MongoDB
Security

Team Emertxe

Security

Security

MongoDB provides some security features such as authentication , access control ,encryption .

Authentication

Authentication is the process of verifying the identity of a user by accepting certain credentials like user name and password.

Varifies : Who are You in MongoDB?

User , Administrator, Monitoring agent

Authentication

Authentication is the process of verifying the identity of a user by accepting certain credentials like user name and password.

What you can do
In MongoDB?

CRUD operation,
Configure database,
Manage Sharding,
User Management

Security Checklist

Authentication Method

MongoDB provides the db.auth() method to authenticate the user.

User Management Interface

- To authenticate a client in MongoDB add a user

  > db.createUser()

  method is used to create User
- A user can have privileges grant or revoke .

db.createUser()
Syntax

```
db.createUser (  {
            user : “<name>” ,
            pwd : “<password>” ,
  customData :  { < any information > },
  roles     :  [ role  { role : “<role>” ,
                         db   :
“<database>” } | “<role>” ] } )
```

User document

Field	Type	Description
user	string	The name of new user
pwd	string	The user’s password
customData	document	Optional . This field is used to store data associated with the particular user like student id.
roles	array	The roles granted to the user. Can specify empty array [ ] to create users without roles.

Roles

Roles	Description
read	Provides the ability to read data on all non-system collections and some other collection e.g system.indexes , system.js ,system.namespaces.
readWrite	Provides the read role and ability to modify data on all non-system collections and system.js.
dbAdmin	Provides the ability to perform administrative task such as schema related task ,indexing ,gathering statistics and not for user .
dbOwner	This role combines the privileges granted by readWrite , dbAdmin and userAdmin.
userAdmin	UserAdmin role allows users to grant any privilege to any user including themselves, indirectly provides superuser access to either database or cluster.

db.createUser()
Example

```
> db.createUser ( { user  :  "studuser" ,
    pwd  : " password" ,
   roles : [ "readWrite" , "dbAdmin" ] } )
Successfully added user: { "user" : "studuser",
"roles" : [ "readWrite", "dbAdmin" ] }
```

Authentication Mechanism

MongoDB supports some authentication mechanism that client can use to verify their identity.

They are :

- SCRAM-SHA-1
- MongoDB challenge and response(MONGODB-CR)
- x.509 Certificate authentication.

Role based Access Control

The Role-based Access control is used to govern access to MongoDB.

A user is granted one or more roles that determines the user's access to database resources and operations.

We can enable authorization using –auth or the security.authorizaton setting.

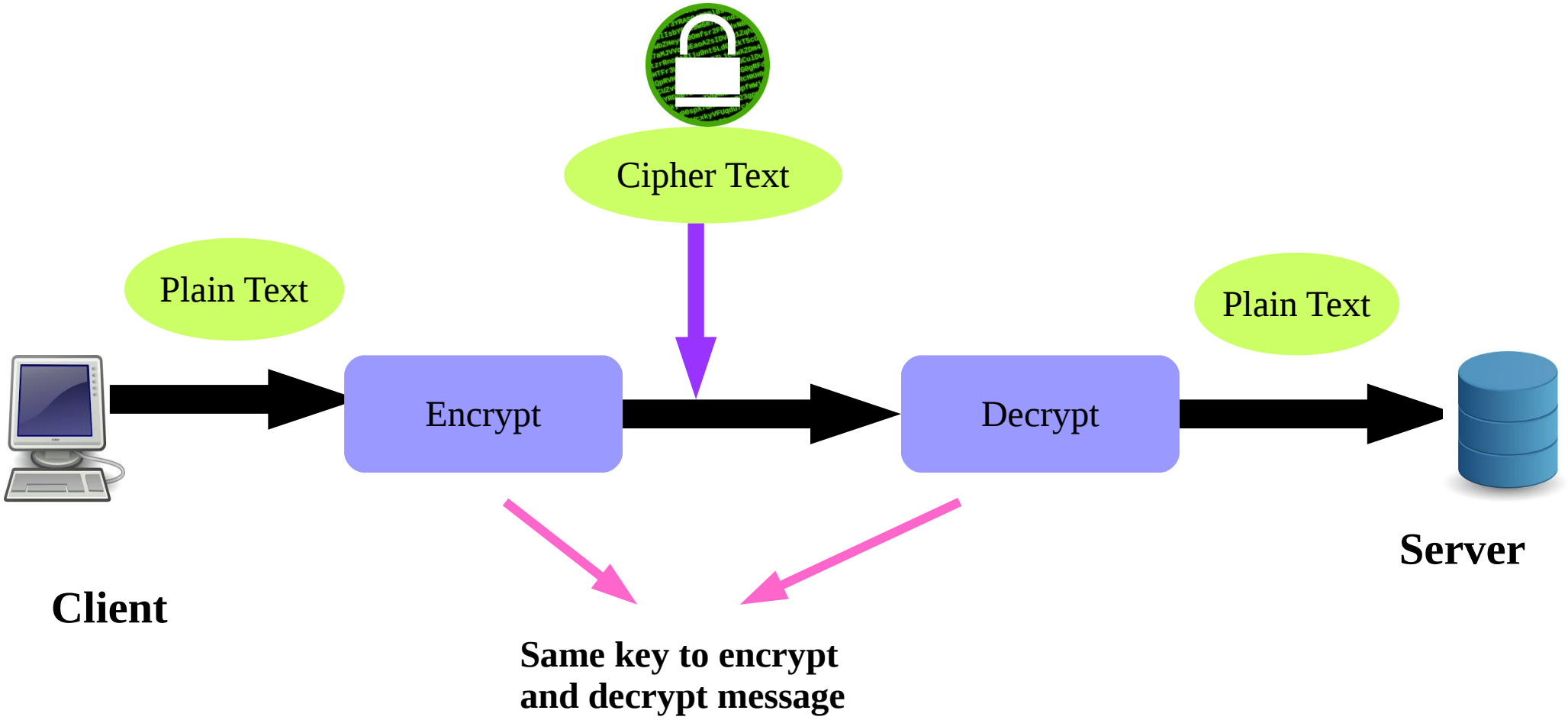
We can update existing user to grant or revoke roles.

