**DATABASE CHECKLISTS**

**CIS MONGODB**

1 Installation and Patching

2 Authentication

3 Access Control

4 Data Encryption

5 Auditing

6 Operating System Hardening

7 File Permissions

*1.1 Ensure the appropriate MongoDB software version/patches are*

*Installed*

**Audit:**

Run the following command from within the MongoDB shell to determine if the MongoDB

software version complies with your organization’s operational needs:

**COMMAND: > db.version()**

**WINDOWS: 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.

*2.1 Ensure that authentication is enabled for MongoDB databases*

**Audit:**

Run the following command to verify whether authentication is enabled (Auth value set to

True) on the MongoDB server.

**COMMAND: cat /etc/mongod.conf | grep “Auth=”**

**WINDOWS: type mongod.conf | findstr “Auth=”**

**Remediation:**

The authentication mechanism should be implemented before anyone accesses the

MongoDB Server.

To enable the authentication mechanism:

• Start the MongoDB instance without authentication.

**COMMAND: mongod --port 27017 --dbpath /data/db1**

**mongod.exe --port 27017 --dbpath /data/db1**

• Create the system user administrator, ensuring that its password meets

organizationally-defined password complexity requirements.

use admin

**COMMAND: db.createUser(**

**{**

**user: "siteUserAdmin",**

**pwd: "password",**

**roles: [ { role: "userAdminAnyDatabase", db: "admin" } ]**

**}**

**)**

• Restart the MongoDB instance with authentication enabled.

**COMMAND: mongod --auth --config /etc/mongod.conf**

**mongod.exe --auth --config /etc/mongod.conf**

**Default Value:**

Not configured

*2.2 Ensure that MongoDB does not bypass authentication via the*

*localhost exception*

**Audit:**

To verify the localhost exception is disabled, run the following command to ensure

that enableLocalhostAuthBypass is set to 0 (false):

**COMMAND: cat /etc/mongod.conf |grep “enableLocalhostAuthBypass”**

**WINDOWS: type mongod.conf | findstr “enable ocalhostAuthBypass”**

**Remediation:**

Since enableLocalhostAuthBypass is not available using the setParameter database

command, use the setParameter option in the configuration file to set it to false.

setParameter:

enableLocalhostAuthBypass: false

**Default Value:**

Not configured

*2.3 Ensure authentication is enabled in the sharded cluster*

**Audit:**

Run the following command to verify that the keyfile parameter is configured:

**COMMAND: cat /etc/mongod.conf | grep “keyFile=”**

**WINDOWS: type mongod.conf | findstr “keyFile”**

**Remediation:**

To enable authentication in the sharded cluster, perform the following steps:

• Generate a key file.

http://docs.mongodb.org/v2.4/tutorial/generate-key-file/#generate-key-file

• On each component in the shared cluster, enable authentication by doing one of the

following:

o In 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

*2.4 Ensure an industry standard authentication mechanism is used*

**Audit:**

To verify the authentication mechanism in use for MongoDB, run the following commands:

**COMMAND: cat /etc/mongod.conf | grep “clusterAuthMode:”**

**COMMAND: cat /etc/mongod.conf | grep “mode:”**

**COMMAND: cat /etc/mongod.conf | grep “authorization:"**

**COMMAND: cat /etc/mongod.conf | grep “authenticationMechanisms:”**

**WINDOWS: type mongod.conf | findstr “clusterAuthMode:” type mongod.conf | findstr “mode:” type mongod.conf | findstr “authorization:" type mongod.conf | findstr “authenticationMechanisms:”**

**Remediation:**

In order to implement an industry standard authentication mechanism, use the

corresponding sample from the list below as a model for specifying the authentication

mechanisms in the MongoDB configuration file.

x.509 Certificates for Client Authentication:

security:

clusterAuthMode: x509

net:

ssl:

mode: requireSSL

PEMKeyFile: *<path to TLS/SSL certificate and key PEM file>*

CAFile: *<path to root CA PEM file>*

See the reference section for a link to a detailed procedure for generating the PEMKeyFile

and CAFile.

