MySQL Security: What's New in MySQL 5.7 & Best Practices

Georgi "Joro" Kodinov
Team Lead
MySQL Server General Team

ORACLE

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

About Me

Former banking IT manager

Veteran software developer

Leading the MySQL server general development team

With MySQL since 2006

Working out of Plovdiv, Bulgaria

@gkodinov, **georgi.kodinov@oracle.com**

ORACLE

Agenda

- Overview of the New Features
- How to Harden your MySQL Installation
- Questions and Answers

So What's New ?

New Security Features at a Glance

- Secure by Default
- Better Account Management
- Refactoring and Housekeeping
- MySQL Enterprise Firewall
- MySQL Enterprise Encryption

ORACLE

Secure By Default

Why Secure by Default ?

- More and more sensitive information stored in computers
- People increasingly dependent on information
- Helps new users avoid common mistakes
- Decreases MySQL attack surface
- Security compromises are explicit acts by the DBAs
- Low usability impact if applied carefully

Secure By Default

Highlights

- Random account passwords on install
- Deploy without tests and demos
- No anonymous accounts
- Limit the scope of the FILE privilege
- Stricter permissions on installation files

Enhanced SSL Support

Highlights

- The server comes fully equipped for SSL connections
  - Generate self signed certificates on site
- Libmysql based clients will attempt SSL by default
- Client option to enforce SSL
- Server option to require secure transport for all connections
  - SSL, shared memory, UNIX sockets

Better User Account Management

Comfortable User Account Management

Highlights

- Fully functional ALTER USER
- IF [NOT] EXISTS clause to user account statements
- Temporarily disable user accounts
- Authentication plugin that disables logins
- Time-based password expiration policy
- “Offline” server mode
- User account names now 32 characters long

Refactoring and Housekeeping Work

Remove Support for “old”, Pre-4.1 Passwords

What’s new in Refactoring ?

- Known to be insecure since mysql-4.1 !
- Responsible for a lot of “spaghetti code”
- Finally out in 5.7 !
- **Incompatible changes in how empty string for mysql.user.plugin column is handled on existing accounts !**

Threat Passwords as Any Other Authentication Data

- Why ?
  - All authentication plugins must be equal
  - Passwords are just one form of authentication
- How ?
  - Merge **mysql.user.password** into **mysql.user.authentication_string**
  - Extend the authentication API to support converting between storage and clear text formats
  - **Clean up stray global plugin related variables and functions**
    - PASSWORD() @@old_passwords

Deprecate ENCODE()/DECODE()

What's new in Refactoring ?

- Old insecure, home grown algorithms
- Replaced by AES based functions

Refactor the ACL Code

What's new in Refactoring ?

- Isolated the ACL code into a sub-directory
- Split the huge sql_acl.cc into smaller logical groups of code

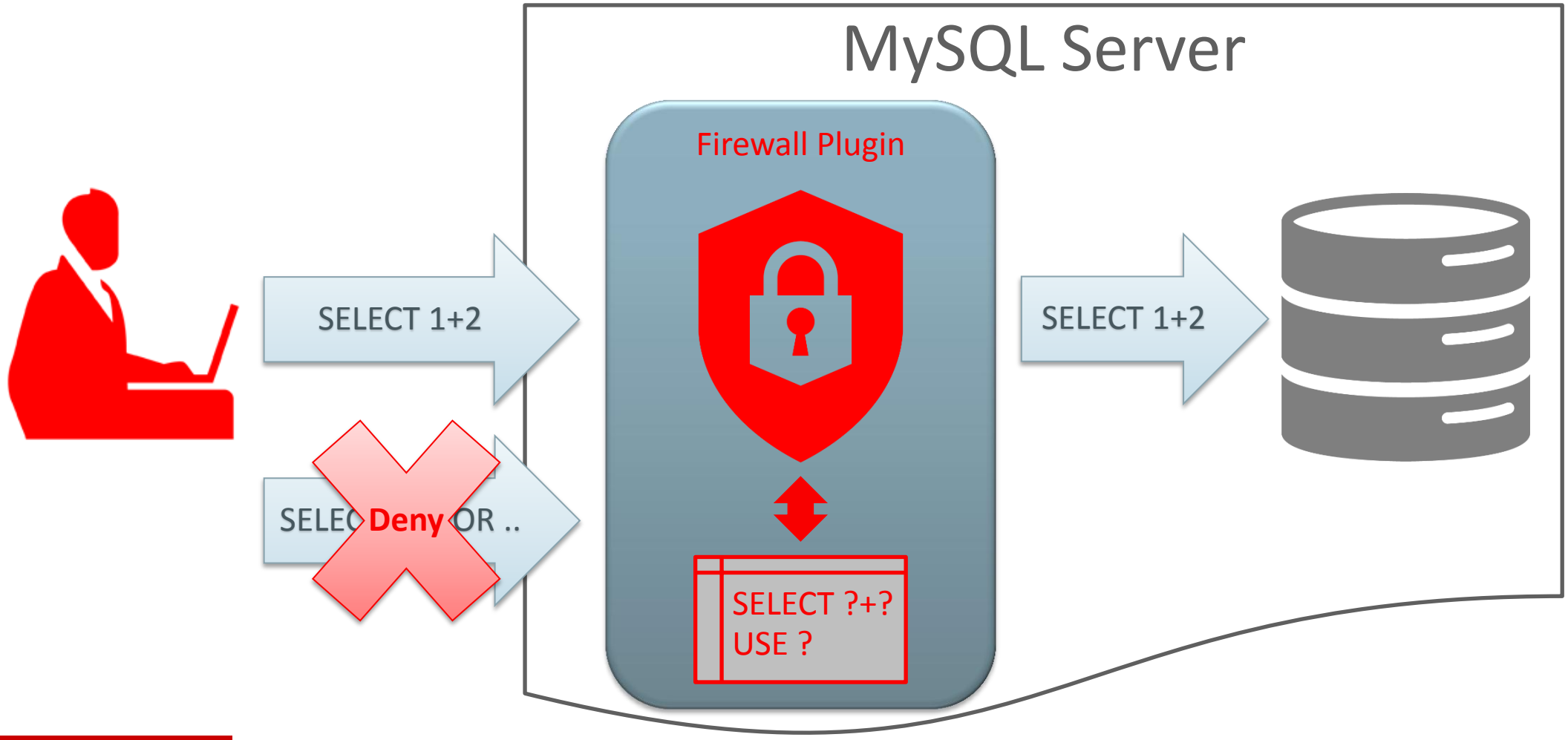
Remove Server Package's Perl Dependencies

