CIS Benchmarks

CIS MongoDB 4 Benchmark - ARCHIVE

v1.0.0 - 07-27-2021

Terms of Use

Please see the below link for our current terms of use:

https://www.cisecurity.org/cis-securesuite/cis-securesuite-membership-terms-of-use/

Table of Contents

Terms of Use 1
Overview 4
Intended Audience 4
Consensus Guidance 5
Typographical Conventions 6
Assessment Status 6
Profile Definitions 7
Acknowledgements 8
Recommendations 9
1 Installation and Patching 9
1.1 Ensure the appropriate MongoDB software version/patches are installed (Manual) 9
2 Authentication 11
2.1 Ensure Authentication is configured (Automated) 11
2.2 Ensure that MongoDB does not bypass authentication via the localhost exception (Automated) 15
2.3 Ensure authentication is enabled in the sharded cluster (Automated) 17
3 Authorization 21
3.1 Ensure least privilege for database accounts (Manual) 21
3.2 Ensure that role-based access control is enabled and configured appropriately (Manual) 23
3.3 Ensure that MongoDB is run using a non-privileged, dedicated service account (Manual) 25
3.4 Ensure that each role for each MongoDB database is needed and grants only the necessary privileges (Manual) 27
3.5 Review Superuser/Admin Roles (Manual) 29
4 Data Encryption 31
4.1 Ensure legacy TLS protocols are disabled (Automated) 31
4.2 Ensure Weak Protocols are Disabled (Automated) 33
4.3 Ensure Encryption of Data in Transit TLS or SSL (Transport Encryption) (Automated) 36

4.4 Ensure Federal Information Processing Standard (FIPS) is enabled (Automated) 39
4.5 Ensure Encryption of Data at Rest (Manual) 42
5 Audit Logging 44
5.1 Ensure that system activity is audited (Automated) 44
5.2 Ensure that audit filters are configured properly (Manual) 47
5.3 Ensure that logging captures as much information as possible (Automated) 49
5.4 Ensure that new entries are appended to the end of the log file (Automated) 51
6 Operating System Hardening 53
6.1 Ensure that MongoDB uses a non-default port (Automated) 53
6.2 Ensure that operating system resource limits are set for MongoDB (Manual) 55
6.3 Ensure that server-side scripting is disabled if not needed (Manual) 57
7 File Permissions 59
7.1 Ensure appropriate key file permissions are set (Manual) 59
7.2 Ensure appropriate database file permissions are set. (Manual) 61
Appendix: Recommendation Summary Table 63
Appendix: Change History 65

Overview

This is the archive of the CIS Benchmark for MongoDB. CIS encourages you to migrate to a more recent, supported version of this technology.

This document, CIS MongoDB 4.0 Benchmark, provides prescriptive guidance for establishing a secure configuration posture for MongoDB version/s 4.x. This guide was tested against MongoDB 4.4 running on Ubuntu Linux, Linux Red hat, and Windows but applies to other distributions as well. To obtain the latest version of this guide, please visit http://benchmarks.cisecurity.org. If you have questions, comments, or have identified ways to improve this guide, please write to us at feedback@cisecurity.org.

Extracting Running Configuration File

To verify the MongoDB running configuration file we need to connect MongoDB instance using MongoDB client with valid username/password and execute this command:

```
db.runCommand( { getCmdLineOpts: 1 } )
```

The response will contain MongoDB running configuration file location. For example:

```
"config" : "/etc/mongod.conf",
```

Important Information

- Automated Assessment Content is provided for Linux platforms only and is set to look for mongod.conf in path /etc/mongod.conf.
- Mongod: The primary daemon process for the MongoDB system. It handles data requests, manages data access, and performs background management operations.
- Mongos: mongos is a routing service for MongoDB Sharded Clusters.mongos requires mongod config, which stores the metadata of the cluster.MongoDB Shard Utility, the controller and query router for sharded clusters. Sharding partitions the data-set into discrete parts.

Intended Audience

This document is intended for system and application administrators, security specialists, auditors, help desk, and platform deployment personnel who plan to develop, deploy, assess, or secure solutions that incorporate MongoDB.

Consensus Guidance

This benchmark was created using a consensus review process comprised of subject matter experts. Consensus participants provide perspective from a diverse set of backgrounds including consulting, software development, audit and compliance, security research, operations, government, and legal.

Each CIS benchmark undergoes two phases of consensus review. The first phase occurs during initial benchmark development. During this phase, subject matter experts convene to discuss, create, and test working drafts of the benchmark. This discussion occurs until consensus has been reached on benchmark recommendations. The second phase begins after the benchmark has been published. During this phase, all feedback provided by the Internet community is reviewed by the consensus team for incorporation in the benchmark. If you are interested in participating in the consensus process, please visit https://workbench.cisecurity.org/.

Typographical Conventions

The following typographical conventions are used throughout this guide:

Convention	Meaning
`Stylized Monospace font`	Used for blocks of code, command, and script examples. Text should be interpreted exactly as presented.
Monospace font	Used for inline code, commands, or examples. Text should be interpreted exactly as presented.
<italic font in brackets>	Italic texts set in angle brackets denote a variable requiring substitution for a real value.
Italic font	Used to denote the title of a book, article, or other publication.
Note	Additional information or caveats

Assessment Status

An assessment status is included for every recommendation. The assessment status indicates whether the given recommendation can be automated or requires manual steps to implement. Both statuses are equally important and are determined and supported as defined below:

Automated

Represents recommendations for which assessment of a technical control can be fully automated and validated to a pass/fail state. Recommendations will include the necessary information to implement automation.

Manual

Represents recommendations for which assessment of a technical control cannot be fully automated and requires all or some manual steps to validate that the configured state is set as expected. The expected state can vary depending on the environment.

Profile Definitions

The following configuration profiles are defined by this Benchmark:

- **Level 1- MongoDB**

  Items in this profile apply to MongoDB and intend to:

  - be practical and prudent;
  - provide a clear security benefit; and
  - not inhibit the utility of the technology beyond acceptable means.

- **Level 2 - MongoDB**

  This profile extends the "Level 1 - MongoDB" profile. Items in this profile apply to MongoDB and exhibit one or more of the following characteristics:

  - are intended for environments or use cases where security is paramount
  - acts as defense in depth measure
  - may negatively inhibit the utility or performance of the technology.

Acknowledgements

This benchmark exemplifies the great things a community of users, vendors, and subject matter experts can accomplish through consensus collaboration. The CIS community thanks the entire consensus team with special recognition to the following individuals who contributed greatly to the creation of this guide:

Author
Vinesh Redkar Security Consultant
Pralhad Chaskar
Emad Al-Mousa

Editor
Randall Mowen

Contributor
Matthew Reagan

Recommendations

1 Installation and Patching

This section provides guidance on ensuring that the MongoDB software is up to date to eliminate easily avoidable vulnerabilities.

1.1 Ensure the appropriate MongoDB software version/patches are installed (Manual)

Profile Applicability:

- Level 1- MongoDB

Description:

The MongoDB installation version, along with the patch level, should be the most recent that is compatible with the organization's operational needs. In addition, regularly view latest minor security patch updates for security vulnerability fixes (CVE Related) from MongoDB website: https://www.mongodb.com/alerts

Rationale:

Using the most recent MongoDB software version along with all applicable patches, helps limit the possibilities for vulnerabilities in the software. The installation version and/or patches applied should be selected according to the needs of the organization. At a minimum, the software version should be supported.

