RED HAT

OPEN SOURCE DAY

Europe, Middle East & Africa

Logo
Partner

#redhatosd

RED HAT
OPEN SOURCE DAY
Europe, Middle East & Africa

Database Security Threats
MariaDB Security Best Practices

Maria Luisa Raviol
MariaDB Senior Solutions Architect

MariaDB

#redhatosd

Security threats

Best practices

The Internet

Threats

Viruses

Hacker attacks

Defense

- Don’t allow TCP connections to MariaDB from the Internet
- Evaluate your DNS infrastructure
- Configure MariaDB to listen only to the application host
- Design your physical network to connect the app to MariaDB
- Use bind-address to bind to a specific network interface
- Use your OS’s firewall
- Keep your OS patched

Applications

Threats

Denial of Service Attacks created by overloading application

SQL query injection attacks

Defense

- Don’t run your application on the MariaDB Server
- Don’t install unnecessary packages
  - An overloaded application can cause MariaDB to be slow or even killed by the OS. (DDoS attack vector)
  - A compromised application or service can have many serious side effects
    - Discovery of MariaDB credentials
    - Direct access to data
    - Privilege escalation

Excessive Trust

Threats

Disgruntled employees

Mistakes and human error

Defense

- Limit users who have:
  - SSH access to your MariaDB server.
  - Sudo privileges on your MariaDB server.
- Set the secure_file_priv option to ensure that users with the FILE privilege cannot write or read MariaDB data or important system files.
- Do not run mysqld as root
- Avoid '%", use specific host names

Excessive Trust

Threats

Disgruntled employees

Mistakes and human error

Defense

- Don’t use the MariaDB “root” user for application access
- Minimize the privileges granted to the MariaDB accounts used by your applications
  - Don’t grant CREATE or DROP privileges.
  - Don’t grant the FILE privilege.
  - Don’t grant the SUPER privilege.
  - Don’t grant access to the mysql database
- Grant only the privileges required

Best Practices: Encryption

- Encrypt sensitive data in the application
  - Credit Card numbers, PII

- Encrypt data at rest
  - InnoDB tablespace encryption
  - InnoDB redo log encryption
  - Binary log encryption

- Encrypt data in transit with SSL
  - From clients to MariaDB
  - From clients to MariaDB MaxScale
  - Between MariaDB replicated servers

Best Practices: Use a database proxy

- Use MariaDB MaxScale as a database firewall
- Restrict the operations that clients are allowed to perform
- Identify and flag potentially dangerous queries
- Customize rules about what’s allowed and what’s not
- Implement connection pooling capabilities

Best Practices: User Management

- Protect MariaDB data and backups via OS permissions
- Use strong passwords
- Allow root access to MariaDB only from local clients—no administrative access over the network
- Use a separate MariaDB user account for each of your applications
- Allow access from a minimal set of IP addresses
- Regularly audit your users and grants

Best Practices: Auditing

- Use MariaDB Audit Plugin to log events to syslog or files
- Ensure regulatory compliance with robust logging
- Record connections, query executions, and tables accessed
- Be selective in what your are monitoring

- Plan auditing resources
  - Budget
  - Processes
- Consider using “Honeypots”
- Have a process to review the logs and follow it… Very Important
- Audit your auditing

