



**PERCONA**  
**LIVE**ONLINE  
**MAY 12 - 13th**  
**2021**

# MongoDB Security Features

# Hello!

## I am Jean da Silva

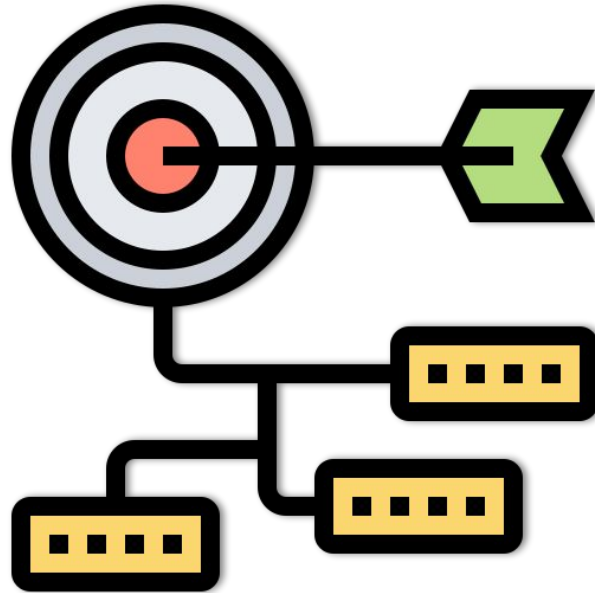
Support Engineer at Percona, student of Database Engineering and Big Data, likes to watch F1 in free time.

You can find me at [linkedin.com/in/jenunes](https://www.linkedin.com/in/jenunes)

# Agenda

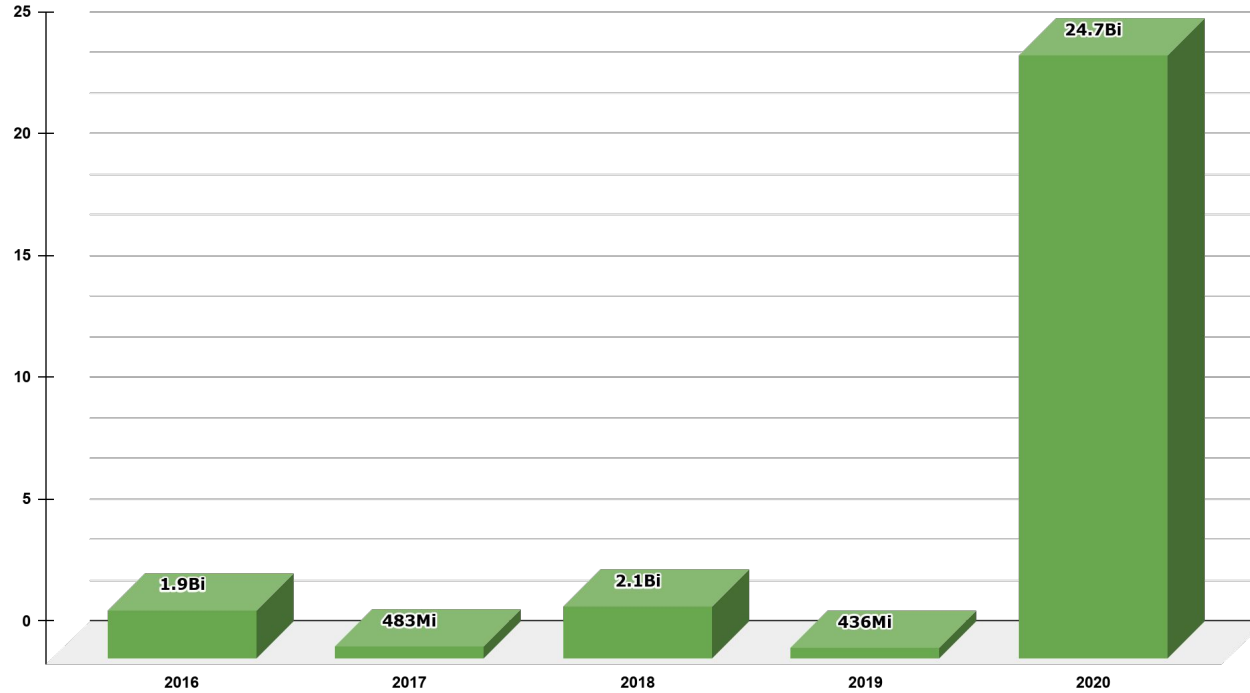
- Authorization and Authentication
- External Authentication
- TLS/SSL
- Auditing
- Log Redaction
- Encryption – Data at Rest and Client Field Encryption