What's new in Refactoring ?

- mysql_secure_install converted to a C program
- mysql_install_db converted to a C program
- mysql_upgrade now doesn't need to call external binaries
- Why is this relevant to security ?
  - No longer need to pass passwords to the external utilities
  - No worries that the utilities may use different configuration files
  - Allowed extra hardening: e.g. remove the multiple "root" user accounts
  - Allows unattended installation

MySQL Enterprise Firewall

MySQL Enterprise Firewall Operation



MySQL Enterprise Firewall Anatomy

Engine

- Compares incoming queries with the allowed list
- Works on normalized statements
- Multiple modes
- Audit log plugin

Statements Cache

- Entirely in memory
- Initialized from disk
- Content visible through INFORMATION_SCHEMA

MySQL Enterprise Firewall Benefits

- Helps preventing SQL Injection
- Can learn to pass-through wide variety of apps
- No need to re-parse
- Easy to install and uninstall
- Works on SQL Statements

MySQL Enterprise Encryption

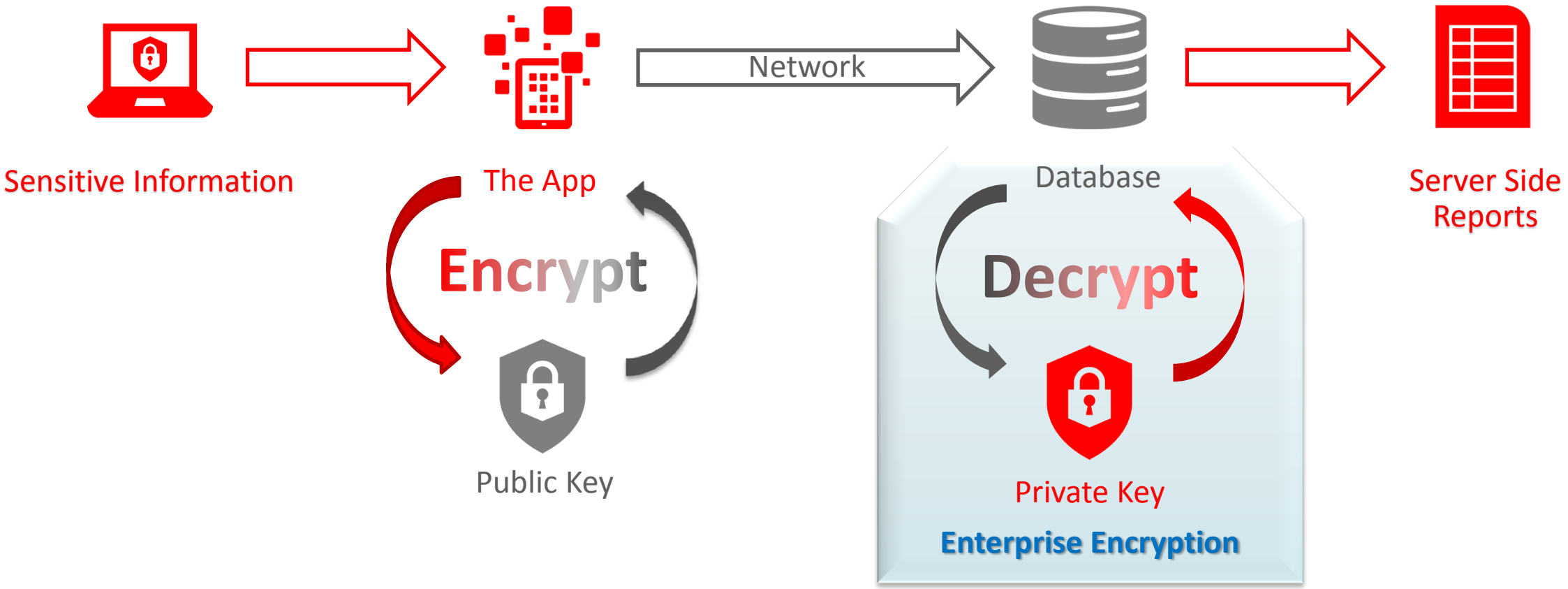
What is MySQL Enterprise Encryption ?