MariaDB Server

Security Features

Authentication

Password Validation

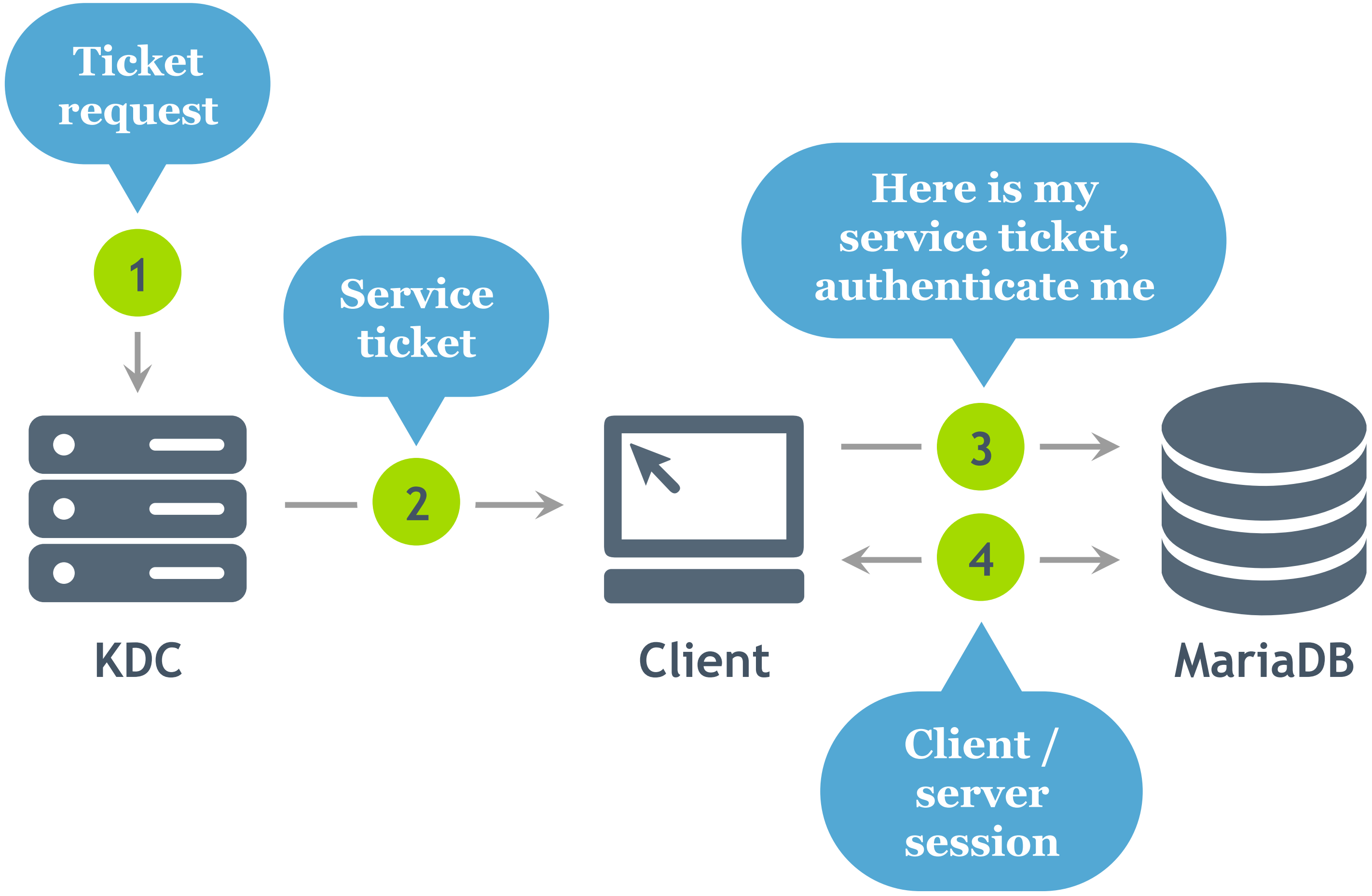
Simple_password_check plugin

Enforce a minimum password length and type/number of characters to be used

External Authentication

- Single Sign On is becoming mandatory in many Enterprises.
  - PAM-Authentication Plugin allows using /etc/shadow and any PAM based authentication like LDAP
  - Kerberos-Authentication as a standardized network authentication protocol is provided GSSAPI based on UNIX and SSPI based on Windows