# Objective



## Amount of Data Breached per Year

Top 5 only



[1] - The biggest and most impactful data breaches of 2016

[2] - The seven most colossal data breaches of 2017

[3] - Top 10 Biggest Data Breaches in 2018

[4] - The 5 biggest data hacks of 2019

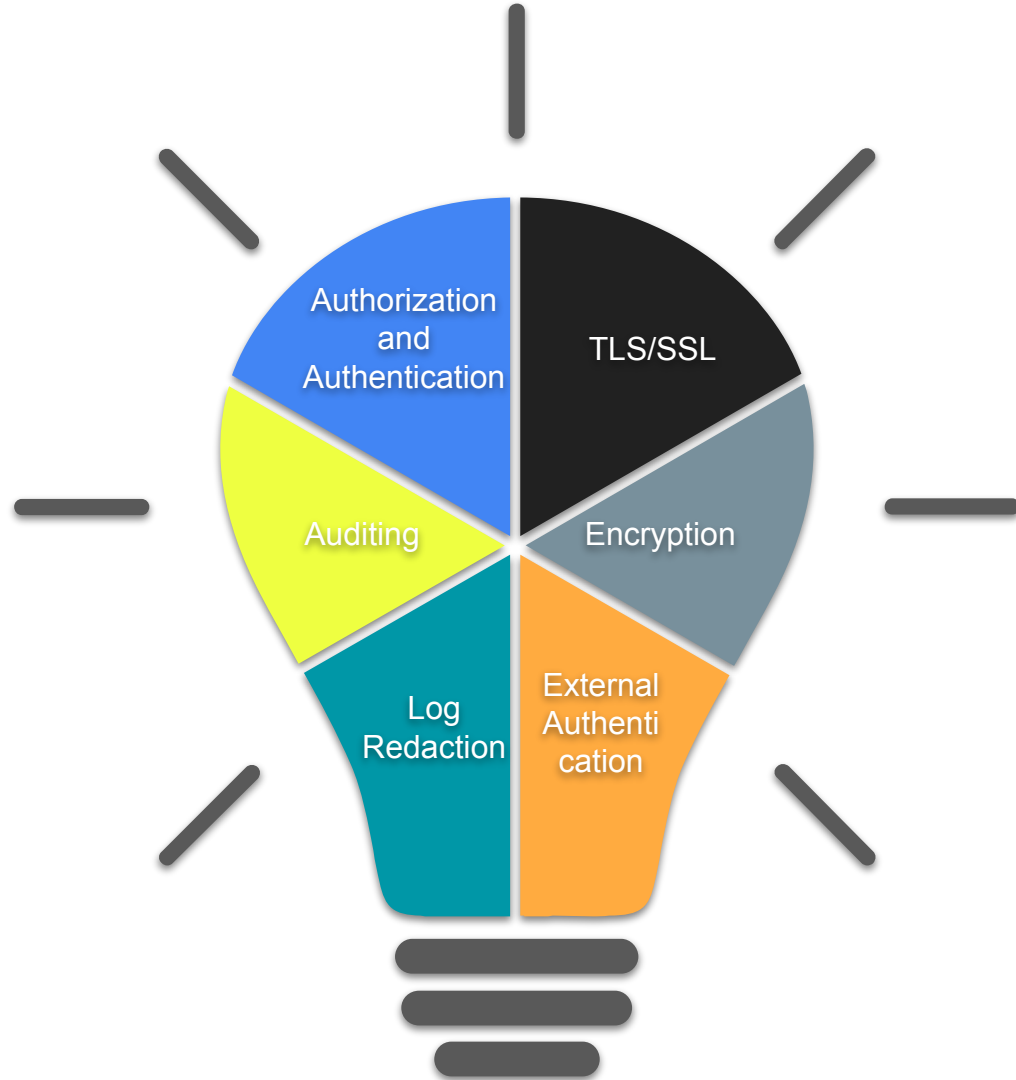
[5] - The top 10 biggest data breaches of 2020

```
# Where and how to store data.
storage:
  dbPath: /var/lib/mongo
  journal:
    enabled: true

# where to write logging data.
systemLog:
  destination: file
  logAppend: true
  path: /var/log/mongo/mongod.log

processManagement:
  fork: true
  pidFilePath: /var/run/mongod.pid

# network interfaces
net:
  port: 27017
  bindIp: 127.0.0.1
```





# 1. Authorization and Authentication

The background of the slide features abstract, wavy shapes in shades of orange and red. On the left side, there are overlapping orange shapes. On the right side, there are overlapping red and pink shapes that curve upwards towards the top right corner.

