MySQL Security: Best Practices

ORACLE

MySQL

89%

of Organizations Experienced Data Breaches, According to New Ponemon Report

Source: *Sixth Annual Benchmark Study on Privacy & Security of Healthcare Data*, conducted by Ponemon Institute

66%

of the largest businesses in the UK have suffered a cyberattack or data breach within the past twelve months

Source: *UK government's Cyber Security Breaches Survey 2016*

25%

experience a repeated breach at least one a month

Source: *UK government's Cyber Security Breaches Survey 2016*

Mega Breaches

429 Million identities exposed in 2015.

75%

Web sites with vulnerabilities. 15% of all websites had a critical vulnerability.

9

In 2015, a record of nine mega-breaches were reported.

One worlds largest 191M.

(Mega-breach = more than 10 million records.)

Mobile Vulnerabilities on the rise – up 214%

Infection by SQL Injection still strong.

Malware attacks on databases

Source: Internet Security Threat Report 2016, Symantec

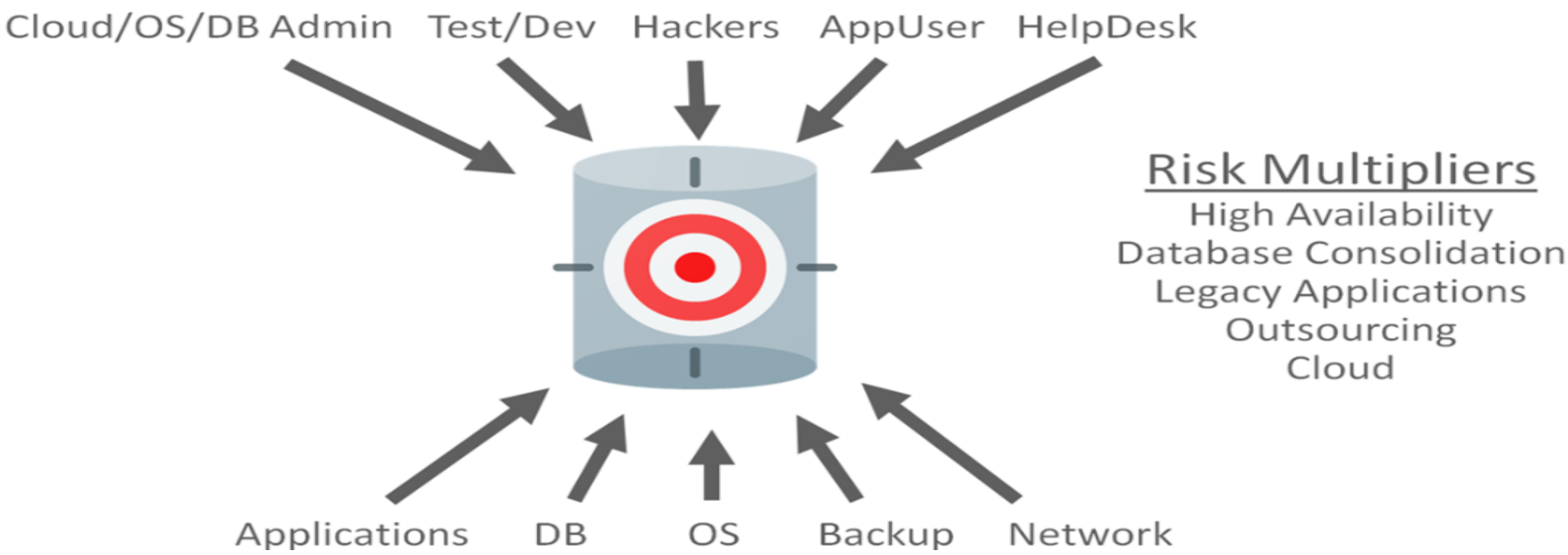
Database Vulnerabilities

- Poor Configurations
  - Set controls and change default setting
- Over Privileged Accounts
  - Privilege Policies
- Weak Access Control
  - Dedicated Administrative Accounts
- Weak Authentication
  - Strong Password Enforcement
- Weak Auditing
  - Compliance & Audit Policies
- Lack of Encryption
  - Data, Backup, & Network Encryption
- Proper Credential & Key Management
  - Use mysql_config_editor , Key Vaults
- Unsecured Backups
  - Encrypted Backups
- No Monitoring
  - Security Monitoring, Users, Objects
- Poorly Coded Applications
  - Database Firewall

Database Attacks

- SQL Injection
  - Prevention: DB Firewall, White List, Input Validation
- Buffer Overflow
  - Prevention: Frequently apply Database Software updates, DB Firewall, White List, Input Validation
- Brute Force Attack
  - Prevention: lock out accounts after a defined number of incorrect attempts.
- Network Eavesdropping
  - Prevention: Require SSL/TLS for all Connections and Transport
- Malware
  - Prevention: Tight Access Controls, Limited Network IP access, Change default settings, Encryption

Attack Vectors and Targets for Databases



Database Malicious Actions

- Information Disclosure: Obtain credit card and other personal information
  - Defense: Encryption – Data and Network, Tighter Access Controls
- Denial of Service: Run resource intensive queries
  - Defense: Resource Usage Limits – Set various limits – Max Connections, Sessions, Timeouts, …
- Elevation of Privilege: Retrieve and use administrator credentials
  - Defense: Stronger authentication, Access Controls, Auditing
- Spoofing: Retrieve and use other credentials
  - Defense: Stronger account and password policies
- Tampering: Change data in the database, Delete transaction records
  - Defense: Tighter Access Controls, Auditing, Monitoring, Backups

Disclaimer

The information in this document may not be construed or used as legal advice about the content, interpretation or application of any law, regulation or regulatory guideline. Customers and prospective customers must seek their own legal counsel to understand the applicability of any law or regulation on their processing of personal data, including through the use of any vendor's products or services.

Regulatory Compliance

- Regulations
  - PCI – DSS: Payment Card Data
  - HIPAA: Privacy of Health Data
  - Sarbanes Oxley, GLBA, The USA Patriot Act:
    Financial Data, NPI "personally identifiable financial information"
  - FERPA – Student Data
  - EU General Data Protection Directive: Protection of Personal Data (GDPR)
  - Data Protection Act (UK): Protection of Personal Data
- Requirements
  - Continuous Monitoring (Users, Schema, Backups, etc)
  - Data Protection (Encryption, Privilege Management, etc.)
  - Data Retention (Backups, User Activity, etc.)
  - Data Auditing (User activity, etc.)

PCI Security Standards Council

HIPAA
Health Insurance Portability and Accountability Act

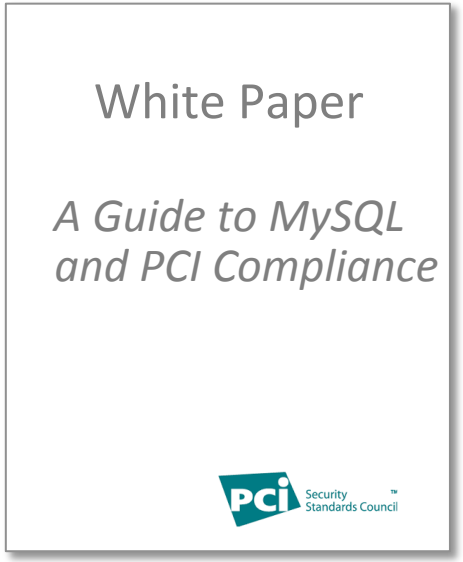
Sarbanes-Oxley
Financial and Accounting Disclosure Information

European Commission
Justice

Data Protection Act 1998

PCI-DSS

- Requirement 2: Secure Configurations, Security Settings & Patching
  - Not Using Vendor Default Passwords and Security Settings
- Requirement 3: Protecting Cardholder Data – Strong Cryptography
  - Protect Stored Cardholder Data
  - Protect Encryption Keys
- Requirement 6: Up to Date Patching and Secure Systems
  - Develop and Maintain Secure Systems and Applications
- Requirement 7: User Access and Authorization
  - Restrict Access to Cardholder Data by Need to Know
- Requirement 8: Identity and Access Management
  - Identify and Authenticate Access to System Components
- Requirement 10: Monitoring, Tracking and Auditing
  - Track and Monitor Access to Cardholder Data



HIPAA / HITECH

- Access Controls
  - Access only to those persons or software programs that have been granted access rights
  - Unique User Identification, Emergency Access Procedure, Automatic Logoff, Encryption and Decryption
- Authentication
  - Verify that a person or entity seeking electronic health information is the one claimed
- Integrity
  - Protect electronic protected health information from improper alteration or destruction
- Transmission Security
  - Guard against unauthorized access that is being transmitted over a network
- Encryption
  - Encrypt electronic protected health information. Separation of encryption keys from the data they protect.
- Audit Control
  - Record and examine activity that contain or use electronic protected health information

Sarbanes Oxley

Sarbanes-Oxley
Financial and Accounting Disclosure Information

- Accurate and factual business and financial reports
  - Verify that the records protected from tampering and modification
- Protect data accuracy and integrity
  - Minimal permissions on data for each employee
  - Deny any privileges above minimal
  - Audit all activity

EU General Data Protection Regulation (GDPR)

- Data privacy as a fundamental right
- Data protection responsibilities, baselines, principles
- Enforment Powers
- Published May 2016, Enforceable by May 2018
- Fines up to greater of 4% of Global revenue or 20M Euros.

Focus on

- Assessment – Processes, Profiles, Data Sensitivity, Ricks
- Prevention – Encryption, Anonymization, Access Controls, Separation of Duties
- Detect – Auditing, Activity monitoring, Alerting, Reporting

MySQL Enterprise Features and GDPR

- Assess Risks (Articles 35, 90, 91)
  - MySQL Enterprise Monitor
  - MySQL Workbench Data Modeling tool
- Prevent Attacks (Articles 32, 83, 28, 26, 5, 20, 27, 30, 64)
  - MySQL Enterprise TDE Transparent Data Encryption)
  - MySQL Enterprise Firewall
- Detect (Articles 30, 82, 33)
  - MySQL Enterprise Audit
  - MySQL Enterprise Firewall

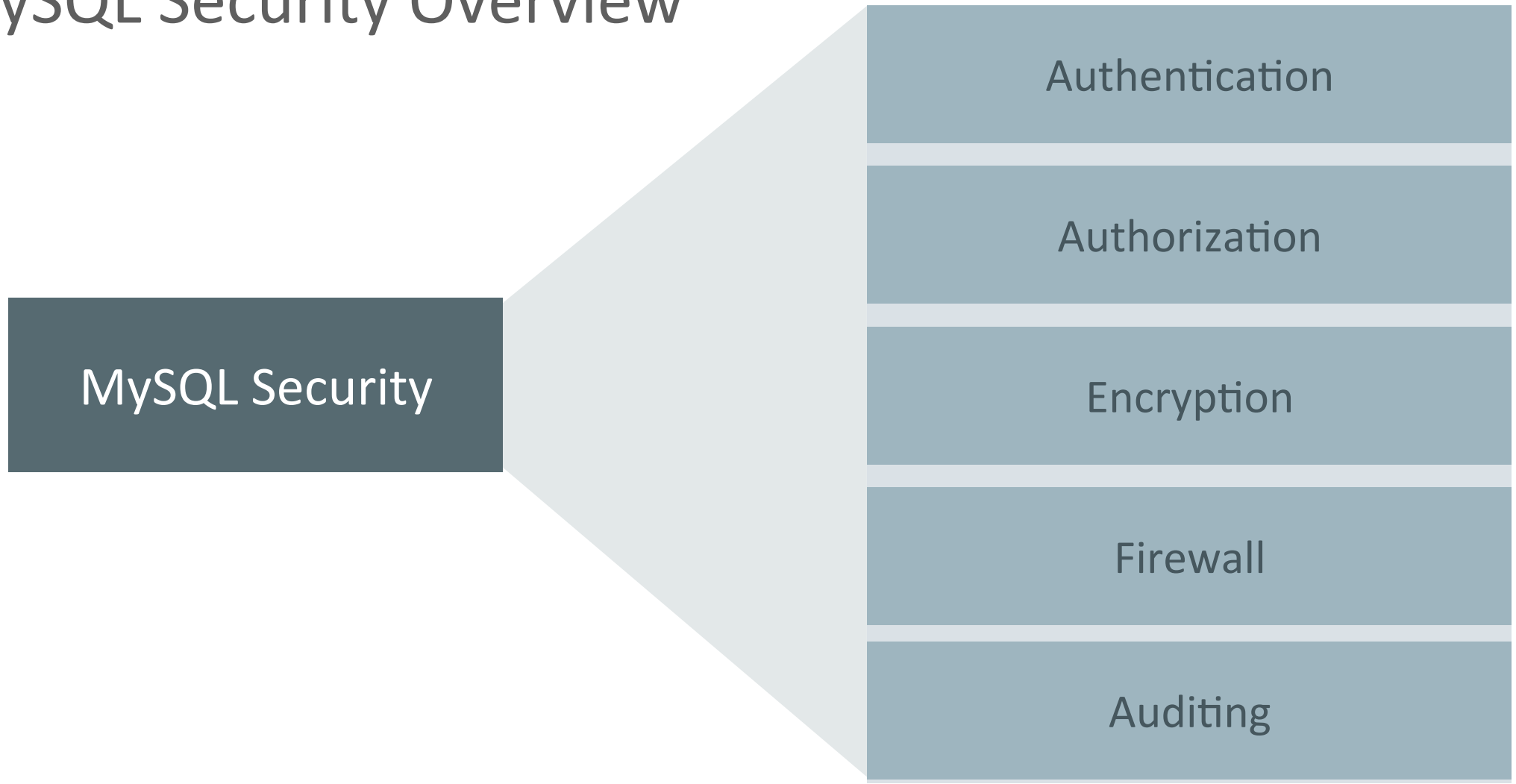
Data Protection Act – UK 1998

- Personal data shall be processed fairly and lawfully
- Personal data shall be obtained only for one or more specified and lawful purposes
- Personal data shall be adequate, relevant and not excessive
- Personal data shall be accurate and, where necessary, kept up to date
- Personal data processed for any purpose shall not be kept for longer than is necessary
- Personal data shall be processed in accordance with the rights of data subjects
- Measures shall be taken against unauthorized or unlawful processing of personal data and against accidental loss or destruction of, or damage to, personal data.
- Personal data shall not be transferred to a country or territory outside the European Economic Area