MariaDB PAM Authentication



GSS-API on Linux

- Red Hat Directory Server
- OpenLDAP

SSPI on Windows

- Active Directory

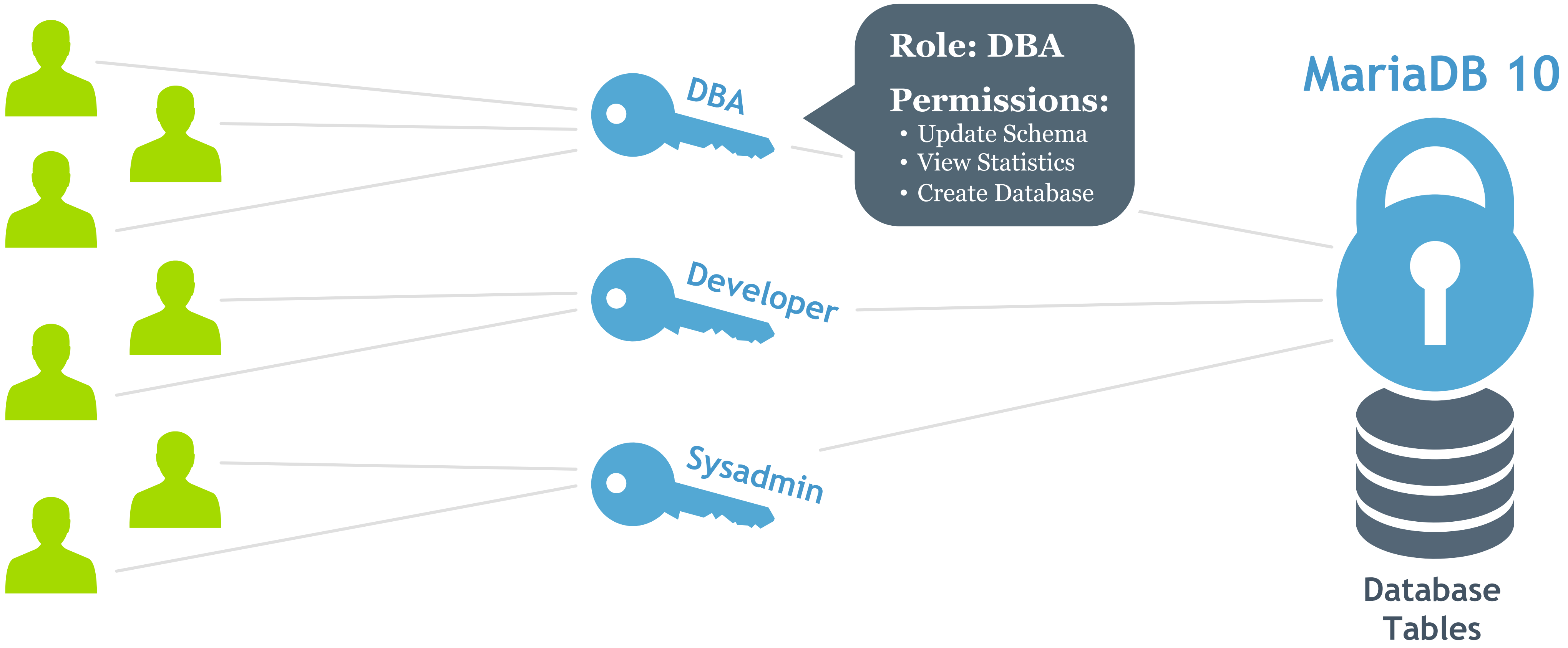
MariaDB 10.2 New User Features

The SHOW CREATE USER statement was introduced.

New CREATE USER options for limiting resource usage and tls/ssl.

New ALTER USER statement.