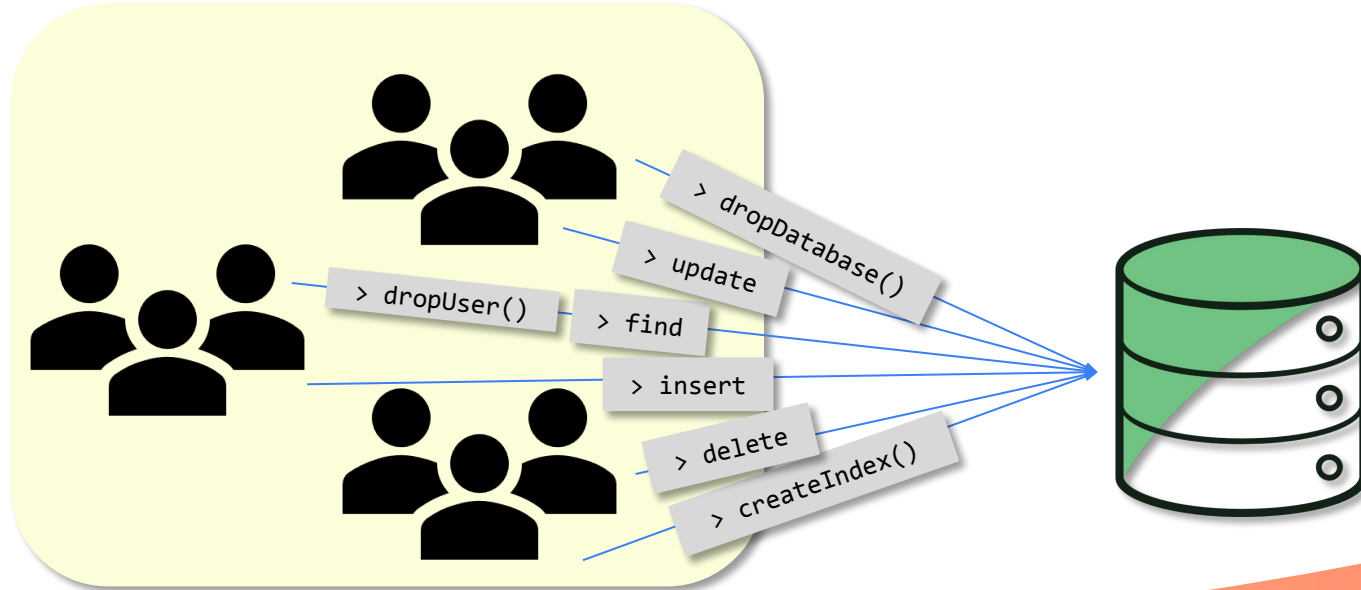
# Some Definitions First:

- **Authorization** *(A.k.a Access Control)* determines the verified user's access to resources and operations.
- **Authentication** is the process to verify the identity of a user.

Enabling **Authorization** on a MongoDB deployment **enforces authentication**, requiring users to identify themselves.

By default, **Authorization is not enabled**; we need to configure and create the root user as the first step for a secure environment.

# Who is Interested in the Default Configuration?



# Enabling Authorization and Enforcing Authentication

- Within an instance without authorization:

```
mongo> use admin

mongo> db.createUser(
  {
    user: "myUserAdmin",
    pwd: passwordPrompt(), // or cleartext password
    roles: [ { role: "userAdminAnyDatabase", db: "admin" },
"readWriteAnyDatabase" ]
  }
)
```

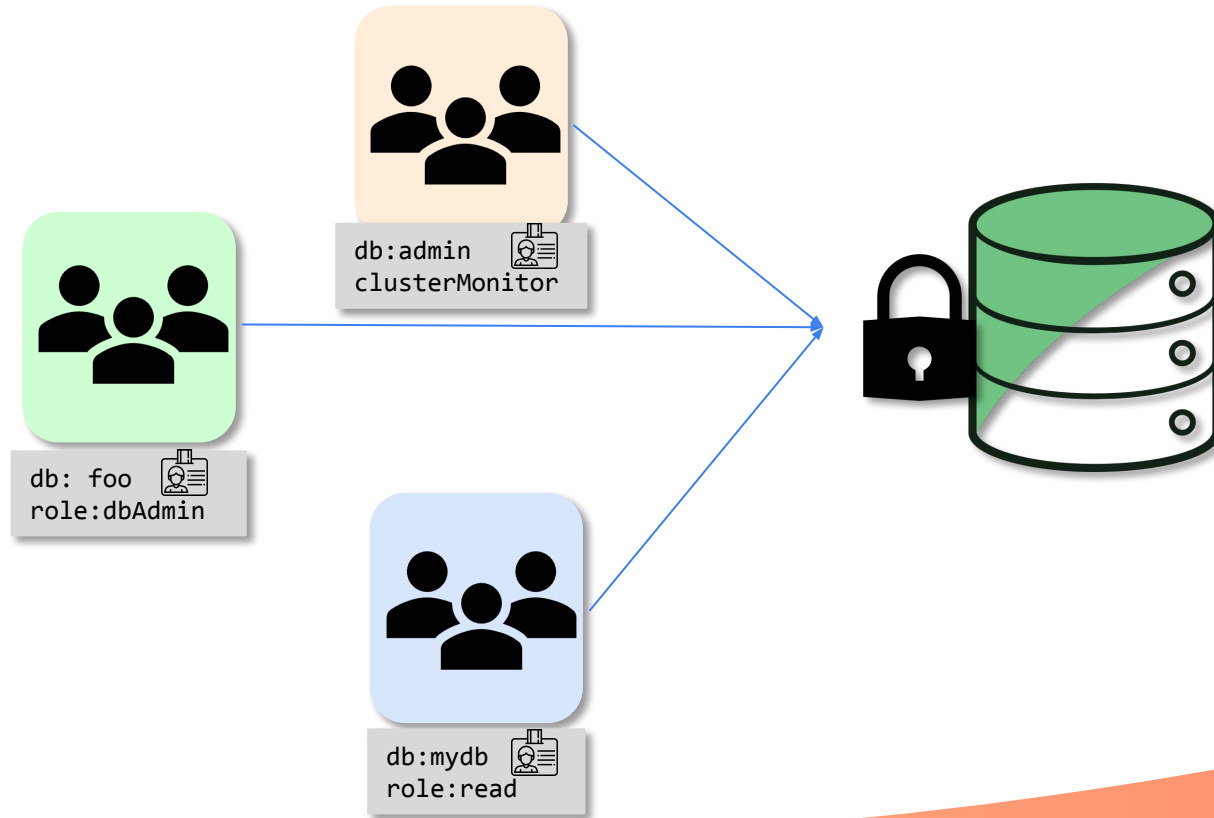
# Starting mongod with Access Control Enabled:

- If you start the **mongod** from the command line, add the **--auth** command line option:

```
mongod --auth --port 27017 --dbpath /var/lib/mongoddb
```

- If you start the **mongod** using a configuration file, add the *security.authorization* in configuration file setting:

```
security:  
  authorization: enabled
```



“

*How Does it Work for a  
ReplicaSet or Sharded Cluster?*

# Enforcing Access Control on a ReplicaSet or Sharded Cluster Requires:

- Security between components of the cluster using **Internal Authentication**.
- Security between connecting clients and the cluster using User Access Controls.



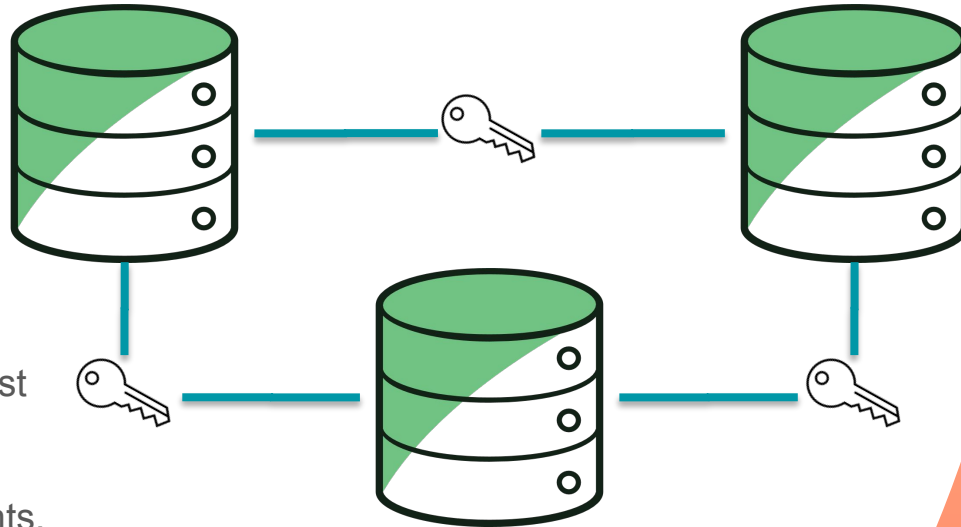
# Internal/Membership Authentication

**Enforcing** internal authentication **also** enforces user access control.

**Keyfile**(*SCRAM*) or **x.509**

Keyfiles are **bare-minimum** forms of security and are best suited for testing or development environments.

**For production** environments, x.509 certificates is recommended.



```
{  
  "t":{  
    "$date":"2021-04-17T23:06:17.487+00:00"  
  },  
  "s":"I",  
  "c":"ACCESS",  
  "id":20250,  
  "ctx":"conn49",  
  "msg":"Successful authentication",  
  "attr":{  
    "mechanism":"SCRAM-SHA-256",  
    "principalName":"__system",  
    "authenticationDatabase":"local",  
    "client":"127.0.0.1:34354"  
  }  
}
```

## 2. External Authentication

The background of the slide features abstract, wavy shapes in shades of orange and red. On the left side, there are overlapping orange shapes. On the right side, there are overlapping red and pink shapes. These shapes create a modern, flowing aesthetic.

# Overview - Authentications

- Normal Client Authentication – SCRAM<sub>(default)</sub> or x.509
- Internal Member Authentication – Keyfile<sub>(SCRAM)</sub> or x.509
- External Authentication – LDAP / ActiveDirectory or Kerberos

# Overview - Benefits



**LDAP**  
Lightweight Directory  
Access Protocol



- Ability to maintain all users in one place
- Automated policy enforcement
- Password length, complexity
- Password expiration
- Account expiration
- Failed attempts lockout