Audit:

On Ubuntu:
Run the following command from within the MongoDB shell to determine if the MongoDB software version complies with your organization's operational needs:

```
> db.version()
```

On Windows:
Navigate to the Installation directory of Mongo DB on the server and run below command

```
mongod.exe --version
```

Remediation:

Upgrade to the latest version of the MongoDB software:

1. Backup the data set.
2. Download the binaries for the latest MongoDB revision from the MongoDB Download Page and store the binaries in a temporary location. The binaries download as compressed files that extract to the directory structure used by the MongoDB installation.
3. Shutdown the MongoDB instance.
4. Replace the existing MongoDB binaries with the downloaded binaries.
5. Restart the MongoDB instance.

Default Value:

Patches are not installed by default.

References:

1. https://docs.mongodb.com/v4.0/tutorial/upgrade-revision/
2. https://docs.mongodb.com/v4.0/release-notes/
3. https://www.mongodb.com/download-center#community
4. https://www.mongodb.com/support-policy
5. https://www.mongodb.com/alerts

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v7	2.2 Ensure Software is Supported by Vendor Ensure that only software applications or operating systems currently supported by the software's vendor are added to the organization's authorized software inventory. Unsupported software should be tagged as unsupported in the inventory system.	●	●	●
v6	4 Continuous Vulnerability Assessment and Remediation Continuous Vulnerability Assessment and Remediation			

2 Authentication

This section contains recommendations for requiring authentication before allowing access to the MongoDB database.

2.1 Ensure Authentication is configured (Automated)

Profile Applicability:

- Level 1- MongoDB

Description:

This setting ensures that all clients, users, servers are required to authenticate before being granted access to the MongoDB database.

Authentication is the process of verifying the identity of a client. When access control, i.e. authorization, is enabled, MongoDB requires all clients to authenticate themselves in order to determine their access.

from MongoDB documentation

> Authentication Mechanisms
>
> MongoDB supports a number of authentication mechanisms that clients can use to verify their identity. These mechanisms allow MongoDB to integrate into your existing authentication system.
>
> MongoDB supports multiple authentication mechanisms:

- SCRAM (Default)
- x.509 Certificate Authentication.

> Certificate Authority
>
> For production use, your MongoDB deployment should use valid certificates generated and signed by a certificate authority. You or your organization can generate and maintain an independent certificate authority, or use certificates generated by third-party TLS/SSL vendors.
>
> In addition to supporting the aforementioned mechanisms, MongoDB Enterprise also supports the following mechanisms:

- LDAP proxy authentication

- Kerberos authentication.

Rationale:

Failure to authenticate clients, users, servers can enable unauthorized access to the MongoDB database and can prevent tracing actions back to their sources.

It's highly recommended that password length and complexity also be in-place. When performing the traditional user/password authentication against MongoDB there is not in-place intrinsic password complexity check and there is no LOCKING mechanism with multiple failure logins. So, MongoDB is prone to brute force attacks compared to other database systems.

Audit:

Run the following command to verify whether an authorization is enabled on the MongoDB server.
On Ubuntu:

```
cat /etc/mongod.conf | grep "authorization"
```

On Windows:

```
type mongod.conf | findstr "authorization"
```

The value for `authorization` must be set to `enabled`.
To authenticate using the mongo shell use the following approach

- Use the mongo command-line authentication options (`--username`, `--password`, and `--authenticationDatabase`) when connecting to the mongod or mongos instance
  Or
- Connect first to the mongod or mongos instance, and then run the authenticate command or the `db.auth()` method against the authentication database.

Remediation:

The authentication mechanism should be implemented before anyone accesses the MongoDB Server.
To enable the authentication mechanism:

- Start the MongoDB instance without authentication.

  ```
  mongod --port 27017 --dbpath /data/db1
  ```

  Or

```
mongod.exe --port 27017 --dbpath db1
```

- Create the system user administrator, ensuring that its password meets organizationally-defined password in terms of length and complexity requirements as there is no in-place locking mechanism for multiple failed login attempts against MongoDB.

```
use admin
db.createUser(
  {
    user: "MongoAdmin",
    pwd: "password",
    roles: [ { role: "root", db: "admin" } ]
  }
)
```

- Open `mongod.conf` and change for authorization value to enabled:

```
security:
    authorization: "enabled"
```

- Restart the MongoDB instance

```
service mongod restart
```

Default Value:

By default, authorization is set to disable.

References:

1. https://docs.mongodb.com/v4.0/core/authentication/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.2 Use Unique Passwords Use unique passwords for all enterprise assets. Best practice implementation includes, at a minimum, an 8-character password for accounts using MFA and a 14-character password for accounts not using MFA.	●	●	●
v8	6.3 Require MFA for Externally-Exposed Applications Require all externally-exposed enterprise or third-party applications to enforce MFA, where supported. Enforcing MFA through a directory service or SSO provider is a satisfactory implementation of this Safeguard.		●	●

Controls Version	Control	IG 1	IG 2	IG 3
v7	16.3 Require Multi-factor Authentication Require multi-factor authentication for all user accounts, on all systems, whether managed onsite or by a third-party provider.		●	●

2.2 Ensure that MongoDB does not bypass authentication via the localhost exception (Automated)

Profile Applicability:

- Level 1- MongoDB

Description:

MongoDB should not be set to bypass authentication via the localhost exception. The localhost exception allows the user to enable authorization before creating the first user in the system. When active, the localhost exception allows all connections from the localhost interface to have full access to that instance. The exception applies only when there are no users created in the MongoDB instance.

Note: This recommendation only applies when there are no users created in the MongoDB instance.

Rationale:

Disabling this exception will prevent unauthorized local access to the MongoDB database. It will also ensure the traceability of each database activity to a specific user. Localhost Exception allows direct connect to Mongod's without any UN/PW.

Audit:

Run the following command to extract the information about `enableLocalhostAuthBypass` setting on Configuration File.
On Ubuntu:

```
cat /etc/mongod.conf |grep "enableLocalhostAuthBypass"
```

On Windows:

```
type mongod.conf | findstr "enableLocalhostAuthBypass"
```

The value for `enableLocalhostAuthBypass` must be `false`.

Remediation:

To disable local authentication on the MongoDB database.
Type OS Console Command

```
mongod --setParameter enableLocalhostAuthBypass=0
```

or

To manually configure use the `setParameter` option in the mongo configuration file to set it to `false`.

```
setParameter:
  enableLocalhostAuthBypass: false
```

Default Value:

By default, localhost exception value (`enableLocalhostAuthBypass`) is `true`.