MariaDB Role Based Access Control



Role: DBA

Permissions:

- Update Schema
- View Statistics
- Create Database

Encryption for Data in Motion

Secured Connections

SSL Connections based on the TLSv1.2 Protocol

Between MariaDB Connectors and Server

Between MariaDB Connectors and MaxScale

SSL can also be enabled for the replication channel

Encryption

- Application control of data encryption
- Based on the AES (Advanced Encryption Standard) or DES (Data Encryption Standard) algorithm

Encryption for Data at Rest

Data-at-rest Encryption

- Everything:
  - Tables or tablespaces
  - Log files
- Independent of encryption capabilities of applications
- Based on encryption keys, key ids, key rotation and key versioning
- Low performance overhead

Key Management Services

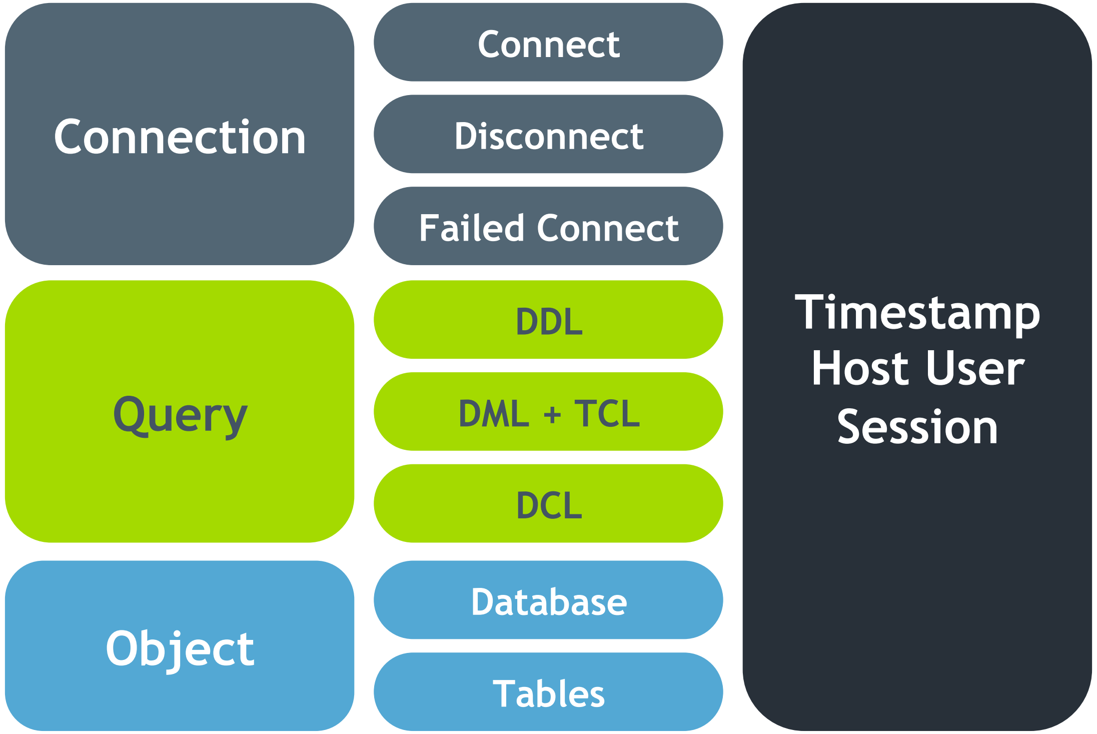
- Encryption plugin API offers choice
  - Plugin to implement the data encryption
  - Manage encryption Keys
- MariaDB Server options
  - Simple Key Management included
  - Amazon AWS KMS Plugin included
  - Eperi KMS for on premise key management – optional

Auditing for Security and Compliance

MariaDB Audit Plugin

- Logs server activity
  - Who connected to the server
  - Source of connection
  - Queries executed
  - Tables touched