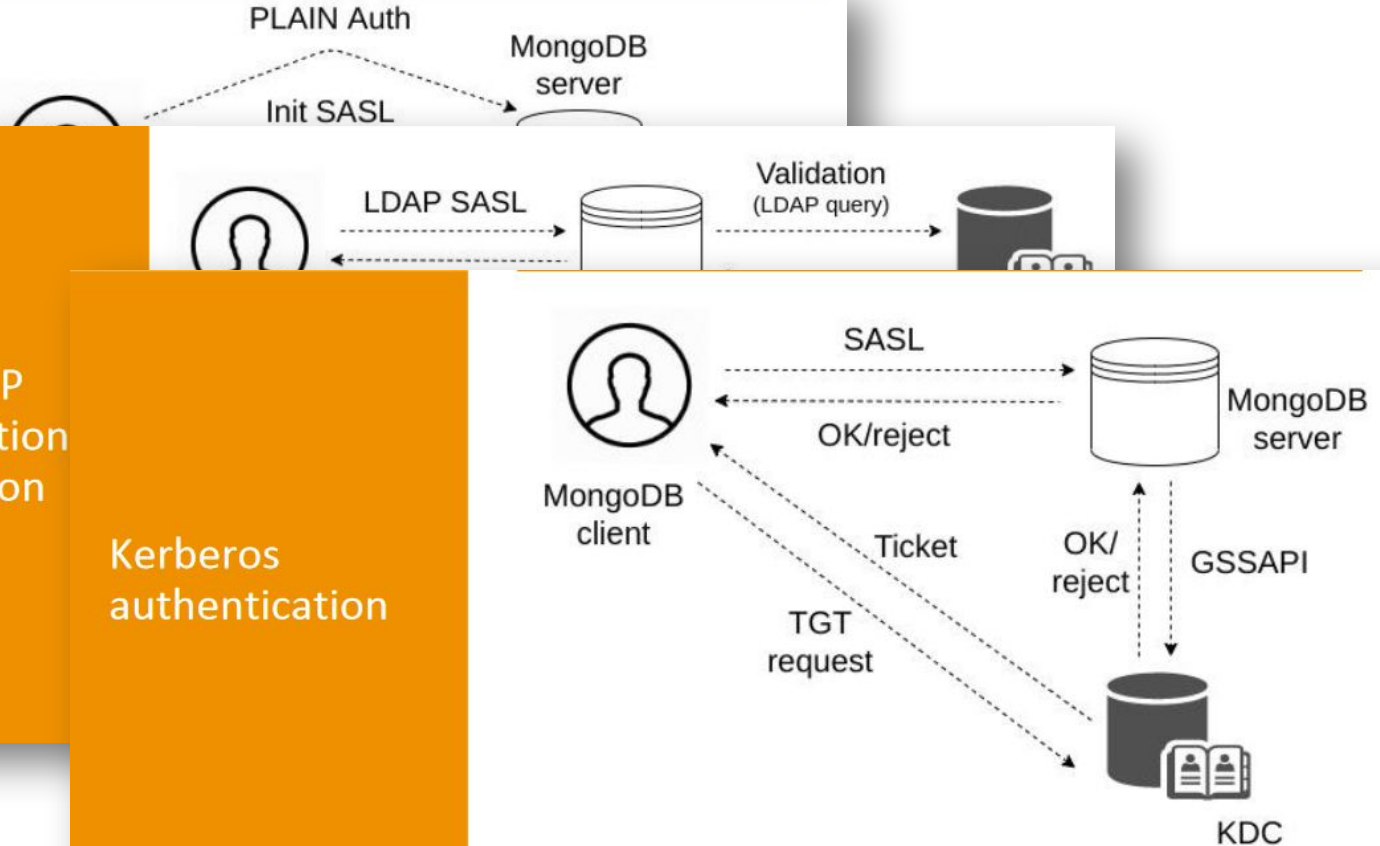
# Enterprise Authentication - Percona Server for MongoDB

Sandra Romanchenko - QA Engineer @Percona

LDAP Proxy  
Authenticat

Native LDAP  
authentication  
authorization

Kerberos  
authentication



“

*Is the External Authentication  
an Enterprise feature?*

Security	MongoDB Community	MongoDB Enterprise Advanced	Percona Server for MongoDB
LDAP Authentication	❌	✅	✅
LDAP Authorization	❌	✅	✅
Kerberos	❌	✅	✅ Starting from release 4.2.6-6



# 3. TLS/SSL

Transport Encryption

# Overview

**MongoDB traffic is not** encrypted until you create a set of TLS/SSL certificates and keys and apply them in the mongod and mongos configuration files of your entire cluster (or non-sharded replica set).



