fc976b7a13de566dbad79056be5ef795 > backup.xb
```

- Extract the backup using the xbstream utility.

```
xbstream -x -C backup/ <  backup.xb
```

- Prepare backup:

```
xtrabackup --prepare --target-dir=backup/ --transition-key=fc976b7a13de566dbad79056be5ef795
```

- Restore!

```
xtrabackup --copy-back --target-dir=backup/ --datadir=/var/lib/mysql --transition-key=fc976b7a13de566dbad79056be5ef795 --generate-new-master-key --keyring-vault-config=/var/lib/mysql-keyring/keyring_vault.conf
```

Requirement 3.3
Secure cryptographic key with Keyring plugin

- Load the keyring plugin before InnoDB:

```
early-plugin-load = keyring_file.so
keyring_file_data = /var/lib/mysql-keyring/keyring
```

- Using keyring_file plugin does not totally comply with PCI DSS

Requirement 3.3
Secure cryptographic key with Vault plugin

- Install and configure Hashicorp Vault
- Enable vault plugin:
  early-plugin-load="keyring_vault=keyring_vault.so"
  loose-keyring_vault_config=".../mysql-keyring/keyring_vault.conf”
- Create keyring vault with following content:
  [root@ps4vault ~]# cat /var/lib/mysql-keyring/keyring_vault.conf
  vault_url = https://10.222.95.198:8200
  secret_mount_point = secret/mysql
  token = s.JKrm9uAEDXkvYiMVgFAJnMIB
  vault_ca = /etc/vault_ca/support.crt
- Restart MySQL and check plugin was installed successfully

Requirement 3.4
Cryptographic key change after a defined period of time

- Master key rotation is an atomic, instance-level operation
- Rotating master key re encrypts all tablespace keys
- Master key can be rotated with the following command:

```
mysql> ALTER INSTANCE ROTATE INNODB MASTER KEY;
```

- ALTER is replicated like any other SQL
- Example with ~4k tables, operation took ~1 second

Requirement 4: Encrypt transmission of cardholder data across open, public networks

Requirement 4:
Encrypt transmission of cardholder data across open, public networks

- SSL and early TLS (v1.0) have vulnerabilities and is not considered safe
- By default, upstream uses TLS v1.1 (yaSSL) but PS uses TLS v1.2
- Upstream 5.6.46, 5.7 and PS 5.6.31 supports TLS 1.2 (openssl)

On 5.6, client has to have ssl certificates:

```
mysql --ssl-key=ssl/client-key.pem --ssl-cert=ssl/client-cert.pem
```

On 5.7 is simpler!

```
mysql -h ... --ssl
```

To make it compulsory for clients, create users with “REQUIRE SSL”:

```
mysql> CREATE USER 'jeffrey'@'localhost' REQUIRE SSL;
```

Requirement 6:
Develop and Maintain Secure Systems and Applications

Requirement 6:
Develop and Maintain Secure Systems and Applications

- Install critical security patches after one month of release, low risk patches after two-three months
  CPU: https://www.oracle.com/technetwork/security-advisory/
- Have a back-out procedure before any change, in case it affects security or functioning
- Separate dev/test env from production env
- Remove testing data before releasing into production
- Do not use live PANs for testing

Requirement 7: Restrict Access to Cardholder Data by Business Need to Know

Requirement 7:
Restrict Access to Cardholder Data by Business Need to Know

1. Use “least privilege necessary to perform the job”
   a. **Limit access to system components and cardholder data to only those individuals whose job requires such access**
   b. **Assign access based on individual personnel’s job classification and function**

Requirement 7.1.a
Limit access through proper column-level grants

- Limit access to system components and cardholder data to only those individuals whose job requires such access.
- Only GRANT privileges for needed schemas/tables/columns

mysql> GRANT SELECT ON test.card_holder_data TO ‘test_user’@’%’;

mysql> GRANT SELECT (id, mail) ON test.card_holder_data TO 'test_user'@'%';

Invalid permissions for select:

mysql> SELECT * FROM card_holder_data;

ERROR 1142 (42000): SELECT command denied to user 'test_user'@’%’ for table 'card_holder_data'

Requirement 7.1.a
Use views to avoid errors with restricted colums

- VIEWs can used to show partial data:

```
mysql> select * from view_card_holder_data;
+----+--------------------------+
| id | mail                     |
+----+--------------------------+
|  1 | carlos.tutte@percona.com |
+----+--------------------------+
1 row in set (0.00 sec)
```

```
mysql> select * from card_holder_data;
+----+--------------------------+-------------+
| id | mail                     | card_digits |
+----+--------------------------+-------------+
|  1 | carlos.tutte@percona.com |        5238 |
+----+--------------------------+-------------+
1 row in set (0.00 sec)
```

Requirement 7.1.b
Assign access based on job and function

- MySQL 8.0 implemented roles!
- A role is a named collection of privileges

```
-- Create role
CREATE ROLE 'app_developer';
-- Give priviletes to role
GRANT INSERT,UPDATE ON app_db.* TO 'app_developer';
-- Give user ‘dev1’@’localhost’, grant role to user
GRANT 'app_developer' TO 'dev1'@'localhost';
```

- On 5.7, roles can be emulated using proxy users + PAM + LDAP groups

Requirement 8: Identify and Authenticate Access to System Components

Requirement 8:
Identify and Authenticate Access to System Components

1. User management done by specific authority
2. Unique user ID for each person
3. Revoke access for terminated users
4. Only enable temporary accounts for the period needed
5. Change password every 90 days
6. Limit 6 user connection attempts
7. Strong passwords
8. 2FA for users with administrative access

Requirement 8.1 through 8.4
Using PAM plugin

- Percona developed F/OSS PAM plugin
- Allows to use LDAP to satisfy requirements 8.1 through 8.4

- Install:

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

- A sample /etc/pam.d/mysqld file:

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

Requirement 8.1 through 8.4
Using PAM plugin example with UNIX authentication

- Install with:

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

- MySQL needs to be added to shadow group to read /etc/shadow
- A sample /etc/pam.d/mysqld file:

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

- Crate user:

```
mysql> CREATE USER 'newuser'@'localhost' IDENTIFIED WITH auth_pam;
```

Requirement 8.5
Change password once every 90 days

- Since MySQL 5.7.4, variable default_password_lifetime can be set to automatically expire passwords.

- Expired account running in restricted mode:

  mysql> show databases;

  ERROR 1820 (HY000): You must reset your password using ALTER USER statement before executing this statement.

- Password can be manually set to expired with command:

  mysql> ALTER USER 'testuser'@'localhost' PASSWORD EXPIRE;

Requirement 8.6
Limit User Attempts

- After six attempts, the user should be locked from the account. MySQL does not offer this natively, but has two plugins to help this:
  - INSTALL PLUGIN CONNECTION_CONTROL SONAME 'connection_control.so';
  - INSTALL PLUGIN CONNECTION_CONTROL_FAILED_LOGIN_ATTEMPTS SONAME 'connection_control.so';
- - CONNECTION_CONTROL_FAILED_LOGIN_ATTEMPTS: creates an IS table that shows failed connection attempts.
- - CONNECTION_CONTROL: checks incoming connections and adds a delay to server responses:
  - connection_control_failed_connections_threshold
  - connection_control_min_connection_delay

Requirement 8.7
Strong passwords with Validation Plugin

- Password length should be at lest 7 characters and contain both numeric and alphabetic characters.
- Default plugin MEDIUM policy has higher requirements than this
- This req can be implemented using Validation Plugin variables:

  validate_password_policy = ‘MEDIUM’
  validate_password_length = 7
  validate_password_number_count = 1
  validate_password_mixed_case_count = 1

Requirement 8.8
2FA for users with administrative access

- There is no native support for 2FA in MySQL, but is possible to implement using PAM plugin.
- We have a blogpost about this (search: 2fa site:percona.com/blog/)

- Enable PAM plugin
- Configure PAM for mysqld process by putting into /etc/pam.d/mysqld file
- Create user with auth_pam
- Install pam-google-authenticator and set up authentication
- Instruct PAM to use google authenticator
- Login!

Requirement 10:
Track and Monitor Access to Cardholder Data

Requirement 10
Track and Monitor Access to Cardholder Data

1. **Monitor user access to card information (Audit plugin)**
2. Secure audit trails so they cannot be altered.
3. Retain audit history for one year, having three months for immediate analysis.