References:

1. https://docs.mongodb.com/v4.0/reference/parameters/#param.enableLocalhostAuthBypass

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	6.3 Require MFA for Externally-Exposed Applications Require all externally-exposed enterprise or third-party applications to enforce MFA, where supported. Enforcing MFA through a directory service or SSO provider is a satisfactory implementation of this Safeguard.		●	●
v7	16.3 Require Multi-factor Authentication Require multi-factor authentication for all user accounts, on all systems, whether managed onsite or by a third-party provider.		●	●

2.3 Ensure authentication is enabled in the sharded cluster (Automated)

Profile Applicability:

- Level 1- MongoDB

Description:

Authentication is enabled in a sharded cluster when the certificate or key files are created and configured for all components. This ensures that every client that accesses the cluster must provide credentials, to include MongoDB instances that access each other within the cluster.

With keyfile authentication, each mongod or mongos instance in the sharded cluster uses the contents of the keyfile as the shared password for authenticating other members in the deployment. Only mongod or mongos instances with the correct keyfile can join the sharded cluster.

For Production Environment: x.509 certificate authentication with secure TSL/SSL connection must be used for authentication.

For Development Purpose: Key file can be used as an authentication mechanism between the shared cluster. Keyfiles are bare-minimum forms of security and are best suited for testing or development environments.

Rationale:

Enforcing a key or certificate on a sharded cluster prevents unauthorized access to the MongoDB database and provides traceability of database activities to a specific user or component. A MongoDB sharded cluster can enforce user authentication as well as internal authentication of its components to secure against unauthorized access.

Audit:

Based on recommendations

- `PEMKeyFile`, `clusterFile`, `CAFile` must be configured.
- `clusterAuthMode` should be set to `x509`
- `authenticationMechanisms` should be set to `MONGODB-X509`.

To Check That your Current MongoDB is configured for sharding setup, execute the following command:

```
sh.status()
```

OR

```
db.printShardingStatus()
```

Run the following command to verify that the certificate-based authentication is configured:
On Ubuntu:

```
cat /etc/mongod.conf | grep "PEMKeyFile"
cat /etc/mongod.conf | grep "CAFile"
cat /etc/mongod.conf | grep "clusterFile"
cat /etc/mongod.conf | grep "clusterAuthMode"
cat /etc/mongod.conf | grep "authenticationMechanisms:"
```

On Windows:

```
type mongod.conf | findstr "PEMKeyFile"
type mongod.conf | findstr "CAFile"
type mongod.conf | findstr "clusterFile"
type mongod.conf | findstr "clusterAuthMode"
type mongod.conf | findstr "authenticationMechanisms:"
```

Run the following command to verify that the `key` file-based authentication is configured: (*Only for Development Purpose*)
On Ubuntu:

```
cat /etc/mongod.conf | grep "keyFile="
```

On Windows:

```
type mongod.conf | findstr "keyFile"
```

Remediation:

To authenticate to servers, clients can use x.509 certificates instead of usernames and passwords.
MongoDB supports x.509 certificate authentication for use with a secure TLS/SSL connection. The x.509 client authentication allows clients to authenticate to servers with

certificates rather than with a username and password.
Change the configuration file `/etc/mongod.conf` on each host, adding the following rows:

```
net:
   port: 27017
   tls:
      mode: requireSSL
      PEMKeyFile: /etc/mongodb/ssl/server1.pem
      CAFile: /etc/mongodb/ssl/mongoCA.crt
      clusterFile: /etc/mongodb/ssl/server1.pem
   security:
      authorization: enabled
      clusterAuthMode: x509
```

Restart the daemon

```
sudo service mongodb restart
```

To enable authentication in the sharded cluster, perform the following steps:(*Only for Development Purpose*)

- Generate A Key File
- On each component in the shared cluster, enable authentication by editing the configuration file `/etc/mongod.conf`. Set the `keyFile` option to the key file's path and then start the component with this command:

  ```
  keyFile = /srv/mongodb/keyfile
  ```

- When starting the component, set `--keyFile` option, which is an option for both `mongos` instances and `mongod` instances. Set the `--keyFile` to the key file's path.

Default Value:

Not configured

References:

1. https://docs.mongodb.com/v4.0/tutorial/enforce-keyfile-access-control-in-existing-sharded-cluster-no-downtime/
2. https://docs.mongodb.com/v4.0/tutorial/enforce-keyfile-access-control-in-existing-sharded-cluster/
3. https://docs.mongodb.com/v4.0/tutorial/configure-x509-member-authentication/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	6.6 Establish and Maintain an Inventory of Authentication and Authorization Systems Establish and maintain an inventory of the enterprise's authentication and authorization systems, including those hosted on-site or at a remote service provider. Review and update the inventory, at a minimum, annually, or more frequently.		●	●
v7	1.8 Utilize Client Certificates to Authenticate Hardware Assets Use client certificates to authenticate hardware assets connecting to the organization's trusted network.			●

3 Authorization

MongoDB grants access to data and commands through "role-based" approach , MongoDB is shipped with built-in roles that provide the different levels of access commonly needed in a database system. In addition, you can create custom-roles.

3.1 Ensure least privilege for database accounts (Manual)

Profile Applicability:

- Level 1- MongoDB

Description:

MongoDB grants access to data and commands through "role-based" approach, MongoDB is shipped with built-in roles that provide the different levels of access commonly needed in a database system. In addition, you can create custom-roles.

The following roles provide the ability to assign any user any privilege on any database, which means that users with one of these roles can assign themselves any privilege on any database:

dbOwner role, when scoped to the admin database userAdmin role, when scoped to the admin database userAdminAnyDatabase role

Rationale:

Ensuring highly privileged Roles are granted only for database administrators, and roles are not scoped to "admin" databases will reduce attack surface and follows the least privilege principle.

Audit:

To check accounts with database roles scoped in "admin" database, use the following query:

```
db.system.users.find(
{"roles.role":{$in:["dbOwner","userAdmin","userAdminAnyDatabase"]},"roles.db"
: "admin" } )
```

Remediation:

If any accounts were listed with built in-roles:

```
dbOwner
userAdmin
userAdminAnyDatabase
```

in "admin" database role then drop them.

References:

1. https://docs.mongodb.com/v4.4/reference/built-in-roles/

3.2 Ensure that role-based access control is enabled and configured appropriately (Manual)

Profile Applicability:

- Level 1- MongoDB

Description:

Role-based access control (RBAC) is a method of regulating access to resources based on the roles of individual users within an enterprise. A user is granted one or more roles that determine the user's access to database resources and operations. Outside of role assignments, the user has no access to the system. MongoDB can use RBAC to govern access to MongoDB systems. MongoDB does not enable authorization by default.

Rationale:

When properly implemented, RBAC enables users to carry out a wide range of authorized tasks by dynamically regulating their actions according to flexible functions. This allows an organization to control employees' access to all database tables through RBAC.

Audit:

Connect to MongoDB with the appropriate privileges and run the following command:

Identify users' roles and privileges:

```
> db.getUser()
> db.getRole()
```

Verify that the appropriate role or roles have been configured for each user.

Remediation:

1. Establish roles for MongoDB.
2. Assign the appropriate privileges to each role.
3. Assign the appropriate users to each role.
4. Remove any individual privileges assigned to users that are now addressed by the roles.
5. See the reference below for more Information.