- **TLS/SSL** (Transport Layer Security/Secure Sockets Layer) to **encrypt all of MongoDB's network traffic**. TLS/SSL ensures that MongoDB network traffic is **only readable by the intended client**.
- TLS/SSL encryption only allows use of strong TLS/SSL **ciphers with a minimum of 128-bit** key length for all connections

# Network Traffic Encryption - OFF

```
mongo> use foo
switched to db foo
```

```
mongo>db.inventory.insertOne(    { item: "canvas", qty: 100, tags: ["cotton"], size: { h: 28, w:
35.5, uom: "cm" } } )
```

- Tcpdump on mongod port:

```
01:36:02.739623 IP 127.0.0.1.3017 > 127.0.0.1.42924: Flags [P.], seq 2605:3215, ack 763, win 1035,
options [nop,nop,TS val 36825751 ecr 36825751], length 610
E....;@.^.^$......D.i;.....
.1...1..b....3.....I....cursor.....firstBatch.|....0.z...._id.`~/*j7W3.s.e.item....canva
s..qty.....Y@.tags.....0....cotton...size.'....h.....<@.w.....A@.uom.....cm....1.z...._id.`~
/..#A..0....item....canvas..qty.....Y@.tags.....0....cotton...size.'....h.....<@.w.....A@.uo
m.....cm....2.z...._id.`~0.#A..0....item....canvas..qty.....Y@.tags.....0....cotton...size.'...
.h.....<@.w.....A@.uom.....cm....id.....ns.....percona.inventory...ok.....?.$clusterTim
e.X....clusterTime.....0~`.signature.3...hash.....keyId.....operation
Time.....0~`.....
```

# Certificate Management




- Using an external certificate authority or making a new root certificate just for these MongoDB clusters
- If you are using it just for the internal system authentication between mongod and mongos nodes, or if you are enabling TLS for clients too

## Important

**For production use**, your MongoDB deployment **should use valid certificates generated and signed by a certificate authority**. Although you can use self-made certificates, it is not recommended in production deployments.

# Network Traffic Encryption

- `net.ssl.mode`

Value	Description	
<i>disabled</i>	The server does <b>not</b> use TLS.	
<i>allowTLS</i>	Connections <b>between servers do not use TLS</b> . For incoming connections, the server <b>accepts both TLS and non-TLS</b> .	
<i>preferTLS</i>	Connections <b>between servers use TLS</b> . For incoming connections, the server <b>accepts both TLS and non-TLS</b> .	
<i>requireTLS</i>	The server <b>uses and accepts only TLS encrypted connections</b> .	

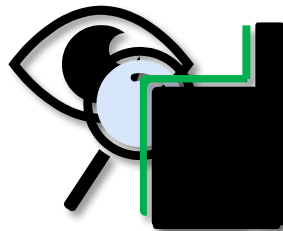
## 4. Auditing



# Overview

The auditing allows administrators and users to **track system activity** for deployments with multiple users and applications.

- Console
- Syslog
- JSON file
- BSON file



The audit feature is available on

- MongoDB Enterprise ✓
- Percona Server for MongoDB ✓

# Audit Events:

Once enabled, the auditing system can record the following operations:

- Schema changes (DDL operations);
- Topology changes on ReplicaSet and Sharded Cluster;
- Authentication failures and successes;
- Authorization changes, and failures;
- CRUD operations (requires *auditAuthorizationSuccess* set to *true*).

## Important

Enabling the `auditAuthorizationSuccess` parameter heavily impacts the performance compared to logging **only** authorization **failures**.



# Configuring Auditing

- On configuration file:

```
auditLog:  
  destination: file  
  format: JSON  
  path: /var/log/mongo/audit.json
```

- On *mongod* for command line:

```
mongod --dbpath data/db --auditDestination file --auditFormat JSON  
--auditPath /var/log/mongo/audit.json
```

# Configuring Auditing Filters

```
auditLog:
  destination: file
  format: JSON
  path: /var/log/mongo/audit.json
  filter: '{ atype: "authCheck", "param.command": { $in: [ "find", "insert",
"delete", "update", "findandmodify" ] } }'
```

- Audit log:

```
{ "atype" : "authCheck", "ts" : { "$date" : "2021-04-20T05:00:04.477+00:00" },
  "local" : { "ip" : "127.0.0.1", "port" : 37017 }, "remote" : { "ip" : "127.0.0.1",
  "port" : 39430 }, "users" : [], "roles" : [], "param" : { "command" : "insert", "ns"
  : "test.products", "args" : { "insert" : "products", "ordered" : false, "lsid" : {
  "id" : { "$binary" : "ZE6Bwd20S7iMEOXhT+vbzQ==", "$type" : "04" } }, "$db" : "test"
  } }, "result" : 0 }
```

## 5. Log Redaction



# Overview

- ***Redacts any message*** accompanying a given log event before logging.
- Prevents the **mongod** or **mongos** from writing potentially sensitive data stored on the database to the diagnostic log.
- ***Metadata*** such as ***error*** or ***operation codes***, line numbers, and source file names are ***still visible in the logs***.