Requirement 10
Monitor user access to card information

- MariaDB developed an Audit plugin comparable to MySQL Enterprise's one
  - Percona provided further enhancements to Audit plugin
- Provides monitoring and logging of connection and query activity

Release on PS from 5.5.37/5.6.17 and Enterprise on 5.5.28

- Information stored in the audit log file, containing NAME field, activity and TIMESTAMP.
- Audit plugin has filtering options: By user/command/database:
  - audit_log_include_accounts
  - audit_log_include_commands
  - audit_log_include_databases

Requirement 10
Audit Plugin Example

- To install PS audit plugin:
  mysql> INSTALL PLUGIN audit_log SONAME 'audit_log.so';
- 4 output formats: {XML OLD, XML NEW, JSON, CSV}, example in JSON:
  {"audit_record":{"name":"Query","record":"4707_2014-08-27T10:43:52","timestamp":"2014-08-27T10:44:19 UTC","command_class":"show_databases","connection_id":"37","status":0,"sqltext":"show databases","user":"root[root] @ localhost []","host":"localhost","os_user":"","ip":""}}
- Audit plugin can generate lot of output depending on server activity, so filtering can come in handy

Conclusions

Conclusions

- MySQL Covers most of the PCI/DSS requirements with built-in features or well supported plugins
  - Some requirements still need improvements to be easy to implement
- Starting from 5.7, many helpful features!
  - easier SSL, data at rest encryption, automatic password expiration
- 8.0 Brings even more
  - Roles, more encryption variables
- Percona brings even more… and for free
  - PAM and Audit plugin, additional encryption variables

References

References

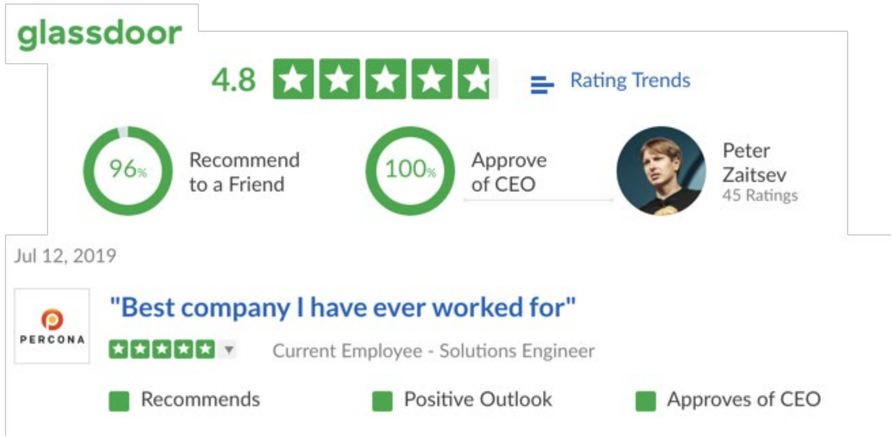
- Official PCI DSS site: https://www.pcisecuritystandards.org/
- Percona Server + plugins: https://www.percona.com/software/mysql-database/percona-server
- Percona blog: https://www.percona.com/blog/

We’re Hiring!

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

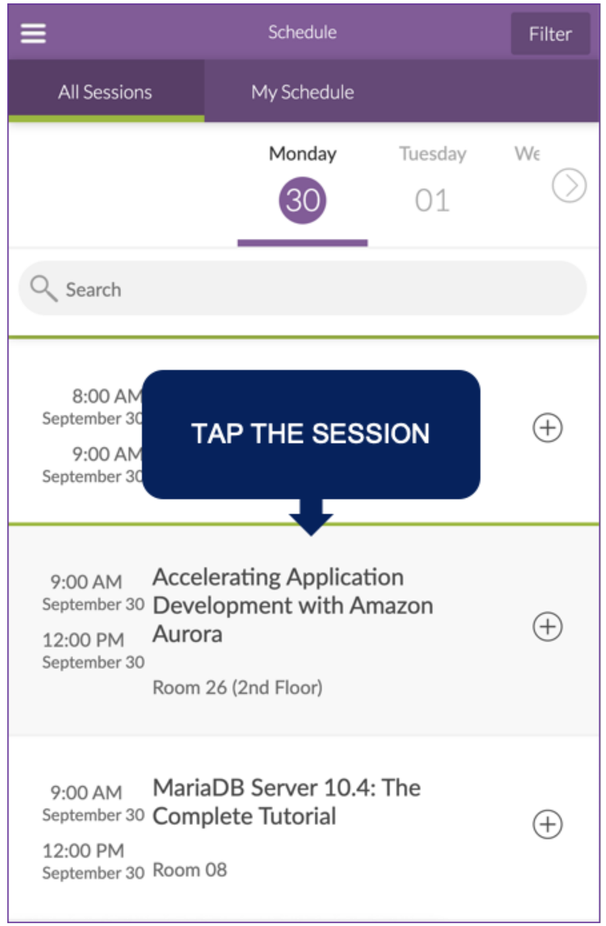
Our staff live in nearly 30 different countries around the world, and most work remotely from home.

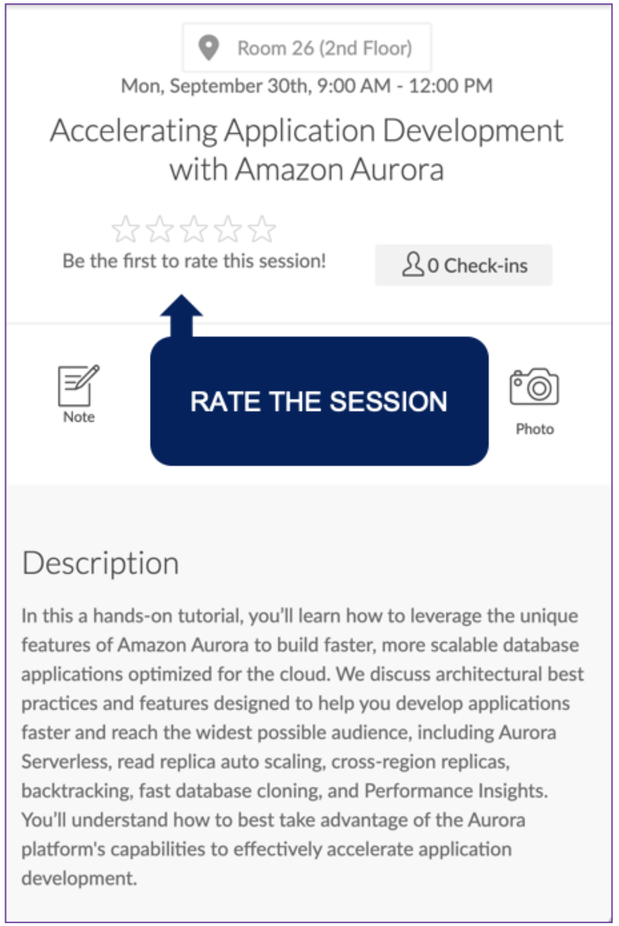
Discover what it means to have a Percona career with the smartest people in the database performance industries, solving the most challenging problems our customers come across.



Any Questions?

Rate My Session





Thank You