- File based or syslog based logging



Connection

Connect

Disconnect

Failed Connect

Query

DDL

DML + TCL

DCL

Object

Database

Tables

Timestamp
Host User
Session

MariaDB MaxScale

Security Features

Attack Protection with MariaDB MaxScale

Database Firewall

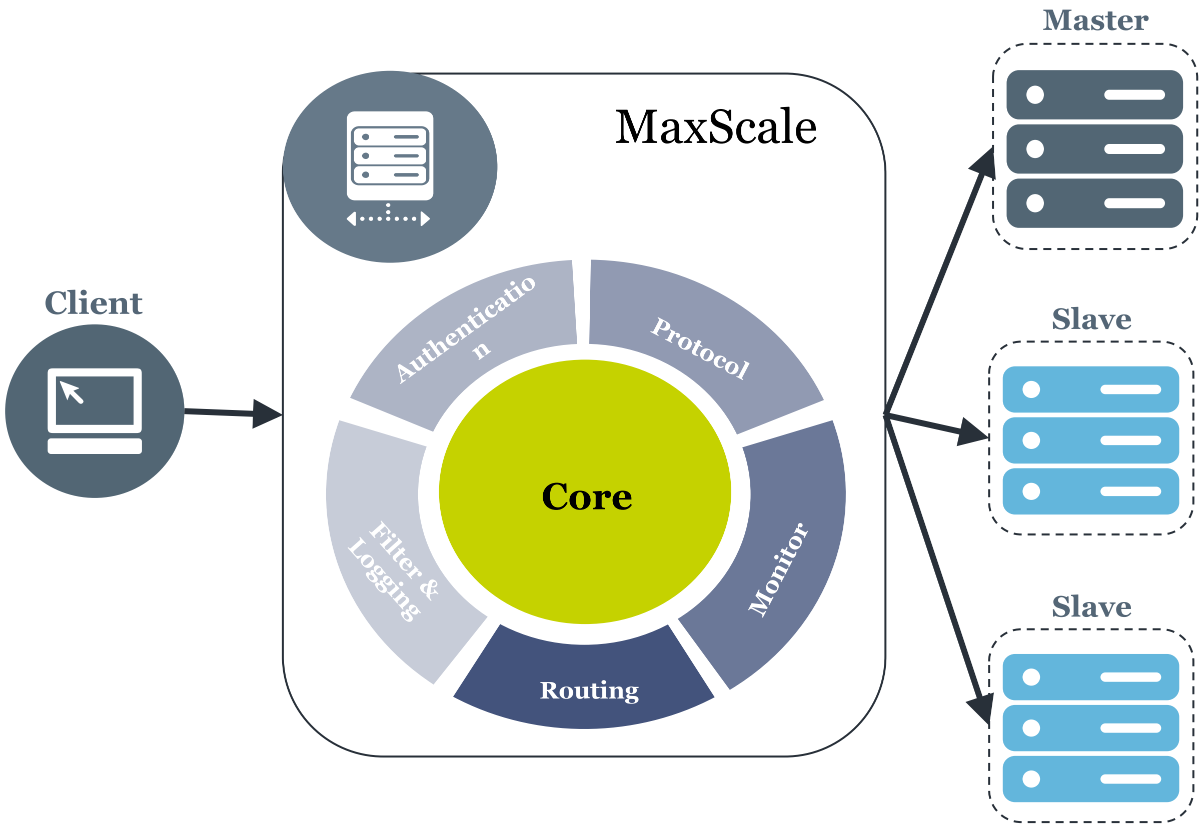
- Protects against SQL injection
- Prevents unauthorized user access and data damage
- White-list or Black-list Queries
  - Queries that match a set of rules
  - Queries matching rules for specified users
  - Queries that match certain patterns, columns, statement types
- Multiple ordered rule

Denial of Service Attack Protection

- MariaDB MaxScale Persistent Connections
- Connection pooling protects against connection surges
- Cache the connections from MaxScale to the database server
- Rate limitation
- Client multiplexing