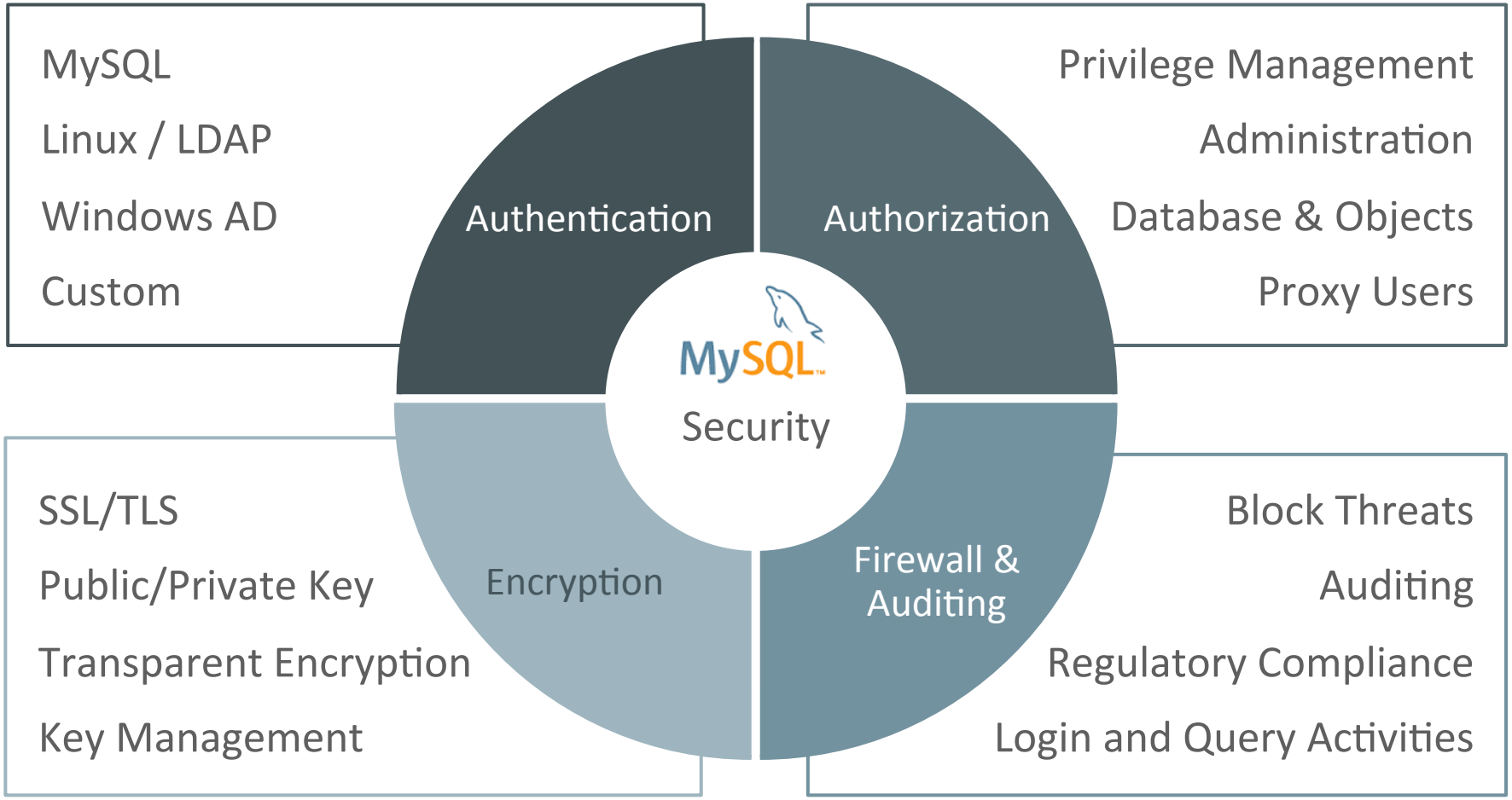
DBA Responsibilities

- Ensure only users who should get access, can get access
- Limit what users and applications can do
- Limit from where users and applications can access data
- Watch what is happening, and when it happened
- Make sure to back things up securely
- Minimize attack surface
- Ensure encryption keys are protected and managed

MySQL Security Overview

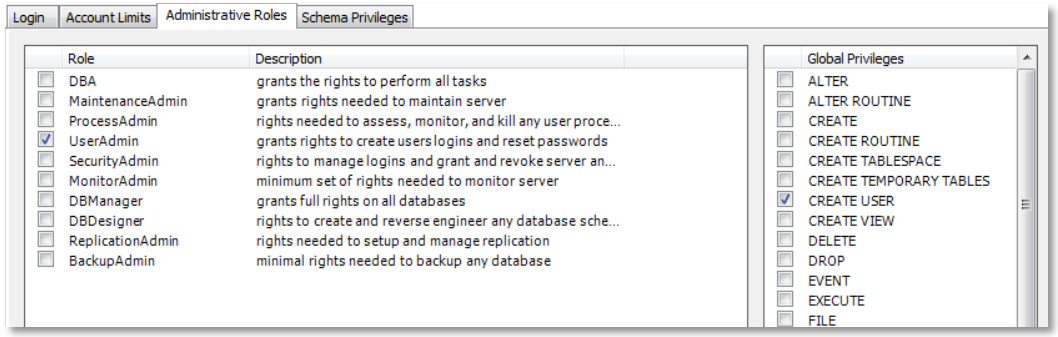


MySQL Security Overview



MySQL Authorization

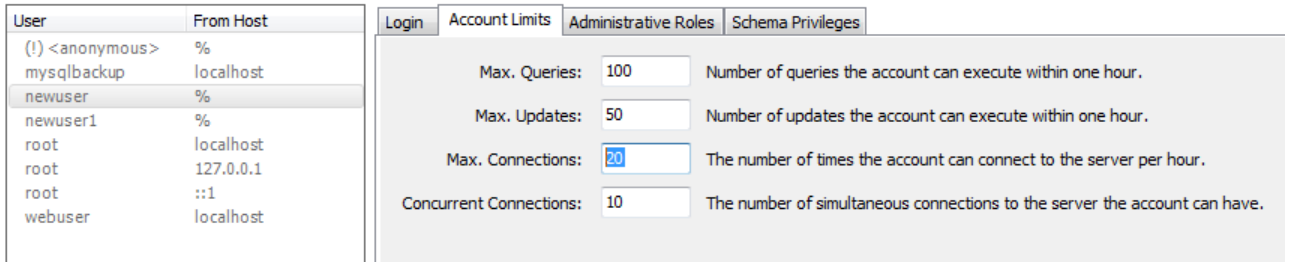
- Administrative Privileges
- Database Privileges
- Session Limits and Object Privileges
- Fine grained controls over user privileges
  - Creating, altering and deleting databases
  - Creating, altering and deleting tables
  - Execute INSERT, SELECT, UPDATE, DELETE queries
  - Create, execute, or delete stored procedures and with what rights
  - Create or delete indexes



Security Privilege Management in MySQL Workbench

MySQL Privilege Management

- user: user accounts, global privileges columns
- db: database-level privileges
- tables_priv: Contains table-level privileges
- columns_priv: Contains column-level privileges
- procs_priv: Contains stored procedure and function privileges
- proxies_priv: Contains proxy-user



MySQL Privilege Management Grant Tables

user

- User Accounts
- Global Privileges

db

- Database Level Privileges
- Database, Tables, Objects
- User and host

tables_priv

- Table level privileges
- Table and columns

columns_priv

- Specific columns

procs_priv

- Stored Procedures
- Functions
- Single function privilege

proxies_priv

- Proxy Users
- Proxy Privileges

MySQL Privilege Management

- Continuous assessment
  - Configuration
  - Users
  - Permissions and Rights
- Audit & Review activity
  - Who – does activity match expectation
  - What – is this it limited as expected
  - When – acts often are at odd / off peak times
  - Where – Connections should be from expected hosts
- MySQL has simple to use controls and privileges to set secure limits

MySQL Authentication

- Built in Authentication
  - `user` table stores users and encrypted passwords
- External Authentication with MySQL Enterprise Authentication
  - Microsoft Active Directory
  - Linux PAMs (Pluggable Authentication Modules)
    - Support LDAP and more
- X.509
  - Server authenticates client via certificates
- MySQL Native, SHA 256 Password plugin
  - Native uses SHA1 or plugin with SHA-256 hashing and per user salting for user account passwords.

MySQL Password Policies

- Accounts without Passwords
  - Assign passwords to all accounts to prevent unauthorized use
- Password Validation Plugin
  - Enforce Strong Passwords
- Password Expiration/Rotation
  - Require users to reset their password
- Account lockout (in v. 5.7)
- Password Retry Rules (in v. 5.7.16+)

MySQL Encryption

- SSL/TLS Encryption
  - Between MySQL clients and Server
  - Replication: Between Master & Slave
- Data Encryption
  - AES Encrypt/Decrypt
- MySQL Enterprise TDE
  - Transparent Data Encryption
  - Key Management (KMIP)
- MySQL Enterprise Encryption
  - Asymmetric Encrypt/Decrypt
  - Generate Public Key and Private Keys
  - Derive Session Keys
  - Digital Signatures
- MySQL Enterprise Backup
  - AES Encrypt/Decrypt

SSL/TLS

- Encrypted connections
  - Between MySQL Client and Server
  - Replication: Between Master & Slave
- MySQL enables encryption on a per-connection basis
  - Identity verification using the X509 standard
- Specify the appropriate SSL certificate and key files
- Will work with trusted CAs (Certificate Authorities)
- Supports CRLs – Certificate Revocation Lists

Database Firewall

- SQL Injection Attacks
  - #1 Web Application Vulnerability
  - 77% of Web Sites had vulnerabilities
- MySQL Enterprise Firewall
  - Monitor database statements in real-time
  - Automatic White List “rules” generation for any application
  - Block SQL Injection Attacks
  - Intrusion Detection System

Database Auditing

- Auditing for Security & Compliance
  - FIPS, HIPAA, PCI-DSS, SOX, DISA STIG, …
- MySQL built-in logging infrastructure:
  - general log, error log
- MySQL Enterprise Audit
  - Granularity made for auditing
  - Can be modified live
  - Contains additional details
  - Compatible with Oracle Audit Vault.

MySQL Database Hardening

Installation

- Mysql_secure_installation
- Keep MySQL up to date
  - MySQL Installer for Windows
  - Yum/Apt Repository

Configuration

- Firewall
- Auditing and Logging
- Limit Network Access
- Monitor changes

User Management

- Remove Extra Accounts
- Grant Minimal Privileges
- Audit users and privileges

Passwords

- Strong Password Policy
- Hashing, Expiration
- Password Validation Plugin

Encryption

- SSL/TLS for Secure Connections
- Data Encryption (AES, RSA)
- TDE

Backups

- Monitor Backups
- Encrypt Backups

MySQL 5.7 Linux Packages - Security Improvements

- Test/Demo database has been removed
  - Now in separate packages
- Anonymous account creation is removed.
- Creation of single root account – local host only
- Default installation ensures encrypted communication by default
  - Automatic generation of SSL/RSA Certs/Keys
    - For EE : At server startup if options Certs/Keys were not set
    - For CE : Through new mysql_ssl_rsa_setup utility
- Automatic detection of SSL Certs/Keys
- Client attempts secure TLS connection by default
- Compile time restriction over location used for data import/export operations
- Ensures location has restricted access
- Only mysql user and group
- Supports disabling data import/export
  - Set secure-file-priv to empty string

MySQL Installer for Windows includes various Security Setup and Hardening Steps

MySQL Database Hardening: Installation

- MySQL_Secure_Installation / MySQL Installer for Windows
  - Set a **strong** password for root account
  - Remove root accounts that are accessible from outside the local host
  - Remove anonymous-user accounts
  - Remove the test database
    - Which by default can be accessed by all users
    - Including Anonymous Users
- Keep MySQL up to date
  - Repos – YUM/APT/SUSE
  - MySQL Installer for Windows

Software Updates - Database and OS Maintenance

- Maintaining security requires keeping Operating System and MySQL security patches up to date.
  - May require a restart (mysql or operating system) to take effect.
- To enable seamless upgrades consider MySQL Replication
  - Allows for changes to be performed in a rolling fashion
    - Best practice to upgrade slaves first
  - MySQL 5.6 and above supports GTID-based replication
    - Provides for simple rolling upgrades
- Follow OS vendor specific hardening Guidelines
  - For example
    - http://www.oracle.com/technetwork/articles/servers-storage-admin/tips-harden-oracle-linux-1695888.html

MySQL Database Hardening: Configuration

- Audit Activity
  - *Use Enterprise Audit*
  - *Alt. Transiently enable Query Logging*
  - Monitor and Inspect regularly
- Disable or Limit Remote Access
  - If local “skip-networking” or bind-address=127.0.0.1
  - If Remote access then limit hosts/IP
- Consider changing default port
- Change root username
- Disable unauthorized reading from local files
  - Disable LOAD DATA LOCAL INFILE
- Run MySQL on non default port
  - More difficult to find database
- Limit MySQL OS User
- Ensure secure-auth is enabled
- Configure External Key Management

MySQL Database Hardening: Best Practices

Parameter	Recommended Value	Why
Secure_file_priv	A Designated Leaf directory for data loads	Only allows file to be loaded from a specific location. Limits use of MySQL to get data from across the OS
Symbolic_links	Boolean – NO	Prevents redirection into less secure filesystem directories
Default-storage_engine	InnoDB	Ensures transactions commits, ???
General-log	Boolean – OFF	Should only be used for debugging – off otherwise
Log-raw	Default - OFF	Should only be used for debugging – off otherwise
Skip-networking or bind-address	ON 127.0.0.1	If all local, then block network connections or limit to the local host.
SSL options	Set valid values	Should encrypt network communication

MySQL Database Hardening: Password Policies

- Enforce Strong Password Policies
- Password Hashing
- Password Expiration
- Password Validation Plugin
- Authentication Plugin
  - Inherits the password policies from the component
  - LDAP, Windows Active Directory, etc.
- Disable accounts when not in use
  - Account lockout (5.7+)