Encryption

Encryption

Encryption is the process of encoding a message or information in such a way that only authorize party can access it.

Encryption process



EMERTXE

Transport Encryption

- TLS/SSL (Transport Security Layer / Secure Socket Layer) is used to encrypt all of MongoDB's network traffic.
- MongoDB TLS/SSL implementation uses open SSL libraries.
- SSL allows for authentication using certificates .
- MongoDB can use any valid SSL certificate issued by a certificate authority or self-signed certificate.
- Using self-signed certificate there is no server validation.
- Using certificate signed by a trusted certificate authority will permit MongoDB drivers to verify the server identity.

FIPS Mode

- Federal Information Processing Standard (FIPS) .
- U.S government computer security standards.
- Certify software modules and libraries that encrypt and decrypt data.
- We can configure MongoDB to run with a FIPS 140-2 certified library for OpenSSL.

Auditing

Auditing

The auditing facilities allows administrator to track system activity for deployments with multiple users and applications.

The audit facility can write audit events to the console, the syslog , JSON file or BSON file.

Auditing :
What You have done?

EMERTXE

Security Hardening

Security Hardening

- In computing, hardening is the process of securing a system by reducing its surface of vulnerability, which is larger when a system performs more functions.
- Security hardening in MongoDB is used to reduce the risk exposure of the entire MongoDB system and ensure that only trusted hosts have access to MongoDB.

Configuration Hardening

- Http status interface , the REST API , and JSON API are disabled to prevent potential exposure and vulnerability to attackers.
- Deprecated since version 3.2

Network Hardening

- Configure firewalls to control access to MongoDB System.
- Use of VPNs can also provide a secure channel.

References

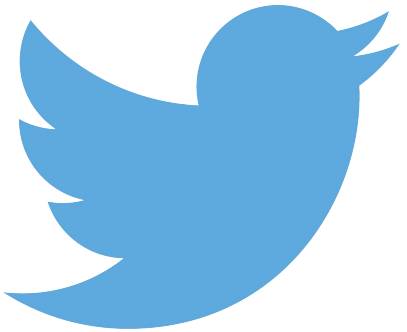
- https://www.wikimedia.org/
- https://docs.mongodb.com/manual/

Stay connected

About us: Emertxe is India's one of the top IT finishing schools & self learning kits provider. Our primary focus is on Embedded with diversification focus on Java, Oracle and Android areas

Emertxe Information Technologies,
No-1, 9th Cross, 5th Main,
Jayamahal Extension,
Bangalore, Karnataka 560046
T: +91 80 6562 9666
E: training@emertxe.com

https://www.facebook.com/Emertxe



https://twitter.com/EmertxeTweet

https://www.slideshare.net/EmertxeSlides

EMERTXE

Thank You