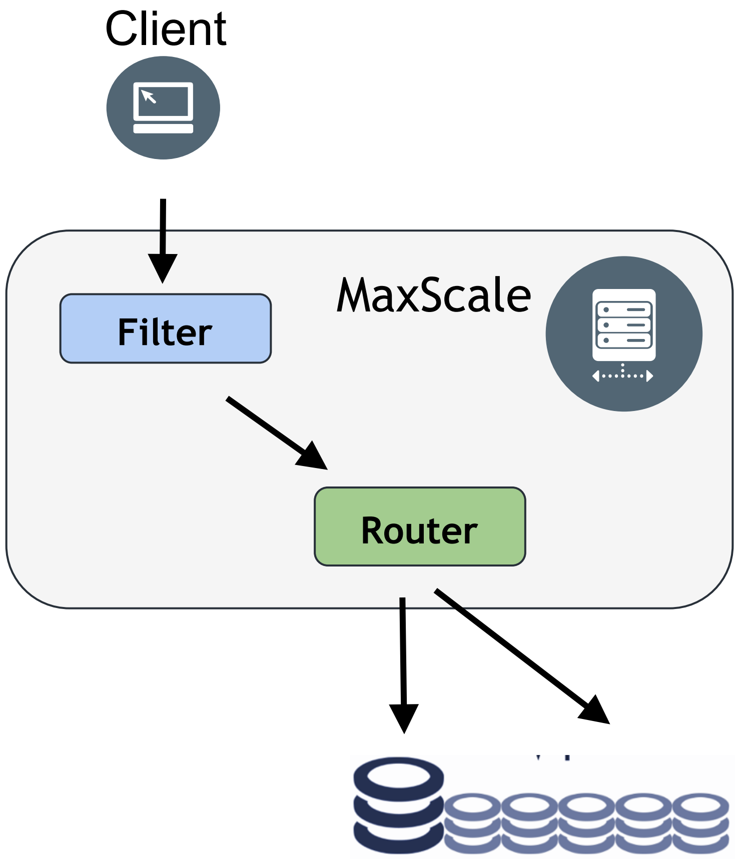
MariaDB MaxScale

- Database Proxy for
  - Security
  - Scalability
  - High Availability
  - Data Streaming
- Insulates client applications from the complexities of backend database cluster.
- Core + functionality provided by plugins
  - Protocol
  - Filters
  - Routers
  - Monitors



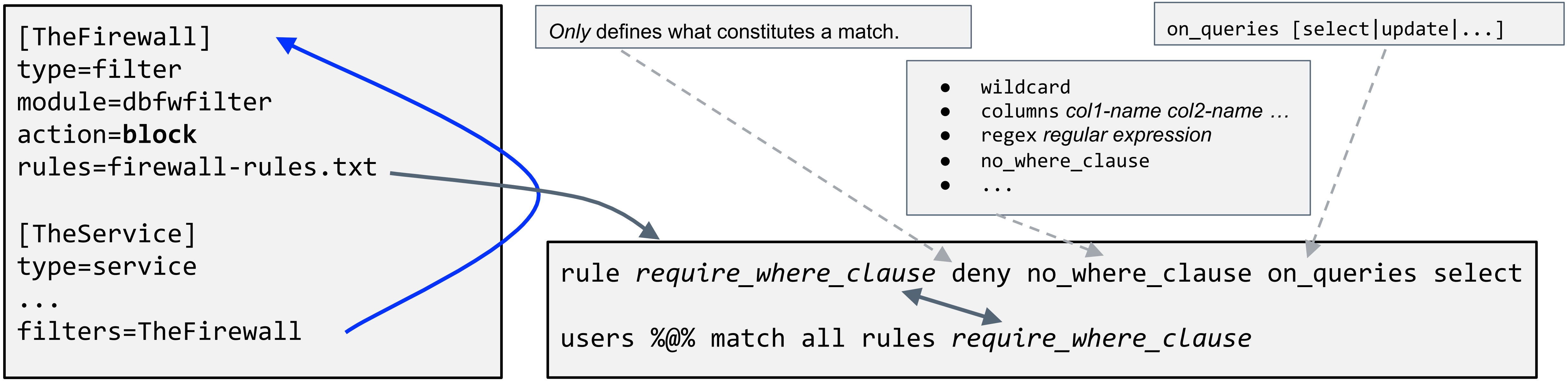
Database Firewall

- A filter installed into the request processing chain.
- Rules define what constitutes a match:
  - wildcard, columns, function, regex, no where clause
  - when to apply
  - what users are affected
  - what statements are affected
- The filter mode defines what to do with a match:
  - allow => whitelist
  - block => blacklist
- `limit_queries` rule sensible only with blacklisting
  - match if more than N queries are made within a time period