MySQL Database Hardening: Encryption

- Encryted Communication and More
- SSL/TLS encrypted for transport
- X.509 adds additional “Factor” – something you have – in addition to username/password or other authentication
  - Assures the client is validated – thus more likely trusted
- Use database and application level encryption of highly sensitive data
- User database or application functions to mask or de-identify data
- Consider Public Keys for Applications that encrypt only
- Use Transparent Data Encryption with external Key Management Services

MySQL Database Hardening: Backups

- Backups are Business Critical
  - Used to restore after attack
  - Migrate, move or clone server
  - Part of Audit Trail
- Regularly Scheduled Backups
- Monitor Backups
- Encrypt Backups

Applications and Credentials - Best Practices

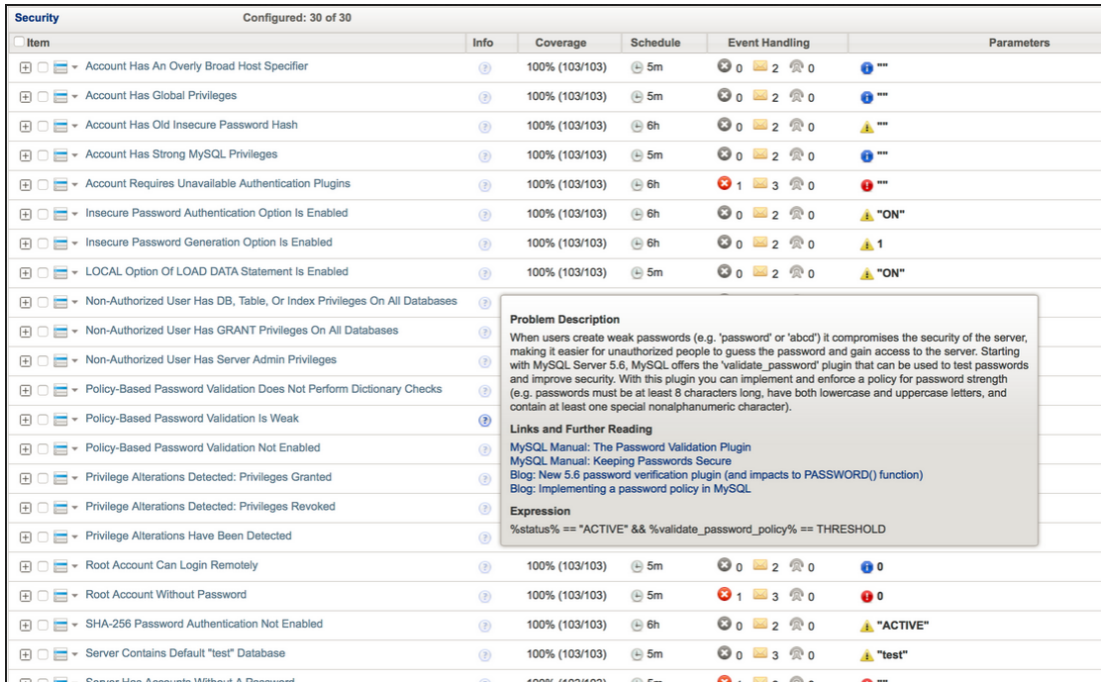
- Applications – minimize sharing a credentials (username/password)
  - Finer grained the better – don't overload across many applications/servers
- Should enable support for credential rotation
  - Do not require all passwords to be changed in synchronization.
  - Facilitates better troubleshooting and root-cause analysis.
- Steps to changing credentials should be secure and straightforward
  - Not embedded in your code
    - Can be changed without redeploying an application
    - Should never be stored in version control and must differ between environments.
    - Applications should get credentials using a secure configuration methodology.

MySQL Enterprise Edition

- MySQL Enterprise Authentication
  - External Authentication Modules
    - Microsoft AD, Linux PAMs
- MySQL Enterprise Encryption
  - Public/Private Key Cryptography
  - Asymmetric Encryption
  - Digital Signatures, Data Validation
- MySQL Enterprise Firewall
  - Block SQL Injection Attacks
  - Intrusion Detection
- 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
- MySQL Enterprise TDE
  - AES 256 encryption
  - Key Management

MySQL Enterprise Monitor

- Enforce MySQL Security Best Practices
  - Identifies Vulnerabilties
  - Assesses current setup against security hardening policies
- Monitoring & Alerting
  - User Monitoring
  - Password Monitoring
  - Schema Change Monitoring
  - Backup Monitoring
  - Configuration Management
  - Configuration Tuning Advice
- Centralized User Management

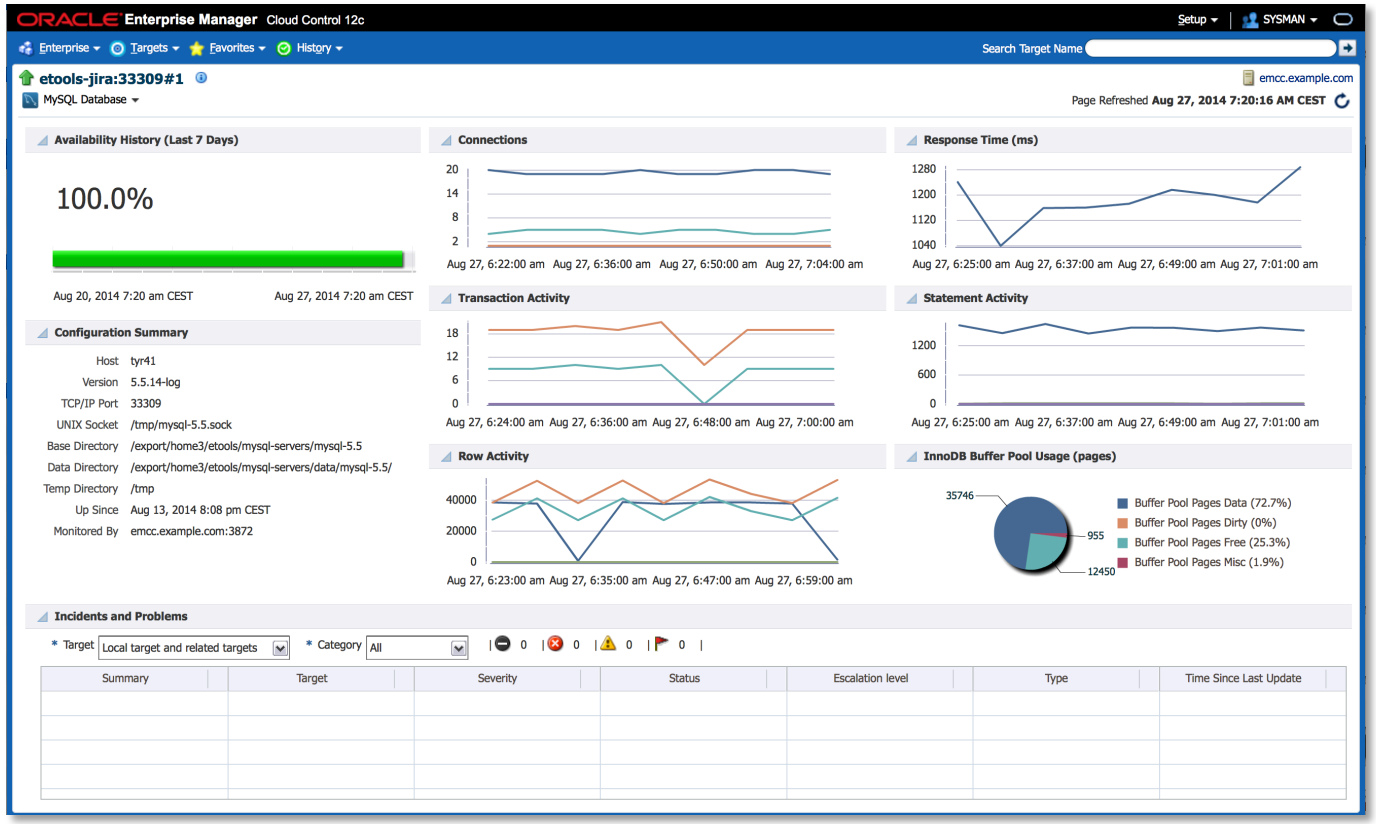


"I definitely recommend the MySQL Enterprise Monitor to DBAs who don't have a ton of MySQL experience. It makes monitoring MySQL security, performance and availability very easy to understand and to act on."

Sandi Barr
Sr. Software Engineer
Schneider Electric

Oracle Enterprise Manager for MySQL

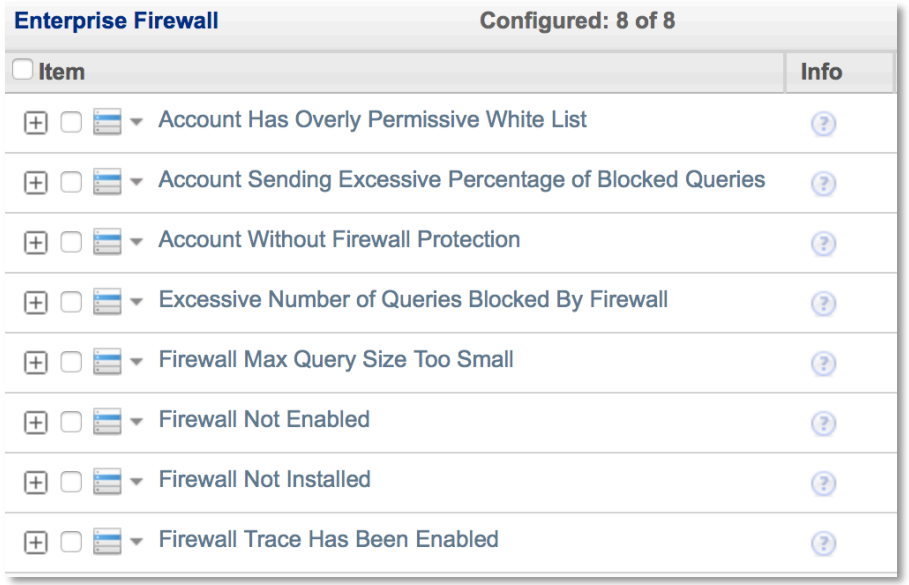
- Availability monitoring
- Performance monitoring
- Configuration monitoring
- All available metrics collected
  - Allowing for custom threshold based incident reports
- MySQL auto-detection



ORACLE ENTERPRISE MANAGER

MySQL Enterprise Firewall

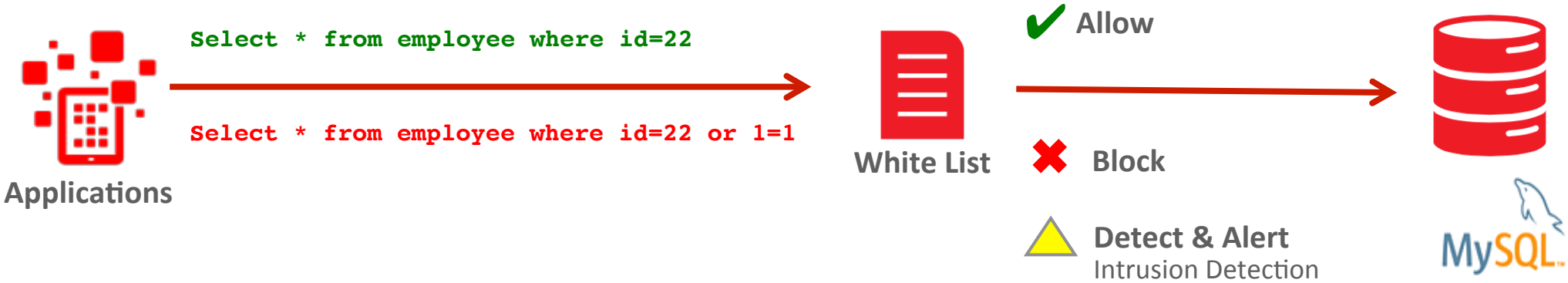
- Real Time Protection
  - Queries analyzed and matched against White List
- Blocks SQL Injection Attacks
  - Block Out of Policy Transactions
- Intrusion Detection
  - Detect and Alert on Out of Policy Transactions
- Learns White List
  - Automated creation of approved list of SQL command patterns on a per user basis
- Transparent
  - No changes to application required



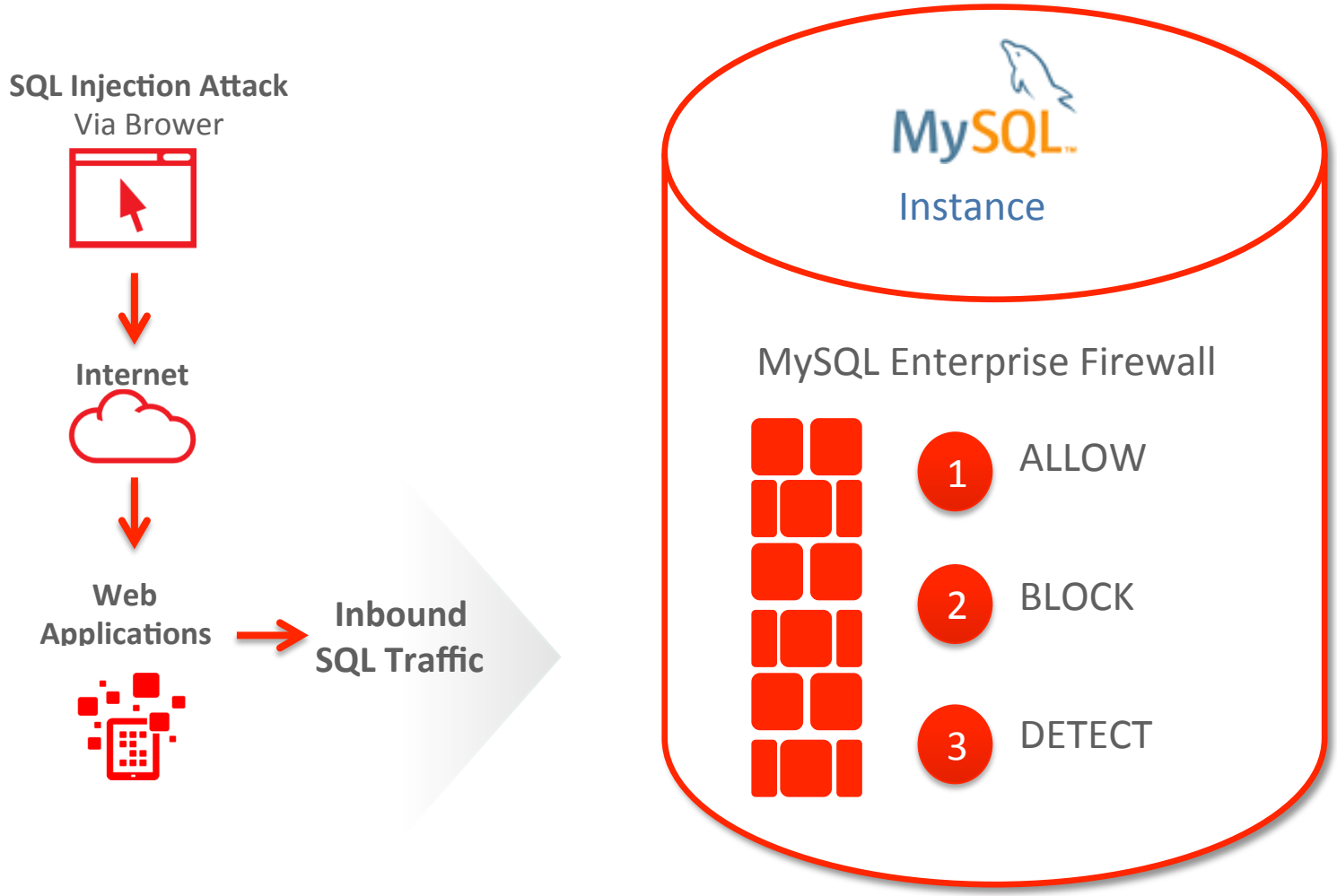
MySQL Enterprise Firewall monitoring

MySQL Enterprise Firewall

- Block SQL Injection Attacks
  - Allow: SQL Statements that match Whitelist
  - Block: SQL statements that are not on Whitelist
- Intrusion Detection System
  - Detect: SQL statements that are not on Whitelist
    - SQL Statements execute and alert administrators