References:

1. http://docs.mongodb.org/manual/tutorial/manage-users-and-roles/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	6.8 Define and Maintain Role-Based Access Control Define and maintain role-based access control, through determining and documenting the access rights necessary for each role within the enterprise to successfully carry out its assigned duties. Perform access control reviews of enterprise assets to validate that all privileges are authorized, on a recurring schedule at a minimum annually, or more frequently.			●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

3.3 Ensure that MongoDB is run using a non-privileged, dedicated service account (Manual)

Profile Applicability:

- Level 1- MongoDB

Description:

The MongoDB service should not be run using a privileged account such as 'root' because this unnecessarily exposes the operating system to high risk.

Rationale:

Using a non-privileged, dedicated service account restricts the database from accessing the critical areas of the operating system which are not required by the MongoDB. This will also mitigate the potential for unauthorized access via a compromised, privileged account on the operating system.

Audit:

Run the following command to get listing of all mongo instances, the PID number, and the PID owner.

```
ps -ef | grep -E "mongos|mongod"
```

Remediation:

1. Create a dedicated user for performing MongoDB database activity.
2. Set the Database data files, the keyfile, and the SSL private key files to only be readable by the mongod/mongos user.
3. Set the log files to only be writable by the mongod/mongos user and readable only by root.

Default Value:

Not configured

References:

1. http://docs.mongodb.org/manual/tutorial/manage-users-and-roles/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.5 Establish and Maintain an Inventory of Service Accounts Establish and maintain an inventory of service accounts. The inventory, at a minimum, must contain department owner, review date, and purpose. Perform service account reviews to validate that all active accounts are authorized, on a recurring schedule at a minimum quarterly, or more frequently.		●	●
v7	4.3 Ensure the Use of Dedicated Administrative Accounts Ensure that all users with administrative account access use a dedicated or secondary account for elevated activities. This account should only be used for administrative activities and not internet browsing, email, or similar activities.	●	●	●

3.4 Ensure that each role for each MongoDB database is needed and grants only the necessary privileges (Manual)

Profile Applicability:

- Level 1- MongoDB

Description:

Reviewing all roles periodically and eliminating unneeded roles as well as unneeded privileges from necessary roles helps minimize the privileges that each user has.

Rationale:

Although role-based access control (RBAC) has many advantages for regulating access to resources, over time some roles may no longer be needed, and some roles may grant privileges that are no longer needed.

Audit:

Perform the following command to view all roles on the database on which the command runs, including both built-in and user-defined roles, as well as the privileges granted by each role. Ensure that only necessary roles are listed and only the necessary privileges are listed for each role.

```
db.runCommand(
   {
      rolesInfo: 1,
      showPrivileges: true,
      showBuiltinRoles: true
   }
)
```

Remediation:

To revoke specified privileges from the user-defined role on the database where the command is run. The `revokePrivilegesFromRole` command has the following syntax:

References:

1. https://docs.mongodb.com/v3.2/reference/method/db.revokePrivilegesFromRole/
2. https://docs.mongodb.com/v3.2/reference/command/revokePrivilegesFromRole/#dbcmd.revokePrivilegesFromRole

Additional Information:

You must have the `dropRole` action on a database to drop a role from that database.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	5.3 Disable Dormant Accounts Delete or disable any dormant accounts after a period of 45 days of inactivity, where supported.	●	●	●
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●

3.5 Review Superuser/Admin Roles (Manual)

Profile Applicability:

- Level 2 - MongoDB

Description:

Roles provide several advantages that make it easier to manage privileges in a database system. Security administrators can control access to their databases in a way that mirrors the structure of their organizations (they can create roles in the database that map directly to the job functions in their organizations). The assignment of privileges is simplified. Instead of granting the same set of privileges to each individual user in a particular job function, the administrator can grant this set of privileges to a role representing that job function and then grant that role to each user in that job function.

Rationale:

Reviewing the Superuser/Admin roles within a database helps minimize the possibility of privileged unwanted access.

Audit:

Superuser roles provide the ability to assign any user any privilege on any database, which means that users with one of these roles can assign themselves any privilege on any database:

```
db.runCommand( { rolesInfo: "dbOwner" } )
db.runCommand( { rolesInfo: "userAdmin" } )
db.runCommand( { rolesInfo: "userAdminAnyDatabase" } )
```

Root role provides access to the operations and all the resources of the `readWriteAnyDatabase`, `dbAdminAnyDatabase`, `userAdminAnyDatabase`, `clusterAdmin` roles, restore combined.

```
db.runCommand( { rolesInfo: "readWriteAnyDatabase" } )
db.runCommand( { rolesInfo: "dbAdminAnyDatabase" } )
db.runCommand( { rolesInfo: "userAdminAnyDatabase" } )
db.runCommand( { rolesInfo: "clusterAdmin" } )
```

Cluster Administration Roles are used for administering the whole system rather than just a single database.

```
db.runCommand( { rolesInfo: "hostManager" } )
```

Remediation:

To remove a user from one or more roles on the current database.

References:

1. https://docs.mongodb.com/v3.0/reference/built-in-roles/#built-in-roles
2. https://docs.mongodb.com/manual/reference/method/db.revokeRolesFromUser/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	6.8 Define and Maintain Role-Based Access Control Define and maintain role-based access control, through determining and documenting the access rights necessary for each role within the enterprise to successfully carry out its assigned duties. Perform access control reviews of enterprise assets to validate that all privileges are authorized, on a recurring schedule at a minimum annually, or more frequently.			●
v7	4.3 Ensure the Use of Dedicated Administrative Accounts Ensure that all users with administrative account access use a dedicated or secondary account for elevated activities. This account should only be used for administrative activities and not internet browsing, email, or similar activities.	●	●	●
v7	16.8 Disable Any Unassociated Accounts Disable any account that cannot be associated with a business process or business owner.	●	●	●

4 Data Encryption

This section contains recommendations for securing data at rest (stored) and data in motion (transiting) for MongoDB.

4.1 Ensure legacy TLS protocols are disabled (Automated)

Profile Applicability:

- Level 2 - MongoDB

Description:

Only modern TLS protocols should be enabled in MongoDB for all client connections and upstream connections. Removing legacy TLS and SSL protocols (SSL 3.0, TLS 1.0 and 1.1), and enabling emerging and stable TLS protocols (TLS 1.2, and TLS 1.3), ensures users are able to take advantage of strong security capabilities and protects them from insecure legacy protocols.

Rationale:

Why disable TLS 1.0: TLS 1.0 was deprecated from use when PCI DSS Compliance mandated that it not be used for any applications processing credit card numbers in June 2018.

Why disable TLS 1.1: Because of the increased security associated with higher versions of TLS, TLS 1.0 should be disabled.

Audit:

To verify that the server uses disables legacy TLS protocols you should check the disabledProtocols directive, run the following commands:

```
mongod --config /etc/mongod.conf
```

Review for disabledProtocols as part of the output shown below:

```
net:
  tls:
    mode: requireTLS
    certificateKeyFile: /etc/ssl/mongodb.pem
    CAFile: /etc/ssl/caToValidateClientCertificates.pem
    disabledProtocols: TLS1_0,TLS1_1
```

Remediation:

Make changes to configuration file, to configure your mongod or mongos instance to disable legacy protocols, shut down the instance and update the configuration file with the following setting:

```
net:
  tls:
    mode: requireTLS
    certificateKeyFile: /etc/ssl/mongodb.pem
    CAFile: /etc/ssl/caToValidateClientCertificates.pem
    disabledProtocols: TLS1_0,TLS1_1
```

Start mongod or mongos instance with the configuration file.

```
mongod --config /etc/mongod.conf
```

Default Value:

`TLS1_0, TLS1_1, TLS1_2`

Note: Starting in version 4.0.4 (and 3.6.9)`TLS1_3` is added to the default value.