Database Firewall Example

MaxScale configuration file.



```
[TheFirewall]
type=filter
module=dbfwfilter
action=block
rules=firewall-rules.txt

[TheService]
type=service
...
filters=TheFirewall
```

Only defines what constitutes a match.

`on_queries [select|update|...]`

- `wildcard`
- `columns` *col1-name col2-name …*
- `regex` *regular expression*
- `no_where_clause`
- `...`

```
rule require_where_clause deny no_where_clause on_queries select

users %@% match all rules require_where_clause
```

```
MySQL [testdb]> select * from table;
ERROR 1141 (HY000): Access denied for user 'johan'@'127.0.0.1': Required WHERE/HAVING
clause is missing.
MySQL [testdb]>
```

Selective Data Masking

- Mask the values of certain columns.
  – Allow the use of column in a query, but do **not** return the actual value.

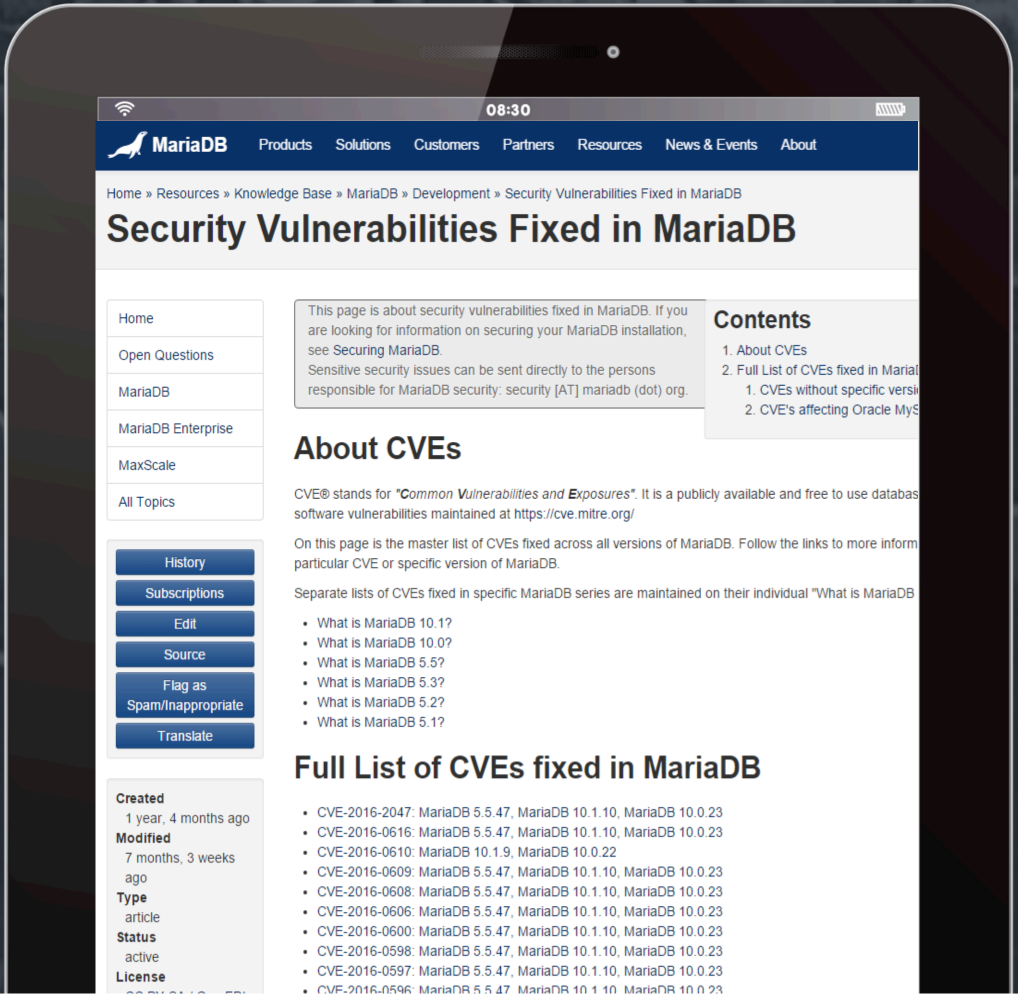
Without masking

```
> SELECT name, ssn FROM person;

+-------+-------------+
+ name  | ssn         |
+-------+-------------+
| Alice | 721-07-4426 |
| Bob   | 435-22-3267 |
...
```

