MySQL/MariaDB Server security essentials

Colin Charles, Chief Evangelist, Percona Inc. colin.charles@percona.com / byte@bytebot.net http://www.bytebot.net/blog/ | @bytebot on Twitter SCALE15X, Pasadena, California 4 March 2017



whoami



- 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

Historically...

- No password for the 'root' user
- There is a default 'test' database
- Find a password from application config files (wp-config.php, drupal's settings.php, etc.)
- Are your datadir permissions secure (/var/lib/mysql)?
- can you run strings mysql/user.MYD?



Can you view privileges to find a user with more access?

```
SELECT host, user, password from mysql.user;
 host
                                   password
               user
 localhost
                root
 sirius
               root
 127.0.0.1
               root
 ::1
                root
 localhost
 sirius
```



More things to think about

- Does replication connection have global permissions?
- If you can start/stop mysqld process, you can reset passwords
- Can you edit my.cnf? You can run a SQL file when mysqld starts with init-file



sql_mode

- 5.6 default = NO_ENGINE_SUBSTITUTION
- SQL_MODE = STRICT_ALL_TABLES, NO_ENGINE_SUBSTITUTION
- Keeps on improving, like deprecating NO_ZERO_DATE,
 NO_ZERO_IN_DATE (5.6.17) and making it part of strict mode



Signs of poor security

- 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



mysql_secure_installation

- Pretty basic to run, but many don't
- Remove anonymous users
- Remove test database
- Remove non-localhost root users
- Set a root password



Creating users

- The lazy way
 - CREATE USER 'foo'@'%';
- GRANT ALL ON *.* TO 'foo'@'%';
- The above gives you access to all tables in all databases + access from any external location
- ALL gives you a lot of privileges, including SHUTDOWN, SUPER, CHANGE MASTER, KILL, USAGE, etc.



SUPER privileges

- Can bypass a read_only server
- Can bypass init_connect
- Can disable binary logging
- Can dynamically change configuration
- Reached max_connections? Can still make one connection
- https://dev.mysql.com/doc/refman/5.6/en/privilegesprovided.html#priv_super
- SUPER Read Only: prohibit client updates for everyone



So... only give users what they need

- CREATE USER 'foo'@'localhost' IDENTIFIED by 'password';
- GRANT CREATE, SELECT, INSERT, UPDATE, DELETE on db.* to 'foo'@'localhost';



And when it comes to applications...

- Viewer? (read only access only)
- SELECT
- User? (read/write access)
- SELECT, INSERT, UPDATE, DELETE
- DBA? (provide access to the database)
- CREATE, DROP, etc.



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



Enable log-warnings

- Enable log_warnings=2
- Can keep track of access denied messages
- Worth noting there are differences here in MySQL & MariaDB
- https://dev.mysql.com/doc/refman/5.6/en/serveroptions.html#option_mysqld_log-warnings
- https://mariadb.com/kb/en/mariadb/server-system-variables/ #log_warnings



MySQL 5.6 improvements

- Password expiry
- ALTER USER 'foo'@'localhost' PASSWORD EXPIRE;
- https://dev.mysql.com/doc/refman/5.6/en/password-expirationsandbox-mode.html
- Password validation plugin
- VALIDATE_PASSWORD_STRENGTH()



MySQL 5.6 II

- 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

- 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



SSL

- You're using the cloud and you're using replication... you don't want this in cleartext
- Setup SSL (note: yaSSL vs OpenSSL can cause issues)
- https://dev.mysql.com/doc/refman/5.6/en/ssl-connections.html
- Worth noting 5.7 has a new tool: mysql_ssl_rsa_setup



Initialise data directory using mysqld now

- mysql_install_db is deprecated in 5.7
- mysqld itself handles instance initialisation
- mysqld —initialize
- mysqld —initialize-insecure



MariaDB passwords

- Password validation plugin (finally) exists now
- https://mariadb.com/kb/en/mariadb/development/mariadb-internalsdocumentation/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 plugins

What you do today

- MySQL stores accounts in the user table of the my mysql database
- CREATE USER 'foo'@'localhost' IDENTIFIED BY 'password';



```
select plugin_name, plugin_status from
information_schema.plugins where plugin_type='authentication';
              | plugin_status |
 plugin_name
 mysql_native_password | ACTIVE
 mysql_old_password | ACTIVE
2 rows in set (0.00 sec)
```



Subtle difference w/MariaDB & MySQL usernames

 Usernames in MariaDB > 5.5.31? 80 character limit (which you have to reload manually)

```
create user 'long12345678901234567890'@'localhost'
identified by 'pass';
Query OK, 0 rows affected (0.01 sec)
vs
ERROR 1470 (HY000): String
'long12345678901234567890' is too long for user
name (should be no longer than 16)
```



Installing plugins

- MariaDB: INSTALL SONAME 'auth_socket'
- MySQL: INSTALL PLUGIN auth_socket SONAME 'auth_socket.so'



auth_socket

- Authenticates against the Unix socket file
- Uses so_peercred socket option to obtain information about user running client
- CREATE USER 'foo'@'localhost' IDENTIFIED with auth_socket;
- Refuses connection of any other user but foo from connecting



sha256_password

- Default in 5.6, needs SSL-built MySQL (if using it, best to set it in my.cnf)
 - default-authentication-plugin=sha256_password
- Default SSL is yaSSL, but with OpenSSL you get RSA encryption
- client can transmit passwords to RSA server during connection
- There exists key paths for private/public keys
- Passwords never exposed as cleartext when connecting