References:

1. https://docs.mongodb.com/manual/tutorial/configure-ssl/
2. https://docs.mongodb.com/manual/reference/configuration-options/#net.tls.disabledProtocols

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	4.1 Establish and Maintain a Secure Configuration Process Establish and maintain a secure configuration process for enterprise assets (end-user devices, including portable and mobile, non-computing/IoT devices, and servers) and software (operating systems and applications). Review and update documentation annually, or when significant enterprise changes occur that could impact this Safeguard.	●	●	●
v7	18.11 Use Standard Hardening Configuration Templates for Databases For applications that rely on a database, use standard hardening configuration templates. All systems that are part of critical business processes should also be tested.		●	●

4.2 Ensure Weak Protocols are Disabled (Automated)

Profile Applicability:

- Level 1- MongoDB

Description:

Servers can be configured to disable specific TLS/SSL protocol versions which may be vulnerable to exploitation and/or lack features which improve the level of security as provided by newer versions of the protocol.

Rationale:

The TLSv1.0 protocol is vulnerable to the BEAST attack when used in CBC mode (October 2011). Unfortunately, the TLSv1.0 uses CBC modes for all of the block mode ciphers, which only leaves the RC4 streaming cipher which is also weak and is not recommended. Therefore, it is recommended that the TLSv1.0 protocol be disabled. The TLSv1.1 protocol does not support Authenticated Encryption with Associated Data (AEAD) which is designed to simultaneously provide confidentiality, integrity, and authenticity.

The NIST SP 800-52r2 guidelines for TLS configuration require that TLS 1.2 is configured with FIPS-based cipher suites be supported by all government TLS servers and clients and requires support of TLS 1.3 by January 1, 2024. A September 2018 IETF draft also depreciates the usage of TLSv1.0 and TLSv1.1 as shown in the references.

Impact:

If an attempt to connect using a disabled protocol is made the connection attempt will fail and may have unanticipated impact on clients attempting to establish the connection.

Audit:

To verify that the server disable TLS v1.0 and v1.1, run one of the following commands:

- On Ubuntu:

```
cat /etc/mongod.conf | grep –A20 ‘net’ | grep –A10 ‘ssl’ | grep
‘disabledProtocols’
```

- On Windows:

```
type mongod.conf | findstr –A20 ‘net’ | findstr –A10 ‘ssl’ | findstr
‘disabledProtocols’
```

If `TLS1_0,TLS1_1` is not included in the string returned by either of these commands this is a fail.

Remediation:

For mongod ("Primary daemon process for the MongoDB system")
In the configuration file `/etc/mongod.conf`, set the `disabledProtocols` option to to include `TLS1_0,TLS1_1`:

```
net:
  ssl:
    mode: requireSSL
    PEMKeyFile: /etc/ssl/mongodb.pem
    CAFile: /etc/ssl/caToValidateClientCertificates.pem
    disabledProtocols: TLS1_0,TLS1_1
```

And restart monogdb instance with

```
mongod --config /etc/mongod.conf
```

Or

```
mongod --sslDisabledProtocols `TLS1_0,TLS1_1
```

Default Value:

`TLS1_0` if TLS 1.1+ is available on the system.

References:

1. https://docs.mongodb.com/v4.0/tutorial/configure-ssl/#disallow-protocols
2. https://docs.mongodb.com/v4.0/reference/program/mongod/#cmdoption-mongod-ssldisabledprotocols

Additional Information:

On macOS, you cannot disable `TLS1_1` and leave both `TLS1_0` and `TLS1_2` enabled. You must also disable at least one of the other two; for example, `TLS1_0,TLS1_1`.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v7	14.4 Encrypt All Sensitive Information in Transit Encrypt all sensitive information in transit.		●	●

4.3 Ensure Encryption of Data in Transit TLS or SSL (Transport Encryption) (Automated)

Profile Applicability:

- Level 1- MongoDB

Description:

Use TLS or SSL to protect all incoming and outgoing connections. This should include using TLS or SSL to encrypt communication between the mongod and mongos components of a MongoDB client as well as between all applications and MongoDB.

MongoDB supports 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.

Please note: As of MongoDB version 4.2 SSL has been deprecated.

Also, starting in MongoDB version 4.0, MongoDB disables support for TLS 1.0 encryption on systems where TLS 1.1+ is available.

Rationale:

This prevents sniffing of cleartext traffic between MongoDB components or performing a man-in-the-middle attack for MongoDB.

Audit:

To verify that the server requires SSL or TLS (`net.tls.mode` value set to `requireTLS`), run one of the following commands:
On Ubuntu:

```
cat /etc/mongod.conf | grep –A20 'net' | grep –A10 'tls' | grep 'mode'
```

On Windows:

```
type mongod.conf | findstr –A20 'net' | findstr –A10 'tls' | findstr 'mode'
```

Remediation:

Configure MongoDB servers to require the use of SSL or TLS to encrypt all MongoDB network communications.
To implement SSL or TLS to encrypt all MongoDB network communication, perform the

following steps:

For mongod ("Primary daemon process for the MongoDB system")

In the configuration file `/etc/mongod.conf`, set the `PEMKeyFile` option to the certificate file's path and then start the component with this command:

```
net:
   tls:
      mode: requireTLS
      certificateKeyFile: /etc/ssl/mongodb.pem
      CAFile: /etc/ssl/caToValidateClientCertificates.pem
```

And restart monogdb instance with

```
mongod --config /etc/mongod.conf
```

Default Value:

Not configured

References:

1. https://docs.mongodb.com/v4.0/core/security-transport-encryption/
2. https://docs.mongodb.com/v4.0/tutorial/configure-ssl/
3. https://docs.mongodb.com/v4.0/tutorial/configure-ssl-clients/
4. https://docs.mongodb.com/v4.0/tutorial/configure-x509-client-authentication/
5. https://docs.mongodb.com/v4.0/tutorial/configure-x509-member-authentication/
6. https://docs.mongodb.com/manual/tutorial/configure-ssl/
7. https://docs.mongodb.com/manual/reference/configuration-options/#net.tls.mode
8. https://docs.mongodb.com/manual/core/security-transport-encryption/
9. https://docs.mongodb.com/manual/reference/configuration-options/#net.tls.disabledProtocols

Additional Information:

```
------------------------------------------------------------------------------

   Value    |                            Description

------------+-----------------------------------------------------------------

disabled    |  The server does not use TLS.

------------+-----------------------------------------------------------------

allowTLS    |  Connections between servers do not use TLS. For incoming
connections,

            |   the server accepts both TLS and non-TLS.
```

```
             |
             |
------------+-------------------------------------------------------------
preferTLS    |  Connections between servers use TLS. For incoming connections,
the server
             |  accepts both TLS and non-TLS.
             |
             |
------------+-------------------------------------------------------------
requireTLS   |  The server uses and accepts only TLS encrypted connections.
             |
-------------------------------------------------------------------------------
```

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v7	14.4 Encrypt All Sensitive Information in Transit Encrypt all sensitive information in transit.		●	●
v6	14.2 Encrypt All Sensitive Information Over Less-trusted Networks All communication of sensitive information over less-trusted networks should be encrypted. Whenever information flows over a network with a lower trust level, the information should be encrypted.			