The redact feature is available on

- MongoDB Enterprise ✓
- Percona Server for MongoDB ✓

# Configuring Log Redaction

Enable log redaction by adding the following to the configuration file:

```
security:  
  redactClientLogData: true
```

To enable log redaction at **runtime**, use the ***setParameter*** command:

```
db.adminCommand( { setParameter: 1, redactClientLogData : true } )
```

```
[...]  
query", "attr": {"type": "command", "ns": "test.clients", "appName": "MongoDB  
Shell", "command": {"insert": "###", "ordered": "###", "lsid": {"id": "###"}, "$db":  
[...]
```

## 6. Encryption - Data at Rest and Client Field Encryption



# Overview – Data At Rest Encryption

- Available for the **WiredTiger** Storage Engine **only**.
- All databases are encrypted when encryption is enabled, it includes admin database.
- Support AES256-CBC(*default*) and AES256-GCM as cipher mode.
- Encryption is not a part of replication; master keys and database keys are not replicated, and data is not natively encrypted over the wire.

Tip 

**Encryption at rest**, when used **in conjunction with** transport encryption(TLS/SSL) **and** good security policies. **It can help** ensure compliance with security and privacy standards, including **HIPAA**, **PCI-DSS**, and **FERPA**.

# Types of Keys – Data At Rest Encryption

- **Database keys to encrypt data.** They are stored internally, near the data that they encrypt.
- The **master key to encrypt database keys.** It is kept separately from the data and database keys and requires external management.



# Supported Key Management

- Local key management using a keyfile:  
MongoDB Enterprise and Percona Server for MongoDB
- Integration with an external key server – HashiCorp Vault:  
Percona Server for MongoDB
- Integration with **KMIP** server for key management:  
MongoDB Enterprise

“

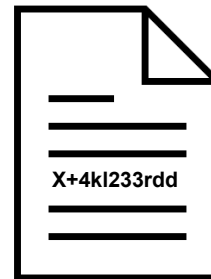
*Do I need a separate master key for each member in a replica set or sharded cluster?*

“

*How would I recover the  
database if I lose its master key?*

# Overview – Client-Side Field Encryption

- **New in version 4.2.** Applications can encrypt fields in documents prior to transmitting data over the wire to the server.
- Only applications with access to the correct encryption keys can decrypt and read the protected data.
- **Deleting an encryption key renders all data encrypted** using that key as **permanently unreadable**.



# Feature details

MongoDB provides **two methods of Field Encryption**:

- Automatic Client-Side Field Level Encryption
- Explicit (Manual) Client-Side Field Level Encryption

**The automatic mode** is available **only on the Enterprise Edition and Atlas**, while the **manual method is supported on the Community Edition** by the MongoDB drivers and mongo shell as well.



*Does Percona Server for MongoDB  
support client-side encryption?*

# How to Manually Encrypt a Field

## 1. Create an encryption key:

```
# openssl rand -hex 50 | head -c 96 | base64 | tr -d '\n' > client.key  
# chmod 600 client.key  
# chown mongod:mongod client.key
```

*A local keyfile is quick, but lower security, thus not recommended in production environment as it is stored alongside the database. For production, please consider using the one of the following services:*

As Key Management Service (KMS), MongoDB supports:

- *Amazon Web Services KMS*
- *Azure Key Vault*
- *Google Cloud Platform KMS*
- *Locally Managed Key*

## 2. Launch the mongo Shell:

```
# mongo --shell --nodb --eval "var LOCAL_KEY = cat('client.key')"
```

Percona Server for MongoDB shell version v4.4.3-5

type "help" for help

```
mongo> LOCAL_KEY
```

```
ODEyMTY2YmNmNDA4YWZlZWVhNTFmOTUyODk4YTJjODc1ODk0NTZiN2EzYWQwZDdjNmM4MDQ5ODUzYzRkMjlnNGZlM2UyZDVmMTNjZWQ1YjAyNjAwNzZmMmQ1ZjVzMzdi
```

## 3. Create the Encryption Configuration:

```
mongo> var ClientSideFieldLevelEncryptionOptions = {  
  "keyVaultNamespace" : "encryption.__dataKeys",  
  "kmsProviders" : {  
    "local" : {  
      "key" : BinData(0, LOCAL_KEY)  
    }  
  }  
}
```



#### 4. Connect with Encryption Support:

```
mongo> csfleDatabaseConnection =  
Mongo("mongodb://dba:secret@localhost:27017/?authSource=admin",  
ClientSideFieldLevelEncryptionOptions)  
connection to localhost:27017
```

#### 5. Create a Key Vault Object:

```
mongo> keyVault = csfleDatabaseConnection.getKeyVault();  
{  
  "mongo" : connection to localhost:27017,  
  "keyColl" : encryption.__dataKeys  
}
```

## 6. Create the data Encryption Key:

```
mongo> keyVault.createKey(  
  "local", /*Local-type key*/  
  "", /*Customer master key, used with external KMSes*/  
  [ "myFirstCSFLEDataKey" ]  
)  
UUID("5bd46d64-3fe8-4e31-a800-219eaa1b6a85")
```