PAM Authentication

- MySQL PAM
- Percona PAM (auth_pam & auth_pam_compat)
- MariaDB PAM (pam)



Let's get somethings out of the way

- PAM = Pluggable Authentication Module
- Use pam_Idap to to authenticate credentials against LDAP server configure /etc/pam_Idap.conf (you also obviously need /etc/ Idap.conf)
- Simplest way is of course /etc/shadow auth



Percona Server

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



For MySQL compatibility

 Just use -pam-use-cleartext-plugin for MySQL to use mysql_cleartext_password instead of dialog plugin



Possible errors

- Connectors don't support it:
- Client does not support authentication protocol requested by server; consider upgrading MySQL client.
- You may have to re-compile connector using libmysqlclient to have said support



Kerberos/GSSAPI

- Every participant in authenticated communication is known as a 'principal' (w/unique name)
- Principals belong to administrative groups called realms. Kerberos Distribution Centre maintains a database of principal in realm + associated secret keys
- Client requests a ticket from KDC for access to a specific asset.
 KDC uses the client's secret and the server's secret to construct the ticket which allows the client and server to mutually authenticate each other, while keeping the secrets hidden.



MariaDB Kerberos plugin

- User principals: <username>@<KERBEROS REALM>
- CREATE USER 'byte' IDENTIFIED VIA kerberos AS 'byte/mariadb@lp';
- so that is <username>/<instance>@<KERBEROS REALM>
- Store Service Principal Name (SPN) is an option in a config file



Works where?

- GSSAPI-based Kerberos widely used & supported on Linux
- Windows supports SSPI authentication and the plugin supports it
- Comes with MariaDB Server 10.1



5.7 mysql_no_login

- mysql_no_login prevents all client connections to an account that uses it
- https://dev.mysql.com/doc/refman/5.7/en/mysql-no-loginplugin.html





SQL Error Logging Plugin

- Log errors sent to clients in a log file that can be analysed later. Log file can be rotated (recommended)
- a MYSQL_AUDIT_PLUGIN
- •install plugin SQL_ERROR_LOG soname
 'sql_errlog.so';



Audit Plugin

- Log server activity who connects to the server, what queries run, what tables touched - rotating log file or syslogd
- MariaDB has extended the audit API, so user filtering is possible
- a MYSQL_AUDIT_PLUGIN
- •INSTALL PLUGIN server_audit SONAME
- 'server_audit.so';



Roles

Bundles users together, with similar privileges - follows the SQL standard

```
CREATE ROLE audit_bean_counters;
GRANT SELECT ON accounts.* to audit_bean_counters;
GRANT audit_bean_counters to ceo;
```





Encryption

- Encryption: tablespace and table level encryption with support for rolling keys using the AES algorithm
- table encryption PAGE_ENCRYPTION=1
- tablespace encryption encrypts everything including log files
- New 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/



Encryption II

- The key file contains encryption keys identifiers (32-bit numbers) and hex-encoded encryption keys (128-256 bit keys), separated by a semicolon.
- don't forget to create keys!
- eg. openssl enc -aes-256-cbc -md sha1 -k secret -in keys.txt -out keys.enc



my.cnf config

```
[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
```



Encryption III

```
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 IV

- Tablespace encryption (Google)
- · again, you need to pick an encryption algorithm
- specify what to encrypt: innodb-encrypt-tables, aria, aria-encrypt-tables, encrypt-tmp-disk-tables, innodb-encrypt-log
- don't forget key rotation:
 - innodb-encryption-threads=4
 - innodb-encryption-rotate-key-age=1800



Encryption V

- we also have tablespace scrubbing
- background process that regularly scans through the tables and upgrades the encryption keys
- scrubbing works for tablespaces and logs
- encrypt-tmp-files
- –encrypt-binlog



Encryption VI

- /etc/my.cnf.d/enable_encryption.preset
- Consider using Eperi Gateway for Databases
- MariaDB Enterprise has a plugin for Amazon Key Management Server (KMS)
- 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 VII

- 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)



Preventing SQL Injections

- MySQL Enterprise Firewall (\$\$\$)
- http://mysqlserverteam.com/new-mysql-enterprise-firewall-preventsql-injection-attacks/
- MaxScale Database Firewall filter
- ProxySQL
- http://www.proxysql.com/



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
- Restricting traffic to your database server (open ports)
- Run a software firewall
- iptables, ufw



Database Ransom Attacks Have Now Hit MySQL Servers

By Catalin Cimpanu Tebruary 25, 2017 O4:55 AM



After the ransacking of MongoDB, ElasticSearch, Hadoop, CouchDB, and Cassandra servers, attackers are now hijacking hundreds of MySQL databases, deleting their content, and leaving a ransom note behind asking for a 0.2 Bitcoin (\$235) payment.

According to breach detection firm GuardiCore, the attacks are happening via brute-force attacks on Internet-exposed MySQL servers, and there's plenty of those laying around since MySQL is one of today's most popular database systems.



Resources

- oak-security-audit
- http://code.openark.org/oak/oak-security-audit
- Encrypting MySQL Data at Google Jeremy Cole & Jonas Oreland
 - http://bit.ly/google_innodb_encryption





April 24th-27th, 2017 Santa Clara, California

Open Source Database Conference
MySQL, MariaDB, MongoDB, PostgreSQL
Time Series, Monitoring, RocksDB and More

Use Discount Code "SCALE15" for 15% off registration

perconalive.com



Colin Charles

colin.charles@percona.com / byte@bytebot.net

http://bytebot.net/blog | @bytebot on twitter

slides: slideshare.net/bytebot