MySQL Enterprise Firewall: Overview

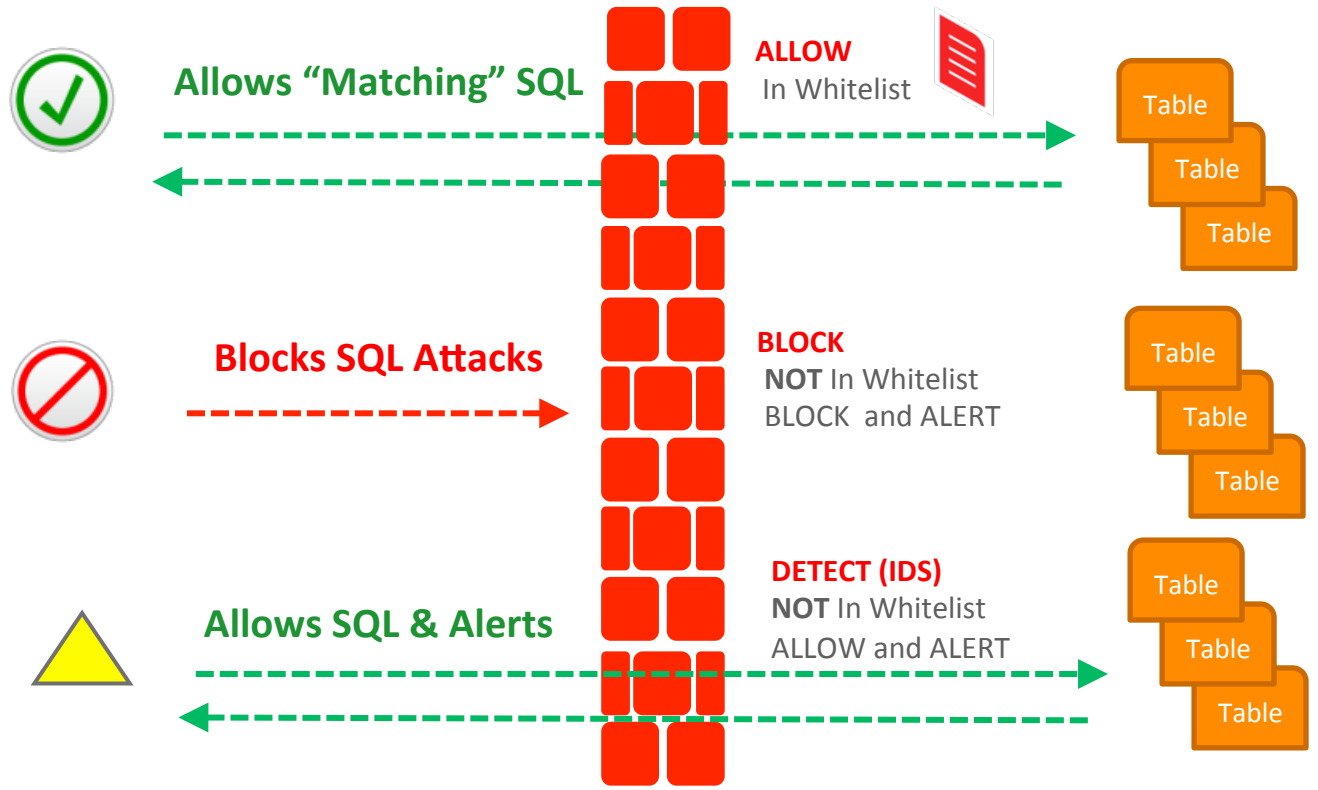


MySQL Enterprise Firewall: Operating Modes

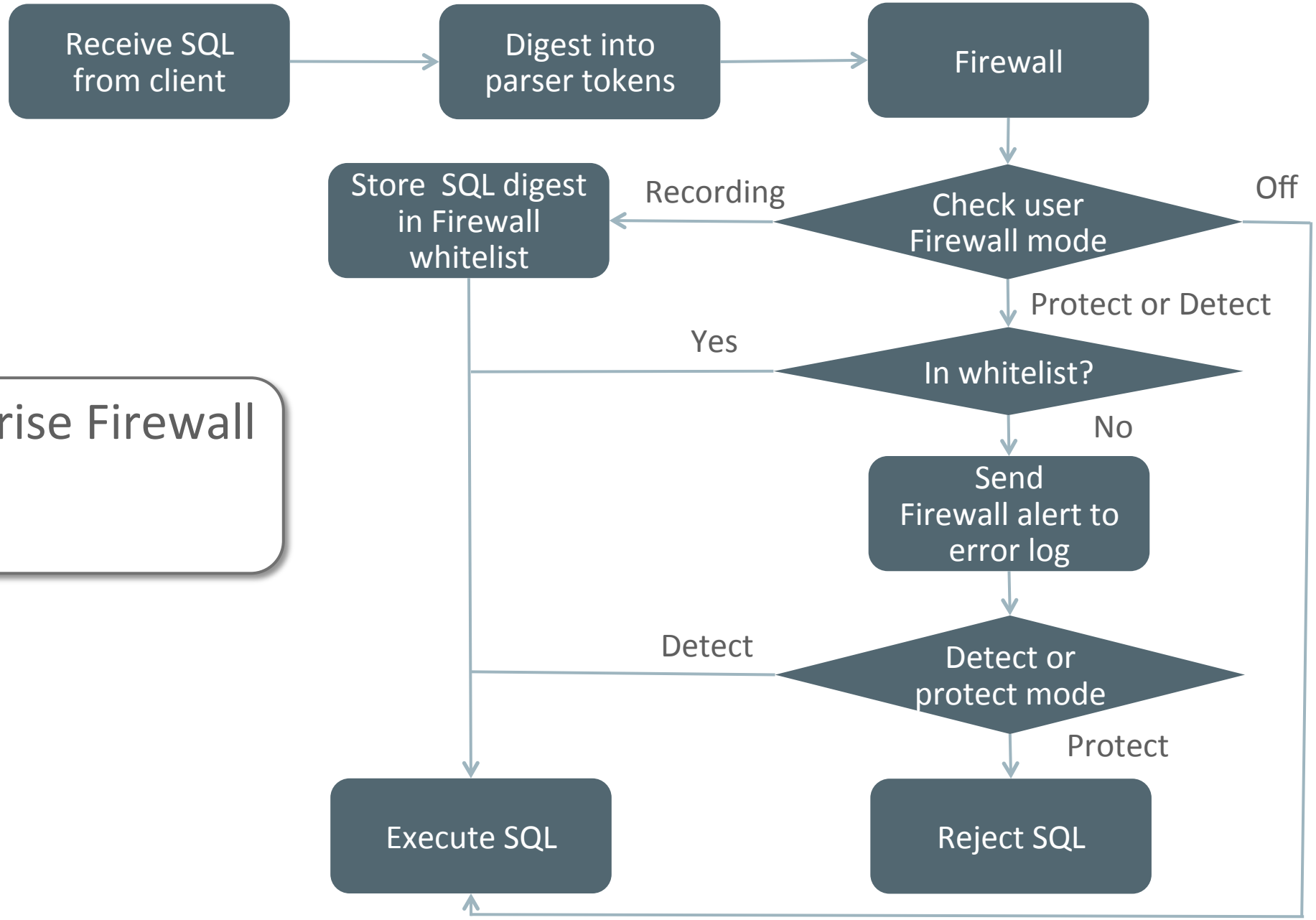
1. ALLOW – Execute SQL
   - SQL Matches Whitelist