4.4 Ensure Federal Information Processing Standard (FIPS) is enabled (Automated)

Profile Applicability:

- Level 2 - MongoDB

Description:

The Federal Information Processing Standard (FIPS) is a computer security standard used to certify software modules and libraries that encrypt and decrypt data securely. You can configure MongoDB to run with a FIPS 140-2 certified library for OpenSSL.

FIPS is a property of the encryption system and not the access control system. However, the environment requires FIPS compliant encryption and access control. Organizations must ensure that the access control system uses only FIPS-compliant encryption.

Rationale:

FIPS is an industry standard which dictates how data should be encrypted at rest and during transmission.

Audit:

On Ubuntu:
To verify that the server uses FIPS Mode (`net.tls.FIPSMode` value set to `true`), run following commands:

```
mongod --config /etc/mongod.conf

net:
 tls:
 FIPSMode: true
```

Or
To verify FIPS mode is running, check the server log file for a message that FIPS is active:

```
FIPS 140-2 mode activated
```

On Windows:
Check `FIPSMode` is `true`

```
type mongod.conf | findstr "FIPSMode"
```

Remediation:

Configuring FIPS mode, ensure that your certificate is FIPS compliant. Run mongod or mongos instance in FIPS mode.
Make changes to configuration file, to configure your mongod or mongos instance to use FIPS mode, shut down the instance and update the configuration file with the following setting:

```
net:
   tls:
      FIPSMode: true
```

Start mongod or mongos instance with a configuration file.

```
mongod --config /etc/mongod.conf
```

Default Value:

Not configured

References:

1. https://docs.mongodb.com/v4.0/tutorial/configure-fips/
2. https://docs.mongodb.com/manual/tutorial/configure-fips/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v7	14.4 Encrypt All Sensitive Information in Transit Encrypt all sensitive information in transit.		●	●
v7	14.8 Encrypt Sensitive Information at Rest Encrypt all sensitive information at rest using a tool that requires a secondary authentication mechanism not integrated into the operating system, in order to access the information.			●
v6	14.2 Encrypt All Sensitive Information Over Less-trusted Networks All communication of sensitive information over less-trusted networks should be encrypted. Whenever information flows over a network with a lower trust level, the information should be encrypted.			
v6	14.5 Encrypt At Rest Sensitive Information Sensitive information stored on systems shall be encrypted at rest and require a secondary authentication mechanism, not integrated into the operating system, in order to access the information.			

4.5 Ensure Encryption of Data at Rest (Manual)

Profile Applicability:

- Level 2 - MongoDB

Description:

Encryption of data at rest must be enabled to ensure compliance with security and privacy standards including HIPAA, PCI-DSS, and FERPA.

Encryption at rest, when used in conjunction with transport encryption and good security policies that protect relevant accounts, passwords, and encryption keys.

Rationale:

Unauthorized users, such as intruders who are attempting security attacks, cannot read the data from storage and back up media unless they have the master encryption key to decrypt it.

Audit:

To verify that the server requires TLS (`net.tls.mode` value set to `requireTLS`), run one of the following commands:
On Ubuntu:

```
cat /etc/mongod.conf | grep --enableEncryption 'yes' | grep --
encryptionKeyFile '<path to keyfile>'
```

On Windows:

```
type mongod.conf | findstr --enableEncryption 'yes' | findstr --
encryptionKeyFile '<path to keyfile>'
```

Remediation:

It is recommended to enable the data at rest encryption to protect the data.
Protecting Data at Rest Including following steps.

- Generating a master key.
- Generating keys for each database.
- Encrypting data with the database keys.
- Encrypting the database keys with the master key.

Only the master key is external to the server and requires external management. To manage the master key, MongoDB's encrypted storage engine supports two key management options:

- Integration with a third-party key management appliance via the Key Management Interoperability Protocol (KMIP). Recommended
- Use of local key management via a keyfile.

The encryption occurs transparently in the storage layer; i.e. all data files are fully encrypted from a filesystem perspective, and data only exists in an unencrypted state in memory and during transmission.

To enable Encryption on Database follow below step mentioned in below Link

https://docs.mongodb.com/manual/tutorial/configure-encryption/

Rotation of Key is also important. This can be enabled by following mentioned steps in below link.

https://docs.mongodb.com/manual/tutorial/rotate-encryption-key/

References:

1. https://docs.mongodb.com/v4.0/core/security-encryption-at-rest/
2. https://docs.mongodb.com/v4.0/tutorial/configure-encryption/

Additional Information:

Available in MongoDB Enterprise only.

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v7	14.8 Encrypt Sensitive Information at Rest Encrypt all sensitive information at rest using a tool that requires a secondary authentication mechanism not integrated into the operating system, in order to access the information.			●

5 Audit Logging

This section contains recommendations related to configuring audit logging in MongoDB.

5.1 Ensure that system activity is audited (Automated)

Profile Applicability:

- Level 1- MongoDB

Description:

Track access and changes to database configurations and data. MongoDB Enterprise includes a system auditing facility that can record system events (e.g. user operations, connection events) on a MongoDB instance. These audit records permit forensic analysis and allow administrators to verify proper controls.

Rationale:

System level logs can be handy while troubleshooting an operational problem or handling a security incident.

Audit:

To verify that system activity is being audited for MongoDB, run the following command to confirm the `auditLog.destination` value is set correctly:
On Ubuntu:

```
cat /etc/mongod.conf |grep -A4 "auditLog" | grep "destination"
```

On Windows:

```
type mongod.conf | findstr -A4 "auditLog" | findstr "destination"
```

Remediation:

Set the value of `auditLog.destination` to the appropriate value from the following options:
syslog
To enable auditing and print audit events to syslog

```
mongod --dbpath data/db --auditDestination syslog
```

console
To enable auditing and print audit events to standard output (i.e., `stdout`)

```
mongod --dbpath data/db --auditDestination console
```

Json File
To enable auditing and print audit events to a file in JSON format. Printing audit events to file in JSON format degrades server performance more than printing to a file in BSON format.

```
mongod --dbpath data/db --auditDestination file --auditFormat JSON --
auditPath data/db/auditLog.json
```

Bson File
To enable auditing and print audit events to a file in BSON binary format

```
mongod --dbpath data/db --auditDestination file --auditFormat BSON --
auditPath data/db/auditLog.bson
```

Default Value:

Not configured

References:

1. http://docs.mongodb.org/manual/tutorial/configure-auditing/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v7	6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.	●	●	●
v7	6.3 Enable Detailed Logging Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.		●	●
v6	6.2 Ensure Audit Log Settings Support Appropriate Log Entry Formatting Validate audit log settings for each hardware device and the software installed on it, ensuring that logs include a date, timestamp, source addresses, destination addresses, and various other useful elements of each packet and/or transaction. Systems should record logs in a standardized format such as syslog entries or those outlined by the			

Controls Version	Control	IG 1	IG 2	IG 3
	Common Event Expression initiative. If systems cannot generate logs in a standardized format, log normalization tools can be deployed to convert logs into such a format.			