## 7. Insert the document encrypting the field manually:

```
mongo> clientEncryption = csfleDatabaseConnection.getClientEncryption();  
mongo> var csfleDB = csfleDatabaseConnection.getDB("percona");  
mongo> csfleDB.getCollection("newcollection").insert({  
  "_id": 1,  
  "medRecNum": 1,  
  "firstName": "Jose",  
  "lastName": "Pereira",  
  "ssn": clientEncryption.encrypt(UUID("5bd46d64-3fe8-4e31-a800-219eaa1b6a85"),  
    "123-45-6789", "AEAD_AES_256_CBC_HMAC_SHA_512-Random"),  
  "comment": "Jose Pereira's SSN encrypted."});  
  
WriteResult({ "nInserted" : 1 })
```

**Perfect!** At this point, we were able to encrypt the field manually.

```
mongo> db.newcollection.find().pretty()
{
  "_id" : 1,
  "medRecNum" : 1,
  "firstName" : "Jose",
  "lastName" : "Pereira",
  "ssn" :
  BinData(6,"AkctD7WYfErwnoNer2ctYIsCVXS2nJYpSEgYF1p80RmZ1i9P0/RGELdm+XxZyN6+1s+KLeDu1L
  QFtIIJs1Bwy5AMnaA3Lf4qAfm0Nmov6Iwuqer67HV2nIQk6dIa98QFLXs="),
  "comment" : "Jose Pereira's SSN encrypted."
}
exit
```

### Important

MongoDB client-side field level encryption **only supports encrypting single fields** in a document.

“

*Can a root user be able to read  
the fields?*

“

*How does it work in a ReplicaSet  
or Sharded Cluster?*

# Summary

Security	MongoDB Community	MongoDB Enterprise Advanced	Percona Server for MongoDB
LDAP Authentication	❌	✅	✅
LDAP Authorization	❌	✅	✅
Kerberos	❌	✅	✅
Auditing	❌	✅	✅
Log Redaction	❌	✅	✅
X509 Authentication	✅	✅	✅

## MongoDB Security Features

	Authentication and Authorization	TLS/SSL network transport encryption	Encryption at rest	Client-side field level encryption (CSFLE)
<b>Database privilege abuse</b> Examples: - Violation of least-privilege (POLP) - Stolen credentials	Data access restricted based on user role. Effective when configured properly and credentials are stored securely.	Not addressed by this security feature	Not addressed by this security feature	Sensitive data is encrypted at all times on the database host and in network transit. Security responsibility is shifted to client.
<b>Network snooping</b> Examples: - Packet sniffing - IP/DNS spoofing	Not addressed by this security feature	Data is encrypted for transport over a trusted network connection.	Not addressed by this security feature	
<b>Data theft</b> Examples: - Lost or stolen physical storage media - Database file exposed		Not addressed by this security feature	Encrypted data on disk remains encrypted unless attacker has access to the encryption key(s).	
<b>Access memory of database host</b> Examples: - Memory dump analysis - RAM scraping		Not addressed by this security feature	Not addressed by this security feature	

# Thanks!



## Any questions?



# THANK YOU !



**PERCONA**  
**LIVE**ONLINE  
**MAY 12 - 13th**  
**2021**

# References:

- **Authentication/ Authorization**

<https://docs.mongodb.com/manual/tutorial/enable-authentication/>  
<https://docs.mongodb.com/manual/core/authentication/>  
<https://www.percona.com/doc/percona-server-for-mongodb/LATEST/enable-auth.html>  
<https://www.percona.com/blog/2020/08/10/securing-mongodb-top-five-security-concerns/>

- **TLS/SSL**

<https://docs.mongodb.com/manual/core/security-transport-encryption/>  
<https://www.percona.com/blog/2019/07/30/network-transport-encryption-for-mongodb/>

- **External Authentication**

<https://www.percona.com/doc/percona-server-for-mongodb/LATEST/authentication.html>  
<https://www.percona.com/resources/technical-presentations/enterprise-authentication-percona-server-mongodb-ldap-and-kerberos>

- **Auditing**

<https://www.percona.com/doc/percona-server-for-mongodb/LATEST/audit-logging.html#auditauthorizationsuccess>  
<https://docs.mongodb.com/manual/core/auditing/>  
<https://www.percona.com/blog/2017/03/03/mongodb-audit-log-why-and-how/>

- **Log Redaction**

<https://docs.mongodb.com/manual/administration/monitoring/>  
<https://www.percona.com/doc/percona-server-for-mongodb/LATEST/log-redaction.html>

- **Encryption - Data at Rest and Client Field Encryption.**

<https://www.percona.com/blog/2020/09/14/q-a-on-webinar-percona-server-for-mongodb-data-at-rest-encryption/>

- **Webinars/Presentations**

<https://www.percona.com/live/19/sessions/enhancing-the-default-mongodb-security>

- **Data Breaches**

<https://digitalguardian.com/blog/biggest-and-most-impactful-data-breaches-2016>  
<https://blog.malwarebytes.com/cybercrime/2017/12/the-seven-most-colossal-data-breaches-of-2017/>  
<https://blog.avast.com/biggest-data-breaches>  
<https://exclusive.multibriefs.com/content/the-top-10-biggest-data-breaches-of-2020/science-technology>  
<https://www.cnbc.com/2019/12/17/the-5-biggest-data-hacks-of-2019.html>