2. BLOCK – Block the request
   - Not in Whitelist

3. DETECT – Execute SQL & Alert
   - Not in Whitelist



MySQL Enterprise Firewall Workflow

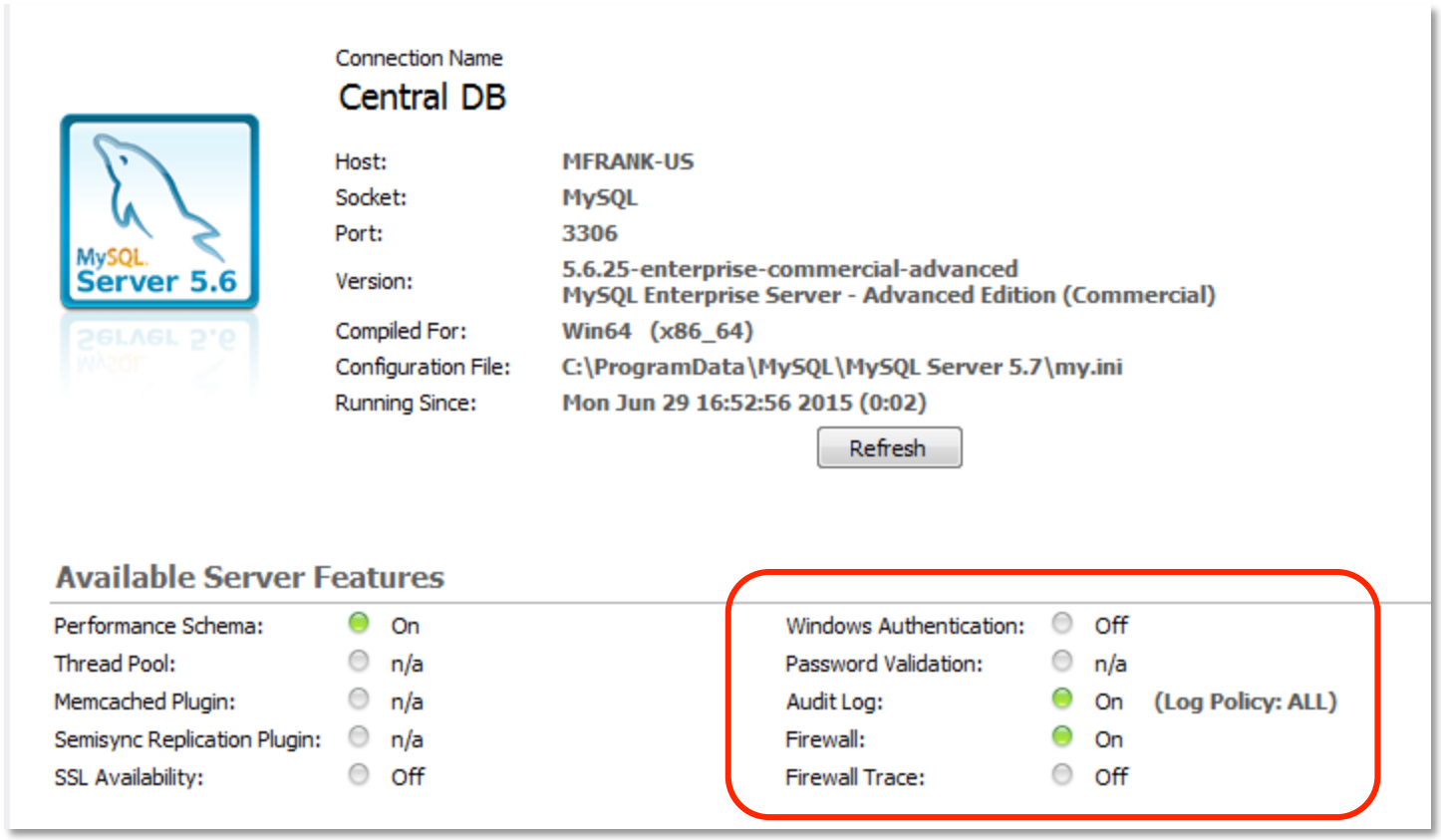


MySQL Enterprise Firewall Details

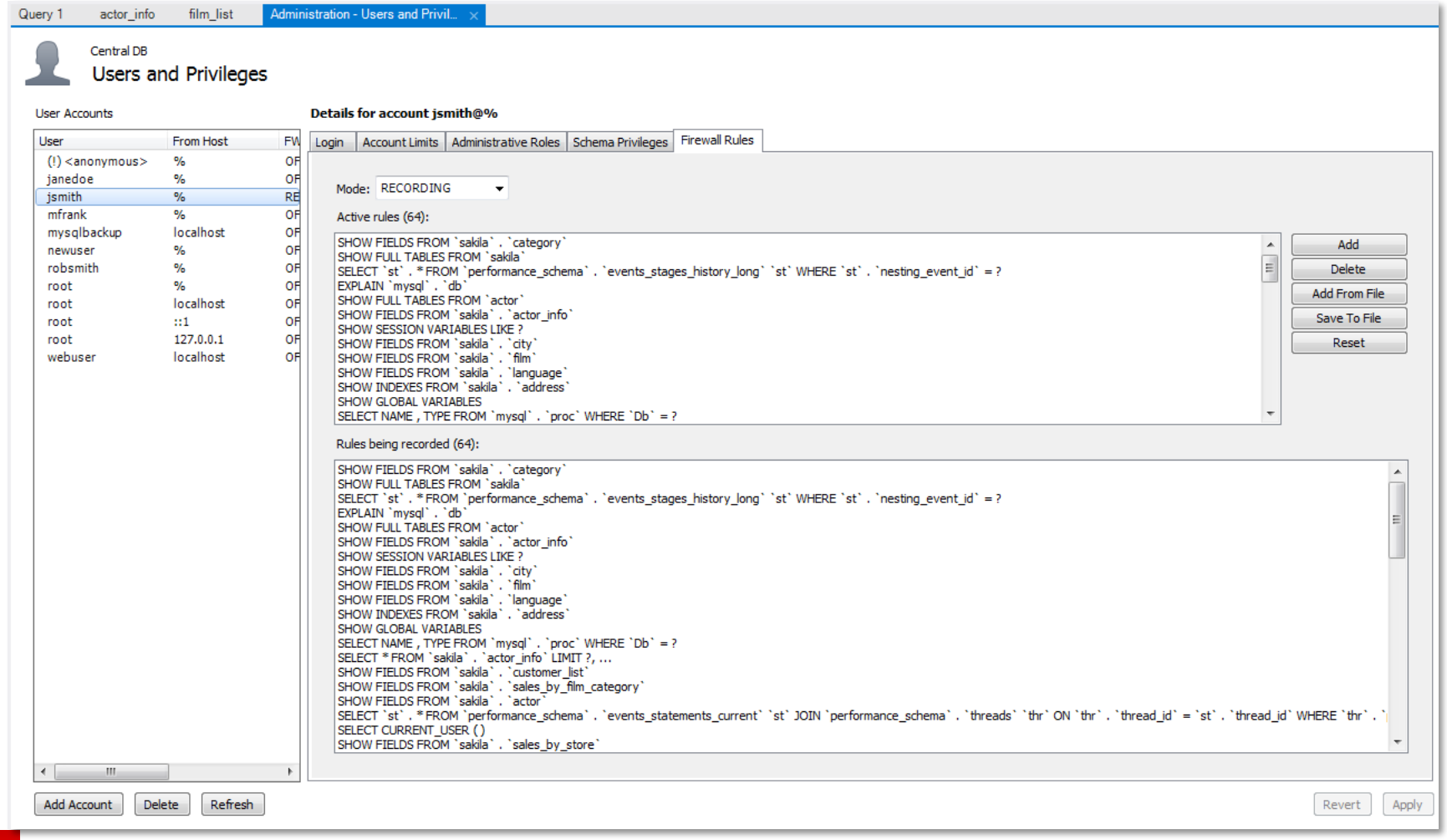
- Firewall operation is turned on at a per user level
- Per User States are
  - RECORDING
  - PROTECTING
  - DETECTING
  - OFF

```
call mysql.set_firewall_mode ('fwuser@localhost', 'RECORDING');
call mysql.set_firewall_mode ('fwuser@localhost', 'PROTECTING');
call mysql.set_firewall_mode ('fwuser@localhost', 'DETECTING');
call mysql.set_firewall_mode ('fwuser@localhost', 'OFF');
```

MySQL Workbench: Firewall Status



MySQL Enterprise Firewall: Per User Whitelists



MySQL Enterprise Firewall:
What happens when SQL is blocked in Protect Mode?

- The client application gets an ERROR

```
mysql> SELECT first_name, last_name FROM customer WHERE customer_id = 1 OR TRUE;
ERROR 1045 (28000): Statement was blocked by Firewall
mysql> SHOW DATABASES;
ERROR 1045 (28000): Statement was blocked by Firewall
mysql> TRUNCATE TABLE mysql.user;
ERROR 1045 (28000): Statement was blocked by Firewall
```

- Reported to the Error Log
- Increment Counter

MySQL Enterprise Firewall: Monitoring

Firewall Status Counters

```
mysql> SHOW GLOBAL STATUS LIKE 'Firewall%';
+------------------------------+-------+
| Variable_name                | Value |
+------------------------------+-------+
| Firewall_access_denied       | 3     |
| Firewall_access_granted      | 2     |
| Firewall_access_suspicious   | 1     |
| Firewall_cached_entries      | 4     |
+------------------------------+-------+
```

MySQL Enterprise Firewall: Whitelist Example

- `mysql> SELECT userhost, substr(rule,1,80) FROM mysql.firewall_whitelist WHERE userhost= 'wpuser@localhost';`

```
+------------------+--------------------------------------------------------------------------------+
| userhost         | substr(rule,1,80)                                                              |
+------------------+--------------------------------------------------------------------------------+
| wpuser@localhost | SELECT * FROM `wp_posts` WHERE `ID` = ? LIMIT ?                                |
| wpuser@localhost | SELECT `option_value` FROM `wp_options` WHERE `option_name` = ? LIMIT ?        |
| wpuser@localhost | SELECT `wp_posts` . * FROM `wp_posts` WHERE ? = ? AND `wp_posts` . `ID` = ? AND  |
...
| wpuser@localhost | UPDATE `wp_posts` SET `comment_count` = ? WHERE `ID` = ?                       |
| wpuser@localhost | SELECT `t` . * , `tt` . * FROM `wp_terms` AS `t` INNER JOIN `wp_term_taxonomy` A |
| wpuser@localhost | SELECT `t` . * , `tt` . * FROM `wp_terms` AS `t` INNER JOIN `wp_term_taxonomy` A |
+------------------+--------------------------------------------------------------------------------+
```

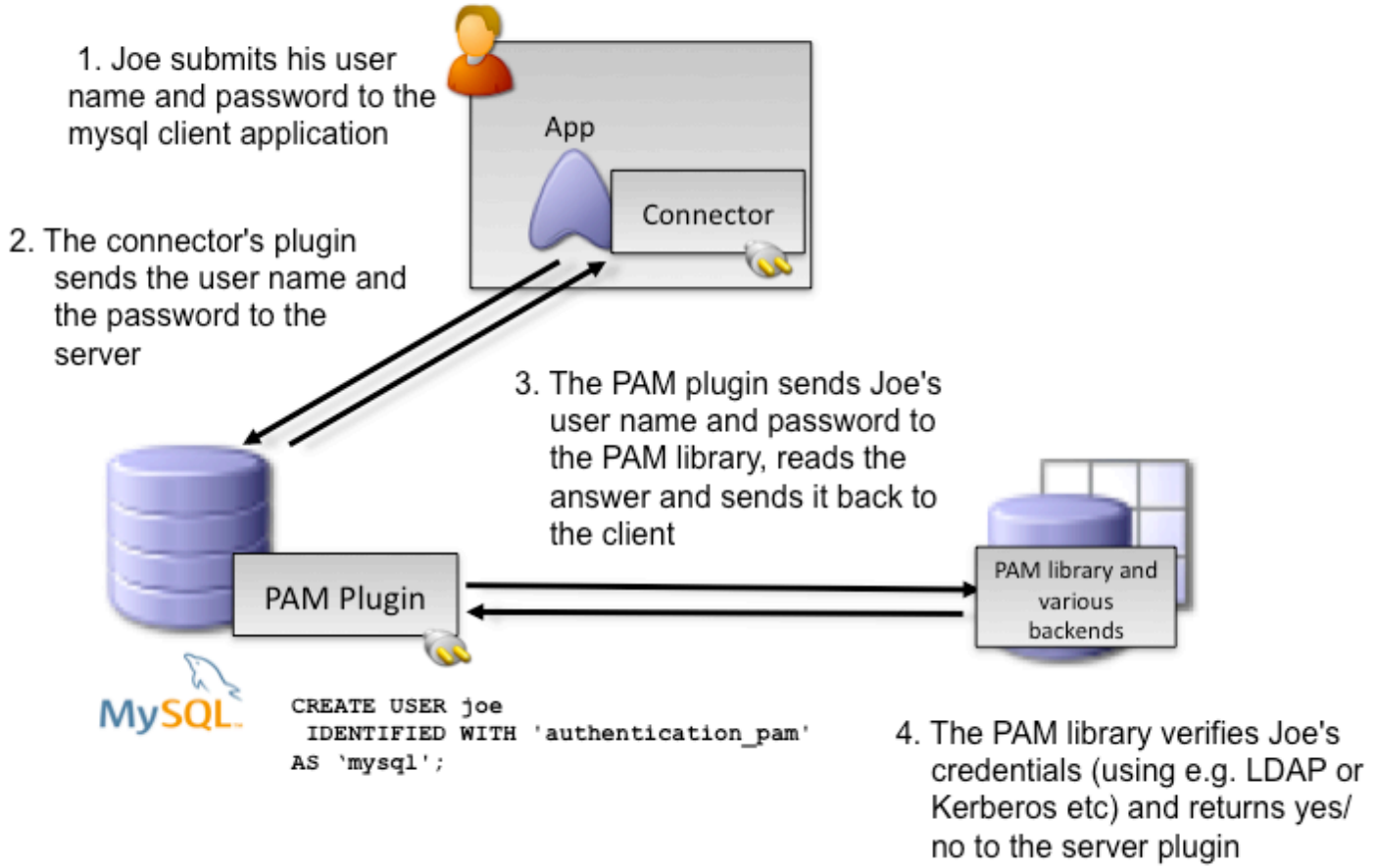
MySQL Enterprise Authentication

Integrates MySQL with existing security infrastructures

- Integrate with Centralized Authentication Infrastructure
  - Centralized Account Management
  - Password Policy Management
  - Groups & Roles
- PAM (Pluggable Authentication Modules)
  - Standard interface (Unix, LDAP, Kerberos, others)
  - Windows
    - Access native Windows service - Use to Authenticate users using Windows Active Directory or to a native host

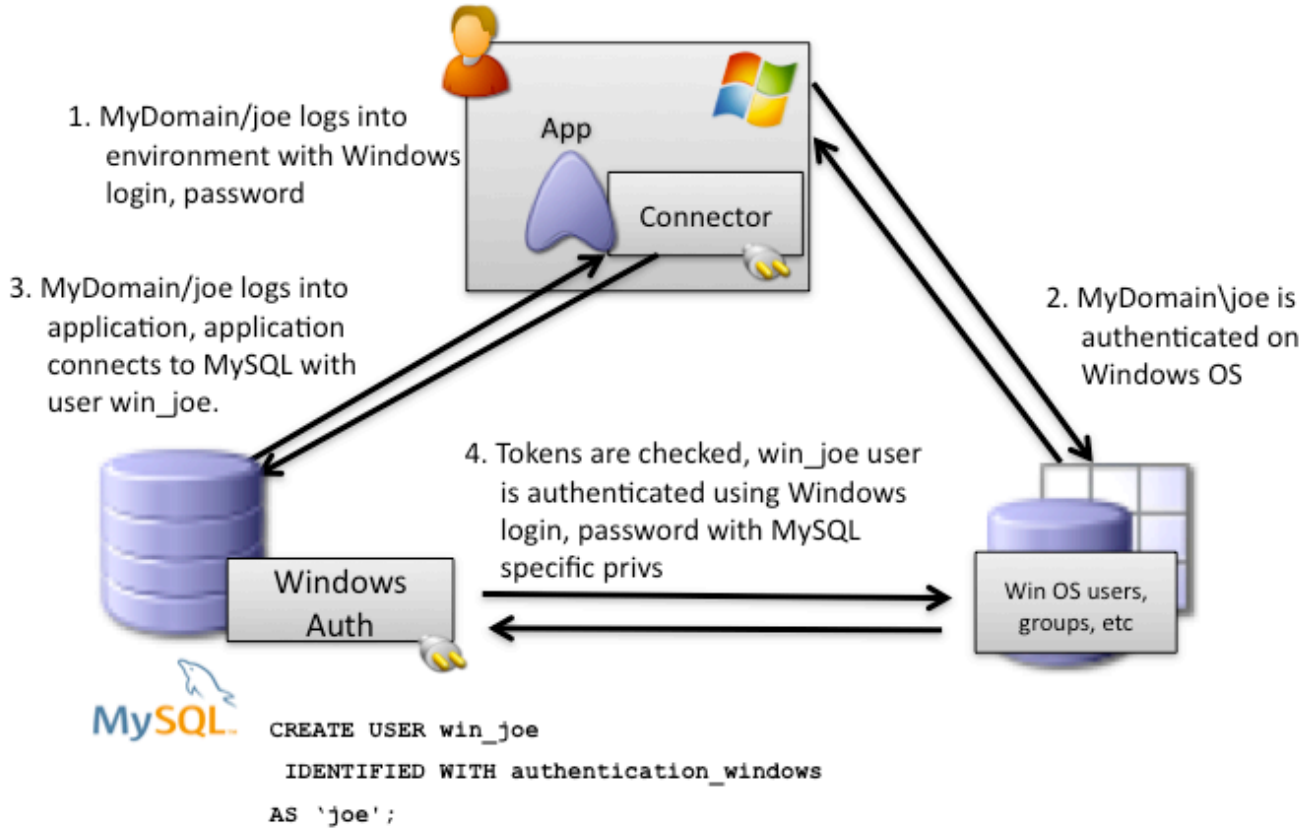
MySQL Enterprise Authentication: PAM

- Standard Interface
  - LDAP
  - Unix/Linux
- Proxy Users



MySQL Enterprise Authentication: Windows

- Windows Active Directory
- Windows Native Services



MySQL Enterprise Authentication: LDAP, Others?