5.2 Ensure that audit filters are configured properly (Manual)

Profile Applicability:

- Level 2 - MongoDB

Description:

MongoDB Enterprise supports auditing of various operations. When enabled, the audit facility, by default, records all auditable operations as detailed in Audit Event Actions, Details, and Results. To specify which events to record, the audit feature includes the `--auditFilter` option. This check is only for Enterprise editions.

Rationale:

All operations carried out on the database are logged. This helps in backtracking and tracing any incident that occurs.

Audit:

To verify that audit filters are configured on MongoDB as per the organization's requirements, run the following command:
On Ubuntu:

```
cat /etc/mongod.conf |grep -A10 "auditLog" | grep "filter"
```

On Windows:

```
type mongod.conf | findstr -A10 "auditLog" | findstr "filter"
```

Remediation:

Set the audit filters based on the organization's requirements.

Default Value:

Not configured

References:

1. https://docs.mongodb.com/manual/reference/audit-message/
2. https://docs.mongodb.com/manual/tutorial/configure-audit-filters/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.2 Collect Audit Logs Collect audit logs. Ensure that logging, per the enterprise's audit log management process, has been enabled across enterprise assets.	●	●	●
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v7	6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.	●	●	●
v7	6.3 Enable Detailed Logging Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.		●	●

5.3 Ensure that logging captures as much information as possible (Automated)

Profile Applicability:

- Level 2 - MongoDB

Description:

The `SystemLog.quiet` option stops logging of information such as:

- connection events
- authentication events
- replication sync activities
- evidence of some potentially impactful commands being run (eg: `drop`, `dropIndexes`, `validate`)

This information should be logged whenever possible. This check is only for Enterprise editions.

Rationale:

The use of `SystemLog.quiet` makes troubleshooting problems and investigating possible security incidents much more difficult.

Audit:

To verify that the `SystemLog: quiet=false` option is disabled (value of `false`), run the following command:
On Ubuntu:

```
cat /etc/mongod.conf |grep "quiet"
```

On Windows:

```
type mongod.conf | findstr "quiet"
```

Remediation:

Set

```
`SystemLog:
      quiet: false`
```

to `false` in the `/etc/mongod.conf` file to disable it.

References:

1. https://docs.mongodb.com/manual/reference/configuration-options/#systemLog.quiet

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.2 Collect Audit Logs Collect audit logs. Ensure that logging, per the enterprise's audit log management process, has been enabled across enterprise assets.	●	●	●
v8	8.5 Collect Detailed Audit Logs Configure detailed audit logging for enterprise assets containing sensitive data. Include event source, date, username, timestamp, source addresses, destination addresses, and other useful elements that could assist in a forensic investigation.		●	●
v7	6.2 Activate audit logging Ensure that local logging has been enabled on all systems and networking devices.	●	●	●
v7	6.3 Enable Detailed Logging Enable system logging to include detailed information such as an event source, date, user, timestamp, source addresses, destination addresses, and other useful elements.		●	●

5.4 Ensure that new entries are appended to the end of the log file (Automated)

Profile Applicability:

- Level 2 - MongoDB

Description:

By default, new log entries will overwrite old entries after a restart of the mongod or Mongols service. Enabling the `systemLog.logAppend` setting causes new entries to be appended to the end of the log file rather than overwriting the existing content of the log when the mongos or mongod instance restarts.

Rationale:

Allowing old entries to be overwritten by new entries instead of appending new entries to the end of the log may destroy old log data that is needed for a variety of purposes.

Audit:

To verify that new log entries will be appended to the end of the log file after a restart (`systemLog: logAppend: true` value set to `true`), run the following command:
On Ubuntu:

```
cat /etc/mongod.conf | grep "logAppend"
```

On Windows:

```
type mongod.conf | findstr "logAppend"
```

Remediation:

Set

```
`systemLog:
    logAppend: true`
```

to `true` in the `/etc/mongod.conf` file.

References:

1. https://docs.mongodb.com/manual/reference/configuration-options/#systemLog.logAppend

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v8	8.10 Retain Audit Logs Retain audit logs across enterprise assets for a minimum of 90 days.		●	●
v7	6.4 Ensure adequate storage for logs Ensure that all systems that store logs have adequate storage space for the logs generated.		●	●

6 Operating System Hardening

This section contains recommendations related to hardening the operating system running below MongoDB.

6.1 Ensure that MongoDB uses a non-default port (Automated)

Profile Applicability:

- Level 1- MongoDB

Description:

Changing the default port used by MongoDB makes it harder for attackers to find the database and target it.

Rationale:

Standard ports are used in automated attacks and by attackers to verify which applications are running on a server.

Impact:

Hackers frequently scan IP addresses for commonly used ports, so it's not uncommon to use a different port to "fly under the radar". This is just to avoid detection, other than that there is no added safety by using a different port.

Audit:

To verify the port number used by MongoDB, execute the following command and ensure that the port number is not `27017`:
On Ubuntu:

```
cat /etc/mongod.conf |grep "port"
```

On Windows:

```
type mongod.conf | findstr "port"
```

Remediation:

Change the port for MongoDB server to a number other than `27017`.
In mongod.conf edit the below lines

```
# network interfaces
net:
  port: $Orginasation Defined port
  bindIp: $Orginasation Defined IP
```

References:

1. https://docs.mongodb.com/v4.0/reference/default-mongodb-port/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v7	9.2 Ensure Only Approved Ports, Protocols and Services Are Running Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.		●	●

6.2 Ensure that operating system resource limits are set for MongoDB (Manual)

Profile Applicability:

- Level 2 - MongoDB

Description:

Operating systems provide ways to limit and control the usage of system resources such as threads, files, and network connections on a per-process and per-user basis

Rationale:

These `ulimits` prevent a single user from consuming too many system resources.

Audit:

To verify the resource limits set for MongoDB, run the following commands.
Extract the process ID for MongoDB:

```
ps -ef | grep mongod
```

View the limits associated with that process number:

```
cat /proc/1322/limits
```

Remediation:

Every deployment may have unique requirements and settings. Recommended thresholds and settings are particularly important for MongoDB deployments:

- `f` (file size): unlimited
- `t` (cpu time): unlimited
- `v` (virtual memory): unlimited [1]
- `n` (open files): 64000
- `m` (memory size): unlimited [1] [2]
- `u` (processes/threads): 64000

Restart the mongod and mongos instances after changing the `ulimit` settings to ensure that the changes take effect.

Default Value:

Not configured

References:

1. https://docs.mongodb.com/v4.0/reference/ulimit/#recommended-ulimit-settings
2. https://docs.mongodb.com/manual/reference/ulimit/

6.3 Ensure that server-side scripting is disabled if not needed (Manual)

Profile Applicability:

- Level 2 - MongoDB

Description:

MongoDB supports the execution of JavaScript code for certain server-side operations: `mapReduce`, `group`, `$where`, `$accumulator`, and `$function` aggregation operations that allow users to define custom aggregation expressions. If you do not use these operations, server-side scripting should be disabled.