At a glance

- A plugin, interfacing the OpenSSL library
- Allows generation and handling of RSA, DSA and DH key pairs
  - **SET @priv = CREATE_ASYMMETRIC_PRIV_KEY('RSA', 1024);**
  - **SET @pub = CREATE_ASYMMETRIC_PUB_KEY('RSA', @priv);**
- Allows encryption, signing and cyrptographic hashes
  - **SET @digest = CREATE_DIGEST('SHA512', 'cleartext');**
  - **SET @sig = ASYMMETRIC_ENCRYPT('RSA', @digest, @priv);**
- Key generators use standard PEM format (compatible with external tools)

Simple Example: Handling Sensitive Information

Using MySQL Enterprise Encryption



Overview: MySQL Enterprise Edition Security Features

- MySQL Enterprise Firewall
  - Block SQL Injection Attacks
  - Intrusion Detection
- MySQL Enterprise Encryption
  - Public/Private Key Cryptography
  - Asymmetric Encryption
  - Digital Signatures, Data Validation
- MySQL Enterprise Authentication
  - External Authentication Modules
    - Microsoft AD, Linux PAMs
- MySQL Enterprise Audit
  - User Activity Auditing, Regulatory Compliance
- MySQL Enterprise Monitor
  - Changes in Database Configurations, Users Permissions, Database Schema, Passwords
- MySQL Enterprise Backup
  - Securing Backups, AES 256 encryption

ORACLE

Immediate Steps

Post Server Installation

Hardening your MySQL installation

- Run mysql_secure_installation ! **Now !**
- Review and restrict the network interfaces that the server listens on
- Generate SSL keys and make sure the server can “talk” SSL
- Enable query logging. Create a log backup policy.
- Remove extra user accounts and privileges
- Remove unneeded files and packages
- Schedule regular backups !

Post Application(s) Installation

Hardening your MySQL installation

- Remove extra user accounts. Restrict the remaining ones
- Review and maximally restrict the grants
- Make sure the user accounts authenticate using a reliable method
- Clean up extra temp files
- Make sure backups are still on and cover the new objects
- Remove unneeded files and packages
- Audit the server configuration for changes. Revert the bogus ones

Daily MySQL Use

Hardening your MySQL installation

- Keep your installation up to date
- Monitor your server logs. Set alerts for “unusual” patterns.
- Monitor security related stats. Set alerts for “unusual” patterns.
- Monitor the server configuration.
- Monitor and verify the backups and their integrity
- Regularly probe your “defenses” by trying bad things on purpose
- Perform regular emergency drills
- Set procedures on maintaining your user account base

More Security

Harden your MySQL Server Instance

Additional steps

- Consider turning off TCP/IP if your setup allows it
- Use and **enforce** SSL if you need TCP/IP
  - Even self-signed will do. Part of PKI is better
- Use SSL certificate requirements for users
  - GRANT … TO …. REQUIRE [CIPHER | ISSUER | SUBJECT] …
- Be careful with your directories
  - tmpdir, datadir, secure-file-priv, plugin-dir

Harden your MySQL Server Instance

Even more steps

- Monitor and keep the logs
  - Consider using an auditing plugin
  - put extra protection on sensitive tables: custom logging triggers etc
- Consider using external authentication
  - PAM, LDAP, windows domain
- Harden your password policy
  - MySQL has a plugin for that !
- Use login paths for your scripts

Harden your MySQL Server Instance

Useful parameters to set

Parameter	Recommended Value
secure_file_priv	Designated directory
symbolic_links	Boolean NO
default-storage-engine	InnoDB
general-log	Boolean ON
log-raw	Default : OFF
skip-networking	ON, if you can afford it.
ssl options	Set to valid values

Harden your MySQL Server Instance

Useful parameters to set

Parameter	Recommended Value
plugin-dir	Designated read-only directory
chroot	Designated directory, if you can afford it
core-file	OFF
des-key-file	File with DES keys
read_only	ON for slaves !
sha256_password RSA key	RSA public private keys if can’t use SSL
tmpdir	Designated directory out of secure-file-priv

ORACLE

Questions and Answers


ORACLE®