- Roadmap includes native LDAP
  - Two modes
    - User/Password
    - Token Based

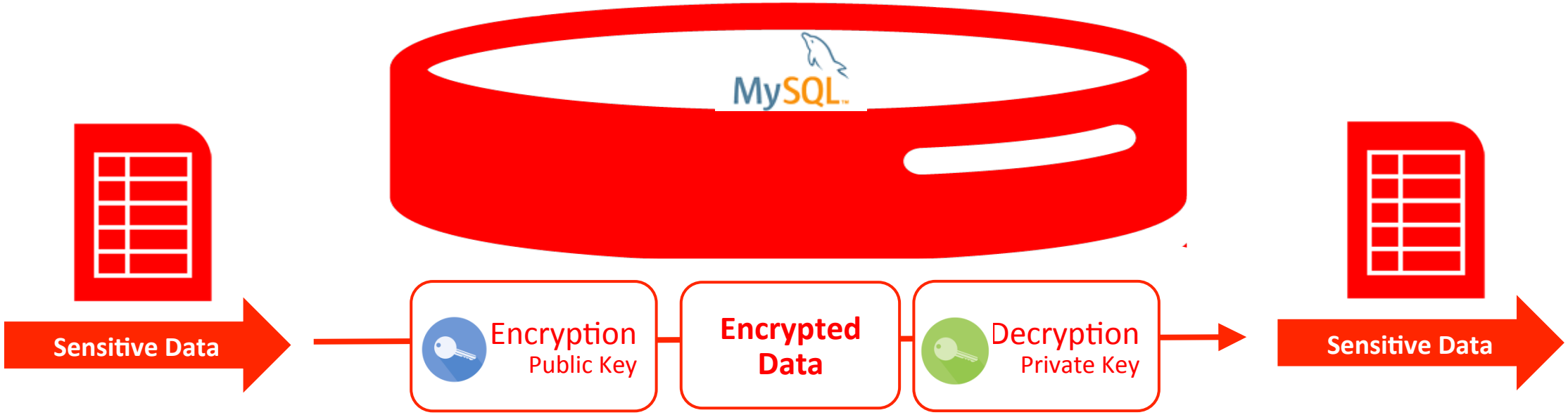
- Other integrations for authentication that are wanted?

MySQL Enterprise Encryption

- MySQL encryption functions
  - Symmetric encryption AES256 (All Editions)
  - Public-key / asymmetric cryptography – RSA
- Key management functions
  - Generate public and private keys
  - Key exchange methods: DH
- Sign and verify data functions
  - Cryptographic hashing for digital signing, verification, & validation – RSA,DSA

MySQL Enterprise Encryption

Encryption/Decryption within MySQL

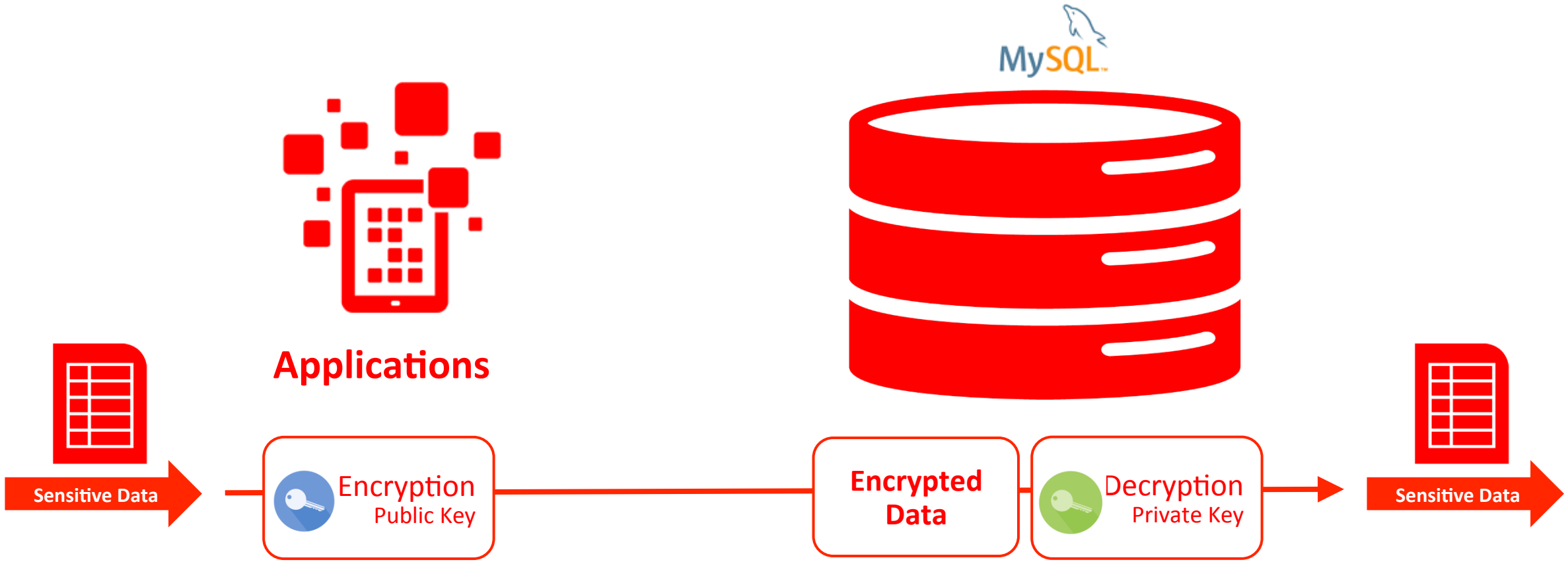


Private / Public Key Pairs

- Generate using MySQL Enterprise Encryption Functions
- Use externally generated (e.g. OpenSSL)

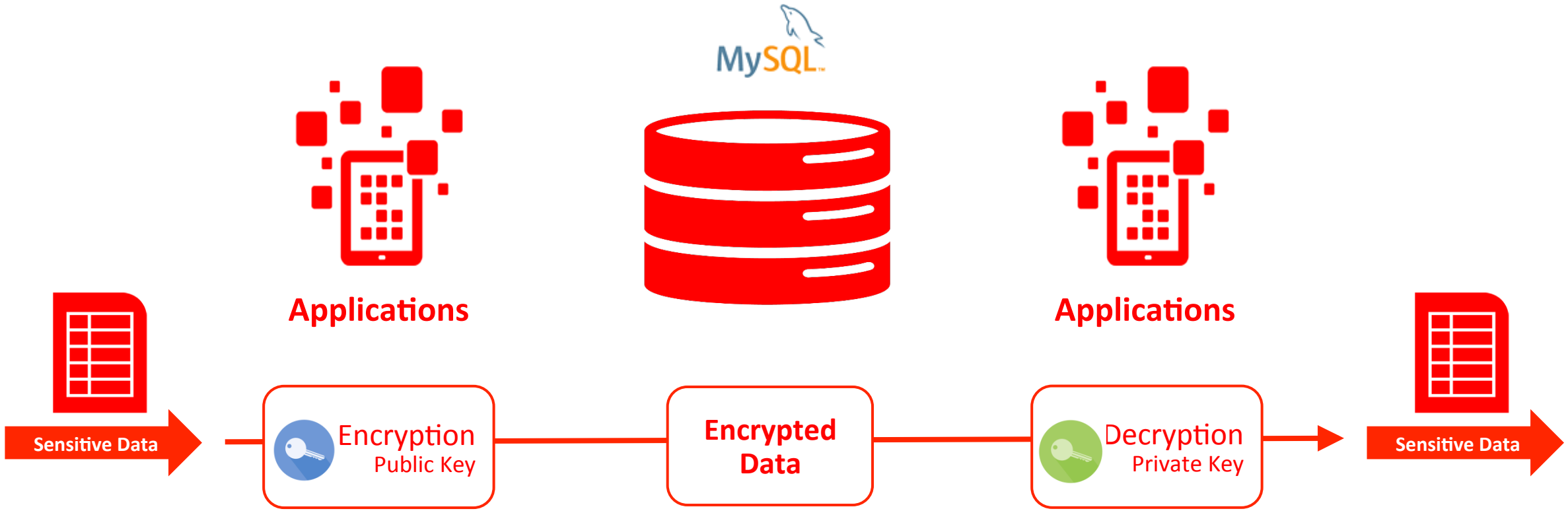
MySQL Enterprise Encryption

App Encrypts/MySQL Decrypts



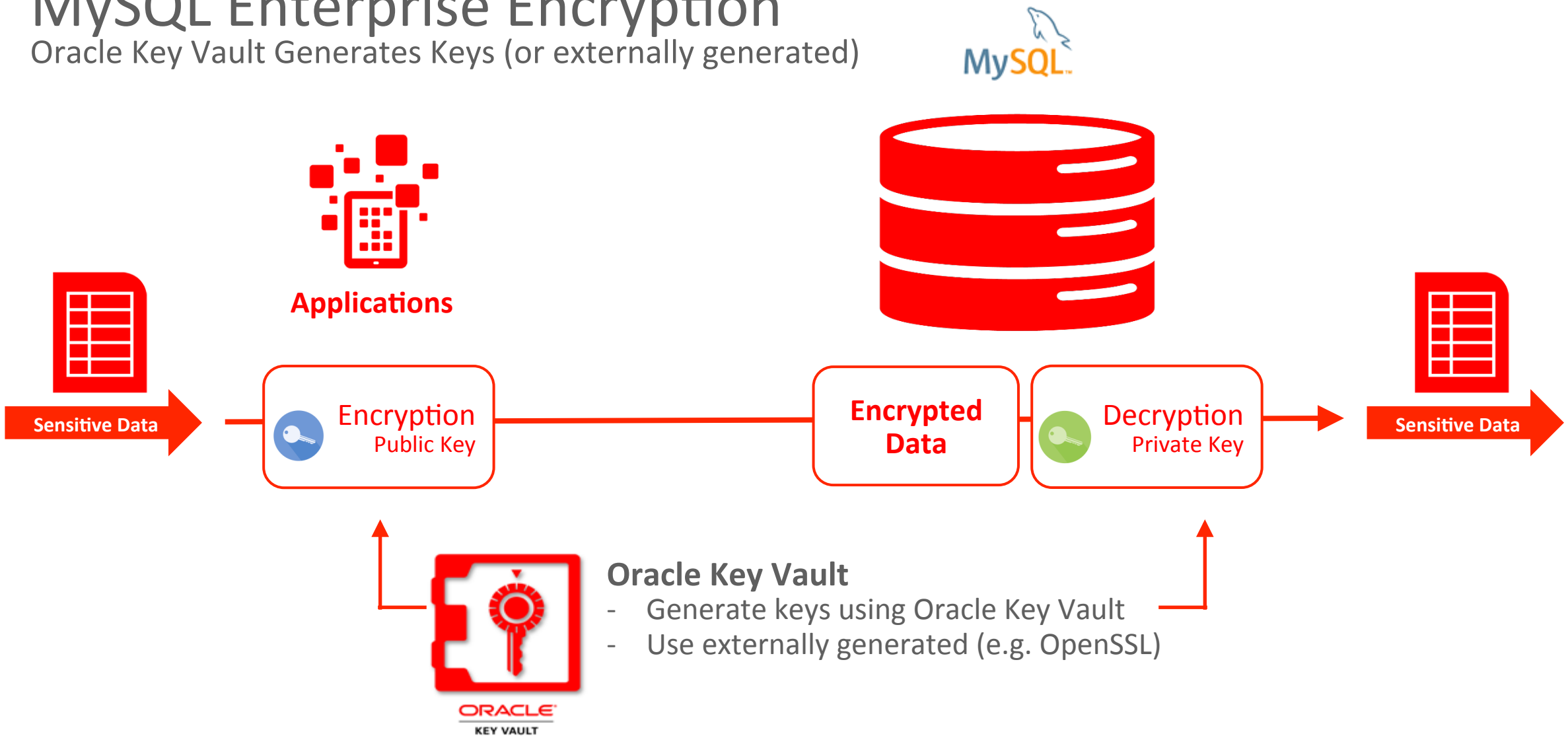
MySQL Enterprise Encryption

App Encrypts / MySQL Stores / MySQL Decrypts



MySQL Enterprise Encryption

Oracle Key Vault Generates Keys (or externally generated)



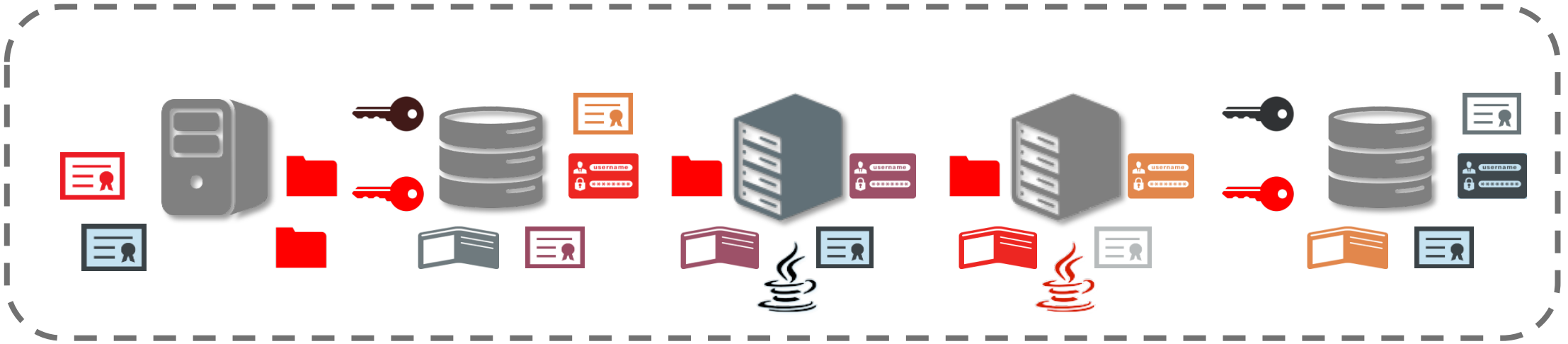
Oracle Key Vault

- Generate keys using Oracle Key Vault
- Use externally generated (e.g. OpenSSL)

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.