With masking

```
> SELECT name, ssn FROM person;

+-------+-------------+
+ name  | ssn         |
+-------+-------------+
| Alice | XXX-XX-XXXX |
| Bob   | XXX-XX-XXXX |
...
```

MariaDB Security Gets Stronger All the Time



Home » Resources » Knowledge Base » MariaDB » Development » Security Vulnerabilities Fixed in MariaDB

Security Vulnerabilities Fixed in MariaDB

This page is about security vulnerabilities fixed in MariaDB. If you are looking for information on securing your MariaDB installation, see Securing MariaDB.
Sensitive security issues can be sent directly to the persons responsible for MariaDB security: security [AT] mariadb (dot) org.

Contents

1. About CVEs
2. Full List of CVEs fixed in MariaI
   1. CVEs without specific versi
   2. CVE's affecting Oracle MyS

About CVEs

CVE® stands for *"Common Vulnerabilities and Exposures"*. It is a publicly available and free to use databas software vulnerabilities maintained at https://cve.mitre.org/

On this page is the master list of CVEs fixed across all versions of MariaDB. Follow the links to more inform particular CVE or specific version of MariaDB.

Separate lists of CVEs fixed in specific MariaDB series are maintained on their individual "What is MariaDB

- What is MariaDB 10.1?
- What is MariaDB 10.0?
- What is MariaDB 5.5?
- What is MariaDB 5.3?
- What is MariaDB 5.2?
- What is MariaDB 5.1?

Full List of CVEs fixed in MariaDB

- CVE-2016-2047: MariaDB 5.5.47, MariaDB 10.1.10, MariaDB 10.0.23
- CVE-2016-0616: MariaDB 5.5.47, MariaDB 10.1.10, MariaDB 10.0.23
- CVE-2016-0610: MariaDB 10.1.9, MariaDB 10.0.22
- CVE-2016-0609: MariaDB 5.5.47, MariaDB 10.1.10, MariaDB 10.0.23
- CVE-2016-0608: MariaDB 5.5.47, MariaDB 10.1.10, MariaDB 10.0.23
- CVE-2016-0606: MariaDB 5.5.47, MariaDB 10.1.10, MariaDB 10.0.23
- CVE-2016-0600: MariaDB 5.5.47, MariaDB 10.1.10, MariaDB 10.0.23
- CVE-2016-0598: MariaDB 5.5.47, MariaDB 10.1.10, MariaDB 10.0.23
- CVE-2016-0597: MariaDB 5.5.47, MariaDB 10.1.10, MariaDB 10.0.23
- CVE-2016-0596: MariaDB 5.5.47, MariaDB 10.1.10, MariaDB 10.0.23

Created
1 year, 4 months ago
Modified
7 months, 3 weeks ago
Type
article
Status
active
License

MariaDB User Community

Quickly identifies new threats

Reports vulnerabilities

Creates solutions

Contributes features

Thank you