Rationale:

If server-side scripting is not needed and is not disabled, this introduces unnecessary risk which may allow an attacker to take advantage of insecure coding.

Impact:

Disabling server-side scripting will block all server-side scripts from executing.

Audit:

If server-side scripting is not required, verify that it is disabled (`javascriptEnabled` value of `false`) using the following command:
On Ubuntu:

```
cat /etc/mongod.conf | grep -A10 "security" | grep "javascriptEnabled"
```

On Windows:

```
type mongod.conf | findstr -A10 "security" | findstr "javascriptEnabled"
```

Remediation:

If server-side scripting is not required, for mongod instance disable it by using the `--noscripting` option on the command line, or setting `security.javascriptEnabled` to `false` in the configuration file.
Starting in MongoDB 4.4 this is also applicable to mongos.

Default Value:

Enabled

References:

1. https://docs.mongodb.com/v4.0/reference/configuration-options/#security.javascriptEnabled
2. https://docs.mongodb.com/manual/core/server-side-javascript/#disable-server-side-js
3. https://docs.mongodb.com/v4.4/core/server-side-javascript/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v7	9.2 Ensure Only Approved Ports, Protocols and Services Are Running Ensure that only network ports, protocols, and services listening on a system with validated business needs, are running on each system.		●	●
v6	18.9 Sanitize Deployed Software Of Development Artifacts For in-house developed applications, ensure that development artifacts (sample data and scripts; unused libraries, components, debug code; or tools) are not included in the deployed software, or accessible in the production environment.			

7 File Permissions

This section provides recommendations for setting permissions for the key file and the database file.

7.1 Ensure appropriate key file permissions are set (Manual)

Profile Applicability:

- Level 1- MongoDB

Description:

In the Shared Cluster, the certificate or keyfile is utilized for authentications. Implementing proper file permissions on the certificate or keyfile will prevent unauthorized access to it.

Rationale:

Protecting the certificate/keyfile strengthens authentication in the sharded cluster and prevents unauthorized access to the MongoDB database.

Audit:

Find the location of certificate/keyfile using the following commands:
On Ubuntu:

```
cat /etc/mongod.conf | grep "keyFile:"
cat /etc/mongod.conf | grep "PEMKeyFile:"
cat /etc/mongod.conf | grep "CAFile:"
```

On Windows:

```
type mongod.conf | findstr "keyFile:"
type mongod.conf | findstr "PEMKeyFile:"
type mongod.conf | findstr "CAFile:"
```

Check the permission of the file using:

```
ls -l certificate_file_locations
ls -l keyfile_locations
```

Remediation:

Set the `keyFile` ownership to mongodb user and remove other permissions by executing these commands:

```
chmod 600 /keyfile
sudo chown mongodb:mongodb /keyfile
```

Default Value:

Not configured

References:

1. https://docs.mongodb.com/v4.0/tutorial/enforce-keyfile-access-control-in-existing-replica-set/

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v7	16.4 Encrypt or Hash all Authentication Credentials Encrypt or hash with a salt all authentication credentials when stored.		●	●
v6	16.14 Encrypt/Hash All Authentication Files And Monitor Their Access Verify that all authentication files are encrypted or hashed and that these files cannot be accessed without root or administrator privileges. Audit all access to password files in the system.			

7.2 Ensure appropriate database file permissions are set. (Manual)

Profile Applicability:

- Level 1- MongoDB

Description:

MongoDB database files need to be protected using file permissions.

Rationale:

This will restrict unauthorized users from accessing the database.

Audit:

To verify that the permissions for the MongoDB database file are configured securely, run the following commands.
Find out the database location using the following command:
On Ubuntu:

```
cat /etc/mongod.conf |grep "dbpath"
or
cat /etc/mongod.conf | grep "dbPath"
```

Use the database location as part of the following command to view and verify the permissions set for the database file:
Example:

```
$ stat -c '%a' /var/lib/mongodb
```

On Windows:

```
type mongod.conf | findstr "dbpath"
```

Use the database location as part of the following command to view and verify the permissions set for the database file:

```
icacls "dbpath"
```

Remediation:

Set ownership of the database file to mongodb user and remove other permissions using the following commands:

```
chmod 770 /var/lib/mongodb
chown mongodb:mongodb /var/lib/mongodb
```

Default Value:

Not configured

References:

1. https://docs.mongodb.com/v4.0/reference/configuration-options/#storage.dbPath

CIS Controls:

Controls Version	Control	IG 1	IG 2	IG 3
v7	14.6 Protect Information through Access Control Lists Protect all information stored on systems with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.	●	●	●
v6	14.4 Protect Information With Access Control Lists All information stored on systems shall be protected with file system, network share, claims, application, or database specific access control lists. These controls will enforce the principle that only authorized individuals should have access to the information based on their need to access the information as a part of their responsibilities.			

Appendix: Recommendation Summary Table

Control		Set Correctly	
		Yes	No
1	**Installation and Patching**		
1.1	Ensure the appropriate MongoDB software version/patches are installed (Manual)	☐	☐
2	**Authentication**		
2.1	Ensure Authentication is configured (Automated)	☐	☐
2.2	Ensure that MongoDB does not bypass authentication via the localhost exception (Automated)	☐	☐
2.3	Ensure authentication is enabled in the sharded cluster (Automated)	☐	☐
3	**Authorization**		
3.1	Ensure least privilege for database accounts (Manual)	☐	☐
3.2	Ensure that role-based access control is enabled and configured appropriately (Manual)	☐	☐
3.3	Ensure that MongoDB is run using a non-privileged, dedicated service account (Manual)	☐	☐
3.4	Ensure that each role for each MongoDB database is needed and grants only the necessary privileges (Manual)	☐	☐
3.5	Review Superuser/Admin Roles (Manual)	☐	☐
4	**Data Encryption**		
4.1	Ensure legacy TLS protocols are disabled (Automated)	☐	☐
4.2	Ensure Weak Protocols are Disabled (Automated)	☐	☐
4.3	Ensure Encryption of Data in Transit TLS or SSL (Transport Encryption) (Automated)	☐	☐
4.4	Ensure Federal Information Processing Standard (FIPS) is enabled (Automated)	☐	☐
4.5	Ensure Encryption of Data at Rest (Manual)	☐	☐
5	**Audit Logging**		
5.1	Ensure that system activity is audited (Automated)	☐	☐
5.2	Ensure that audit filters are configured properly (Manual)	☐	☐
5.3	Ensure that logging captures as much information as possible (Automated)	☐	☐
5.4	Ensure that new entries are appended to the end of the log file (Automated)	☐	☐
6	**Operating System Hardening**		
6.1	Ensure that MongoDB uses a non-default port (Automated)	☐	☐

6.2	Ensure that operating system resource limits are set for MongoDB (Manual)	☐	☐
6.3	Ensure that server-side scripting is disabled if not needed (Manual)	☐	☐
7	**File Permissions**		
7.1	Ensure appropriate key file permissions are set (Manual)	☐	☐
7.2	Ensure appropriate database file permissions are set. (Manual)	☐	☐

Appendix: Change History

Date	Version	Changes for this version
JULY 27, 2021	1.0.0	Initial Release