The Challenges of Key Management



Management

- Proliferation of encryption wallets and keys
- Authorized sharing of keys
- Key availability, retention, and recovery
- Custody of keys and key storage files

Regulations

- Physical separation of keys from encrypted data
- Periodic key rotations
- Monitoring and auditing of keys
- Long-term retention of keys and encrypted data

Regulatory Drivers

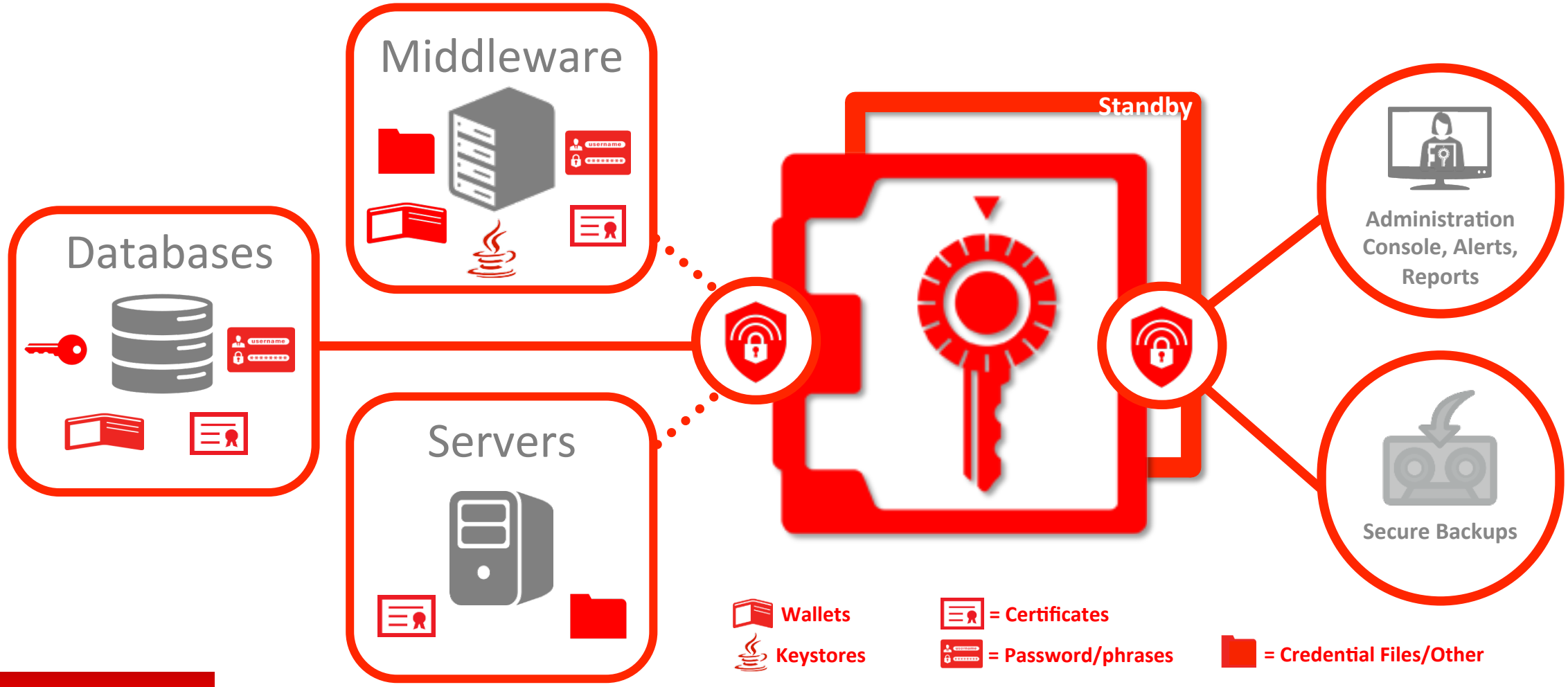
PCI DSS v3.0
November 2013

3.5 Store cryptographic keys in a secure form (3.5.2), in the fewest possible locations (3.5.3) and with access restricted to the fewest possible custodians (3.5.1)

3.6 Verify that key-management procedures are implemented for periodic key changes (3.6.4)

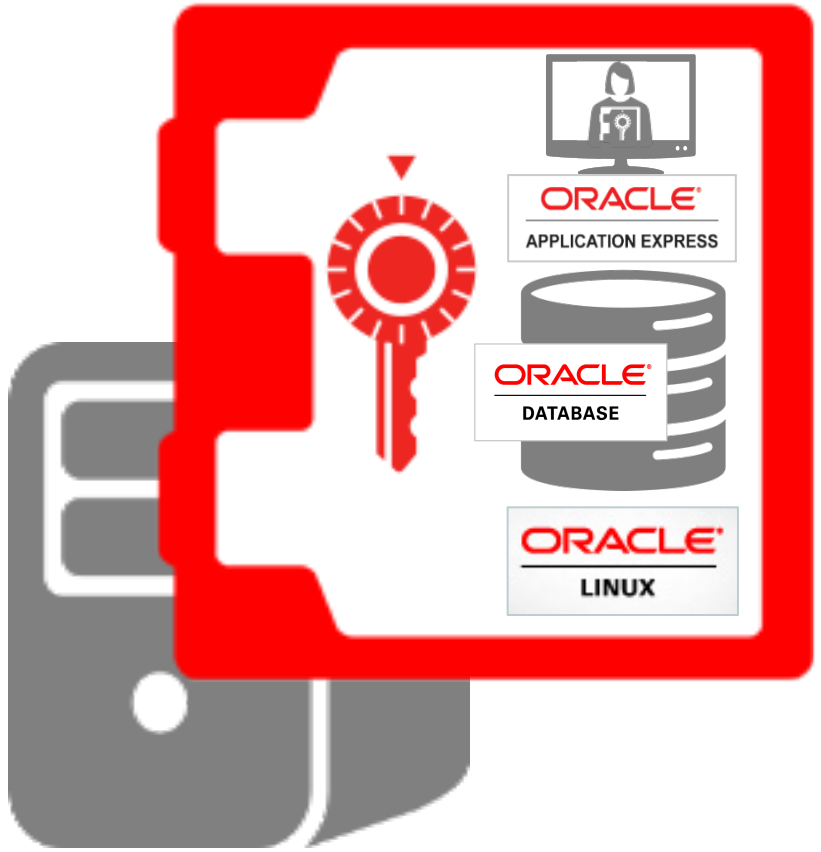
And more!

Key Vault High-Level Architecture

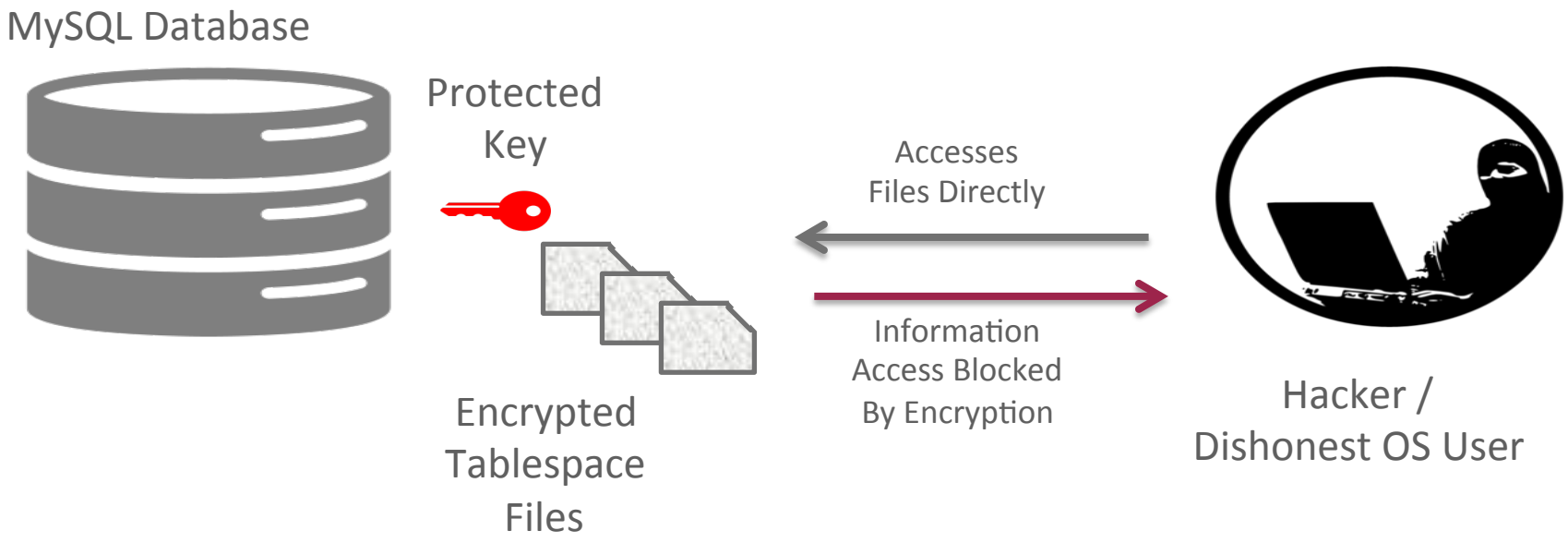


Oracle Key Vault Software Appliance Platform

- Turnkey solution based on hardened stack
- Includes Oracle Database and security options
- Open x86-64 hardware to choose from
- Easy to install, configure, deploy, and patch
- Separation of duties for administrative users
- Full auditing, preconfigured reports, and alerts



Attack on Files



MySQL Transparent Data Encryption Goals

- Protect the Key – Most Important Thing
- Strong Encryption – AES 256
- Keep simple to manage – “complexity is the enemy of security”
  - One master key for whole instance
  - Easy , flexible to manage encrypted tablespaces (support Transportable Tablespace export/import)
- High Performance / Low Overhead
  - Simple Key Rotation without massive decrypt/encryption costs
- High Quality
  - Won’t try to encrypt everything initially – but add more and more options over time
- Provide initial infrastructure to expand
  - Feedback to us from customers - Expand over time - Audit Data Encryption, Various Logs, Columns, Backups …

Introduction to Transparent Data Encryption in MySQL

SQL

- New option in CREATE TABLE ENCRYPTION="Y"
- New SQL : ALTER INSTANCE ROTATE INNODB MASTER KEY

Plugin Infrastructure

- New plugin type : keyring
- Ability to load plugin before InnoDB initialization : --early-plugin-load

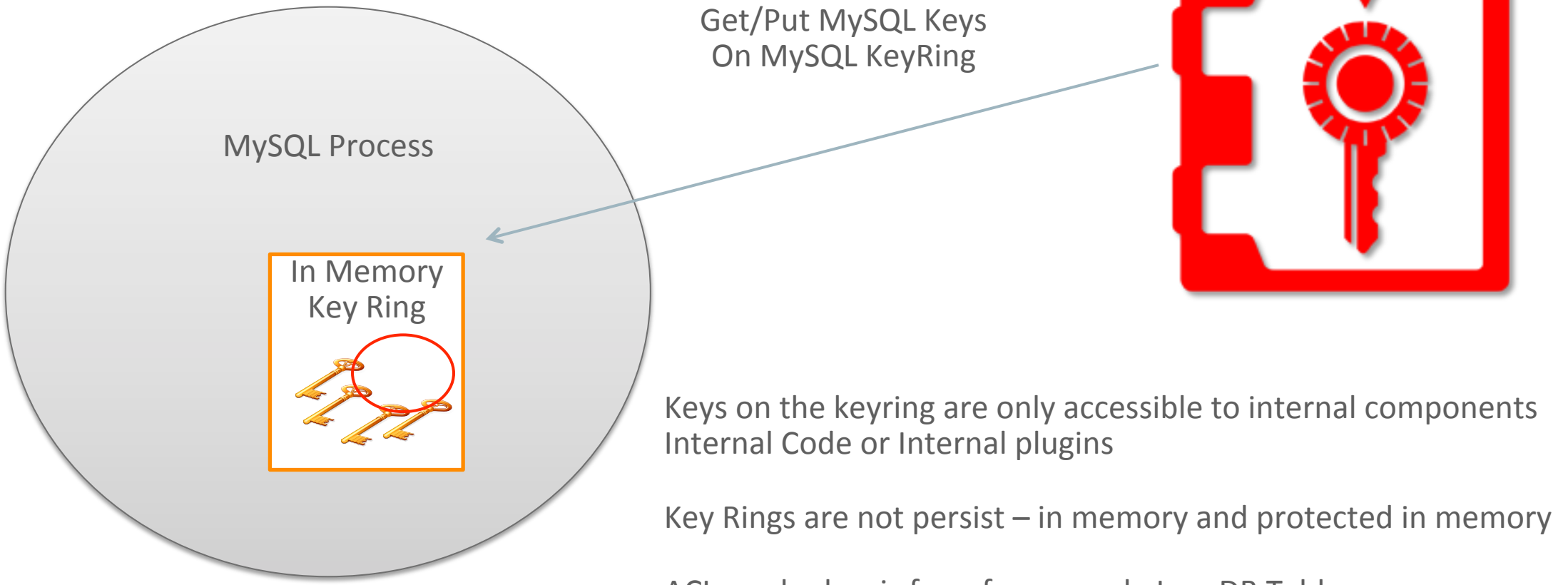
Keyring plugin

- Used to retrieve keys

InnoDB

- Support for encrypted tables
- IMPORT/EXPORT of encrypted tables
- Support for master key rotation

MySQL 5.7 Key Ring

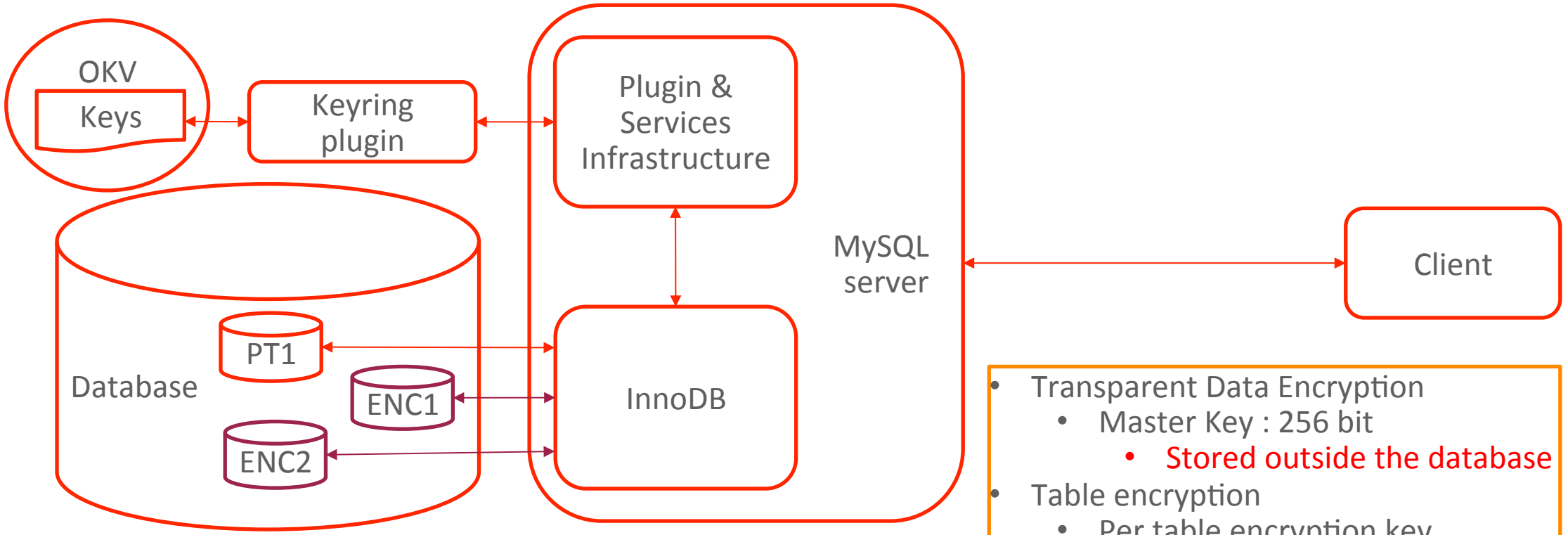


Keys on the keyring are only accessible to internal components
Internal Code or Internal plugins

Key Rings are not persist – in memory and protected in memory

ACLs - who key is for – for example InnoDB Tablespaces

More Detailed Architecture of Transparent Data Encryption in MySQL



- Transparent Data Encryption
  - Master Key : 256 bit
    - Stored outside the database
- Table encryption
  - Per table encryption key
    - Protected by master key
    - 256 bit
  - AES CBC for data encryption

Database Auditing

Maintaining an audit trail is an essential security best practice

- "Trust but verify" approach to security
  - Ensure users with strong privileges don't misuse those privileges
- Business Audit – Data Validity
  - Here's proof my database data is accurate/correct
  - Prove no tampering to data has occurred
- Forensic analysis – as a component of any defense-in-depth strategy
  - Proactive - Am being / Was hacked
  - Reactive – How were we hacked, what was changed, taken, etc.

MySQL Enterprise Audit

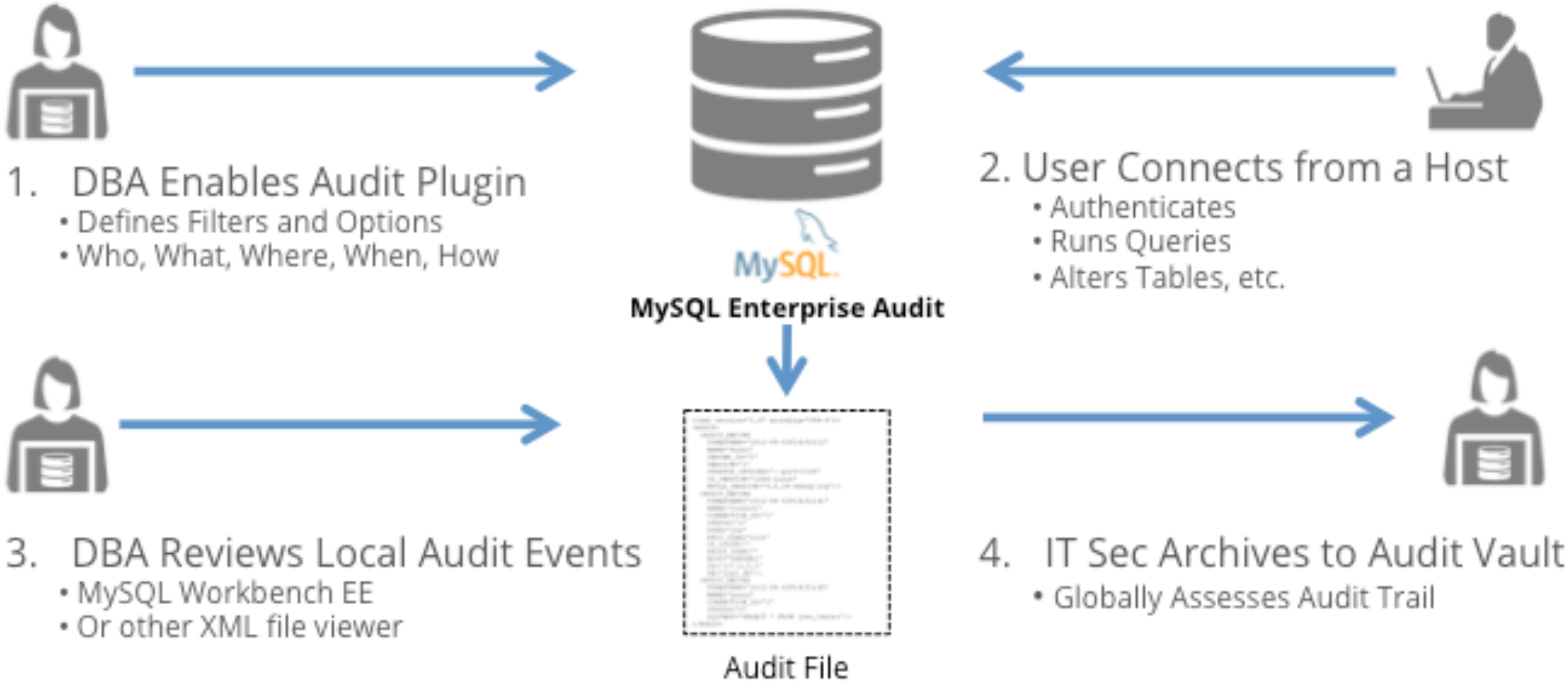
- Out-of-the-box logging of connections, logins, and query
- Simple to fine grained policies for filtering, and log rotation
- Dynamically enabled, disabled: no server restart
- XML-based audit stream
  - Send data to a remote server / audit data vault
    - Oracle Audit Vault
    - Splunk, etc.

Adds regulatory compliance to MySQL applications (HIPAA, Sarbanes-Oxley, PCI, etc.)

Why you need to audit

- Sensitive data
  - Create Audit rules on your most sensitive data, powerful users, exposed applications
- Comply with regulatory standards
  - PCI, HIPAA, FERPA and SOX requirements
- Minimal Overhead
  - Collects critical audit data without minimal performance impact

MySQL Enterprise Audit - Work Flow



Complete Audit Data

Complete event details

- Who
- What
- When
- How
- Status
- From Where
- DB version
- OS version
- Options
- And more

```
<?xml version="1.0" encoding="UTF-8"?>
<AUDIT>
  <AUDIT_RECORD
    TIMESTAMP="2012-08-02T14:52:12"
    NAME="Audit"
    SERVER_ID="1"
    VERSION="1"
    STARTUP_OPTIONS="--port=3306"
    OS_VERSION="i686-Linux"
    MYSQL_VERSION="5.5.28-debug-log"/>
  <AUDIT_RECORD
    TIMESTAMP="2012-08-02T14:52:41"
    NAME="Connect"
    CONNECTION_ID="1"
    STATUS="0"
    USER="joe"
    PRIV_USER="root"
    OS_LOGIN=""
    PROXY_USER=""
    HOST="SERVER1"
    IP="127.0.0.1"
    DB="joes_db"/>
  <AUDIT_RECORD
    TIMESTAMP="2012-08-02T14:53:45"
    NAME="Query"
    CONNECTION_ID="1"
    STATUS="0"
    SQLTEXT="SELECT * FROM joes_table;"/>
</AUDIT>
```

New Audit Filtering

- Starting with MySQL Enterprise 5.7.13
- Allows DBAs to “custom” design audit process
  - Use very fine grained rules
    - Reduce audit log file size
    - Reduce File System IO and Storage / Increases performance (less items logged).
    - Increases audit log post processing efficiency – less data to process for immediate answers.
    - Defined using JSON
  - Coarse grained rules
    - When you need to watch everything
    - Obsolete. Recommended is to use new audit log filtering.

New Auditing Solution Uses

- Expanded “Event” model
  - Allows for very fine grained auditing
- Simple but powerful
  - Uses JSON to define filters

Event class	Event subclass
GENERAL	STATUS
CONNECTION	CONNECT
	CHANGE_USER
	DISCONNECT
TABLE_ACCESS	READ
	INSERT
	UPDATE
	DELETE

Connection Event Fields

Name	Type	Description	Name	Type	Description
status	INT	Status of the event: 0: OK, otherwise error state.	**host.str**	STRING	Host string.
user.str	STRING	Connecting user string.	**ip.str**	STRING	IP string.
priv_user.str	STRING	Account user string.	**database.str**	STRING	Initial database name.
external_user.str	STRING	External plugin authenticated user.	**connection_type**	INT	Connection type value: 0: Undefined; 1: TCP/IP; 2: Socket; 3: Named pipe; 4: SSL; 5: Shared memory
proxy_user.str	STRING	Proxy user string.			

... and other fields.

Table Event Fields

Name	Type	Description
event_subclass	INT	Table access type (read, insert, update or delete).
connection_id	STRING	Unique connection id.
sql_command_id	UINT	SQL statement type (SELECT, INSERT ... SELECT, LOAD XML INFILE etc.).
query	STRING	Query string accessing the table.
table_database	STRING	Database (schema) name.
table_name	STRING	Table name.

Filters can be SIMPLE

Log all connection events:

- succeessfull and failed connection attempts;
- disconnects;
- user change during session (change_user command).

```
{ "filter": {
    "class": {
      "name": "connection" } } }
```

Filters can be Specific - Log Failed SSL Connects

Log failed SSL connection attempts:

```
{ "filter": {
   "class": {
    "name": "connection",
    "event": {
     "name": "connect",
     "log": {
      "and": [
       { "not": { "field": { "name": "status",
                             "value": 0 } } },
       { "field": { "name": "connection_type",
                    "value": "::ssl" }}]}}}}}
```

Rules can be Specific related to Tables

All deletions, insertions, updates on bank_database.accounts

```
{ "filter":{
   "class": {
    "name": "table_access",
    "event": {
     "name": [ "delete", "insert", "update" ],
     "log": {
       "and": [ { "field": { "name": "table_database.str",
                             "value": "bank_database" } },
                { "field": { "name": "table_name.str",
                             "value": "accounts" }}]}}}}
```

A Lot of Filtering Possibilities- String Functions

```
{ "filter": {
    "class": {
      "name": "general",
      "event": {
        "name": "status",
        "log": {
          "and":[
            { "field":
              { "name": "general_command.str",
                "value": "Query" } },
            { "function":
              { "name": "string_find",
                "args": [ "create_db,create_user",
                          { "field": "general_sql_command.str" }
                        ]}}]}}}}}
```

New Auditing Solution Uses

- Expanded “Event” model
  - Allows for very fine grained auditing
- Simple but powerful
  - Uses JSON to define filters

Basic Audit Log Events

Event class	Event subclass
GENERAL	STATUS
CONNECTION	CONNECT
	CHANGE_USER
	DISCONNECT
TABLE_ACCESS	READ
	INSERT
	UPDATE
	DELETE

New audit log filtering can filter against event fields values!

Connection Event Fields

Name	Type	Description	Name	Type	Description
status	INT	Status of the event: 0: OK, otherwise error state.	**host.str**	STRING	Host string.
user.str	STRING	Connecting user string.	**ip.str**	STRING	IP string.
priv_user.str	STRING	Account user string.	**database.str**	STRING	Initial database name.
external_user.str	STRING	External plugin authenticated user.	**connection_type**	INT	Connection type value: 0: Undefined; 1: TCP/IP; 2: Socket; 3: Named pipe; 4: SSL; 5: Shared memory
proxy_user.str	STRING	Proxy user string.			

... and other fields.

Table Event Fields

Name	Type	Description
event_subclass	INT	Table access type (read, insert, update or delete).
connection_id	STRING	Unique connection id.
sql_command_id	UINT	SQL statement type (SELECT, INSERT ... SELECT, LOAD XML INFILE etc.).
query	STRING	Query string accessing the table.
table_database	STRING	Database (schema) name.
table_name	STRING	Table name.

Filters can be SIMPLE

Log all connection events:

- succeessfull and failed connection attempts;
- disconnects;
- user change during session (change_user command).

```
{ "filter": {
    "class": {
      "name": "connection" } } }
```

Filters can be Specific - Log Failed SSL Connects

Log failed SSL connection attempts:

```
{ "filter": {
   "class": {
    "name": "connection",
    "event": {
     "name": "connect",
     "log": {
      "and": [
       { "not": { "field": { "name": "status",
                             "value": 0 } } },
       { "field": { "name": "connection_type",
                    "value": "::ssl" }}]}}}}}
```

Rules can be Specific related to Tables

All deletions, insertions, updates on bank_database.accounts

```
{ "filter":{
   "class": {
    "name": "table_access",
    "event": {
     "name": [ "delete", "insert", "update" ],
     "log": {
       "and": [ { "field": { "name": "table_database.str",
                             "value": "bank_database" } },
                { "field": { "name": "table_name.str",
                             "value": "accounts" }}]}}}}
```

A Lot of Filtering Possibilities- String Functions

```
{ "filter": {
    "class": {
      "name": "general",
      "event": {
        "name": "status",
        "log": {
          "and":[
            { "field":
              { "name": "general_command.str",
                "value": "Query" } },
            { "function":
              { "name": "string_find",
                "args": [ "create_db,create_user",
                          { "field": "general_sql_command.str" }
                        ]}}]}}}}}
```

MySQL Enterprise Backup

- Online Backup for InnoDB (scriptable interface)
- Full, Incremental, Partial Backups (with compression)
- Strong Encryption (AES 256)
- Point in Time, Full, Partial Recovery options
- Metadata on status, progress, history
- Scales – High Performance/Unlimited Database Size
- Windows, Linux, Unix
- Certified with Oracle Secure Backup, NetBackup, Tivoli, others

MySQL Enterprise Oracle Certifications

- Oracle Enterprise Manager for MySQL
- Oracle Linux (w/DRBD stack)
- Oracle VM
- Oracle Solaris
- Oracle Solaris Clustering
- Oracle Clusterware
- Oracle Audit Vault and Database Firewall
- Oracle Secure Backup
- Oracle Fusion Middleware
- Oracle GoldenGate
- My Oracle Support

MySQL integrates into your Oracle environment

Oracle Audit Vault and Database Firewall

- Oracle DB Firewall
  - Oracle, MySQL, SQL Server, IBM DB2, Sybase
  - Activity Monitoring & Logging
  - White List, Black List, Exception List
- Audit Vault
  - Built-in Compliance Reports
  - External storage for audit archive


Thank You
ORACLE
MySQL

Index

- MySQL Enterprise Security
- MySQL Enterprise Authentication
- MySQL Enterprise Firewall
- MySQL Enterprise Transparent Data Encryption
- MySQL Enterprise Audit