MongoDB with Kerberos Authentication on Linux:

security:

authorization: enabled

setParameter:

authenticationMechanisms: GSSAPI

storage:

dbPath: /opt/mongodb/data

*3.1 Ensure that role-based access control is enabled and configured*

*Appropriately*

**Audit:**

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

**COMMAND: mongo --port 27017 -u *<siteUserAdmin>* -p *<password>* --authenticationDatabase <*database***

***name>***

Identify users' roles and privileges:

**COMMAND: > db.getUser()**

**COMMAND: > 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.

*3.2 Ensure that MongoDB only listens for network connections on*

*authorized interfaces*

**Audit:**

1. Verify that network exposure is limited, review the settings in the MongoDB

configuration file:

**COMMAND: cat /etc/mongod.conf |grep –A12 “net” | grep “bindIp“**

2. Verify the relevant network settings on the Linux system itself:

**COMMAND: iptables –L**

**WINDOWS: type mongod.conf | findstr “bindIp”**

**Remediation:**

Configure the MongoDB configuration file to limit its exposure to only the network

interfaces on which MongoDB instances should listen for incoming connections.

**Default Value:**

Not configured

*3.3 Ensure that MongoDB is run using a non-privileged, dedicated*

*service account*

**Audit:**

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

PID owner.

**COMMAND: 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

*3.4 Ensure that each role for each MongoDB database is needed and*

*grants only the necessary privileges*

**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.

**COMMAND: 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:

{

revokePrivilegesFromRole: *"<role>*",

privileges:

[

{ resource: { *<resource>* }, actions: [ "<action>", ... ] },

...

],

}

*3.5 Review User-Defined Roles*

**Audit:**

Check each role for each database using one of the following commands.

To specify a role from the current database, specify the role by its name:

**COMMAND: db.runCommand( { rolesInfo: *"<rolename>*" } )**

To specify a role from another database, specify the role by a document that specifies the

role and database:

**COMMAND: db.runCommand( { rolesInfo: { role: *"<rolename>*", db: *"<database>*" } } )**

**Remediation:**

To remove a user from one or more roles on the current database, use the following

command:

use <dbName>

db.revokeRolesFromUser( *"<username>*", [ *<roles>* ])

*3.6 Review Superuser/Admin Roles*

**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:

**COMMAND: 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.

**COMMAND: 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.

**COMMAND: db.runCommand( { rolesInfo: "hostManager" } )**

**Remediation:**

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

use <dbName>

db.revokeRolesFromUser( *"<username>*", [ *<roles>* ])

*4.1 Ensure TLS or SSL protects all network communications*

**Audit:**

To verify that the server requires SSL or TLS use (net.ssl.mode value set to requireSSL),

run one of the following commands:

**COMMAND: mongos --config /etc/mongos.conf**

or

**COMMAND: cat /etc/mongos.conf | grep –A20 ‘net’ | grep –A10 ‘ssl’ | grep ‘mode’**

**WINDOWS: type mongos.conf | findstr –A20 ‘net’ |findstr –A10 ‘ssl’ | findstr ‘mode’**

**Remediation:**

Configure MongoDB servers to require the use of SSL or TLS to encrypt all MongoDB

network communications.

**Default Value:**

Not configured

*4.2 Ensure Federal Information Processing Standard (FIPS) is enabled*

**Audit:**

To verify that the server uses FIPS Mode (net.ssl.FIPSMode value set to true), run

following commands:

**COMMAND: mongos --config /etc/mongos.conf**

**net:**

**ssl:**

**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**

**WINDOWS: 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:

ssl:

FIPSMode: true

Start mongod or mongos instance with a configuration file.

mongod --config /etc/mongod.conf

**Default Value:**

Not configured

*5.1 Ensure that system activity is audited*

**Audit:**

To verify that system activity is being audited for MongoDB, run the following command to

confirm the auditLog.destination value is set correctly:

**COMMAND: cat /etc/mongod.conf |grep –A4 “auditLog” | grep “destination”**

**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 a 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

*5.2 Ensure that audit filters are configured properly*

**Audit:**

To verify that audit filters are configured on MongoDB as per the organization’s

requirements, run the following command:

**COMMAND: cat /etc/mongod.conf |grep –A10 “auditLog” | grep “filter”**

**WINDOWS: type mongod.conf | findstr –A10 “auditLog” | findstr “filter”**

**Remediation:**

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

**Default Value:**

Not configured

*5.3 Ensure that logging captures as much information as possible*

**Audit:**

To verify that the SystemLog.quiet option is disabled (value of False), run the following

command:

**COMMAND: cat /etc/mongod.conf |grep “SystemLog.quiet”**

**WINDOWS: type mongod.conf | findstr “SystemLog.quiet”**

**Remediation:**

Set SystemLog.quiet to False in the /etc/mongod.conf file to disable it.

*5.4 Ensure that new entries are appended to the end of the log file*

**Audit:**

To verify that new log entries will be appended to the end of the log file after a restart

(systemLog.logAppend value set to true), run the following command:

**COMMAND: cat /etc/mongod.conf |grep “systemLog.logAppend”**

**WINDOWS: type mongod.conf | findstr “systemLog.logAppend”**

**Remediation:**

Set systemLog.logAppend to true in the /etc/mongod.conf file.

*Mongodb Database Running with Least Privileges*

**Audit:**

Connect MongoDB Service

**COMMAND: mongod –port 27017 –dbpath /data/db1**

**Remediation:**

Create a user which is only used for running Mongodb and directly related processes. This user must not have administrative rights to the system. Steps to create user

**COMMAND: useradd –m –d /home/mongodb –s /bin/bash –g mongodb –u 1234 [username] –p [password]**

**COMMAND: sudo chown –R mongodb:mongodb /data/db**

*Ensure that MongoDB uses a non-default port*

**Audit:**

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

**COMMAND: cat /etc/mongod.conf |grep “port”**

**WINDOWS: type mongod.conf | findstr “port”**

**Remediation:**

Change the port for MongoDB server to a number other than 27017

**Remediation:**

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 that there is no added safety by using a different port.

*6.1 Ensure that the HTTP status interface is disabled*

**Audit:**

To verify that the HTTP status interface is disabled on MongoDB (nohttpinterface has the

value True), execute the following command:

**COMMAND: cat /etc/mongod.conf |grep “nohttpinterface”**

**nohttpinterface = False**

**WINDOWS:**

**Remediation:**

Disable the HTTP status interface by setting nohttpinterface = True in the

/etc/mongod.conf file.

**Default Value:**

Enabled

*6.3 Ensure that operating system resource limits are set for MongoDB*

**Audit:**

To verify the resource limits set for MongoDB, run the following commands.

Extract the process ID for MongoDB:

**COMMAND: 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

*6.4 Ensure that server-side scripting is disabled if not needed*

**Audit:**

If server-side scripting is not required, verify that it is disabled (javascriptEnabled value

of False) using the following command:

**COMMAND: cat /etc/mongod.conf |grep –A10 “security” | grep “javascriptEnabled”**

**WINDOWS: type mongod.conf | findstr –A10 “security” | findstr “javascriptEnabled”**

**Remediation:**

If server-side scripting is not required, disable it by using the --noscripting option on the

command line.

**Default Value:**

Enabled

*6.5 Ensure that the HTTP interface is disabled*

**Audit:**

Verify the HTTP interface is disabled (parameter value set to false) by running the

following command:

**COMMAND: cat /etc/mongod.conf |grep –A12 “net” | grep –A10 “http” | grep “enabled”**

**Remediation:**

Set the parameter value to false to disable the HTTP interface.

**Default Value:**

false

*6.6 Ensure that JSONP access via an HTTP interface is disabled*

**Audit:**

Verify that JSONP access is disabled (parameter value set to false) by running the

following command:

**COMMAND: cat /etc/mongod.conf |grep –A12 “net” | grep –A10 “http” | grep “JSONPEnabled”**

**Remediation:**

Set the parameter value to false to disable JSONP access.

**Default Value:**

false

*6.7 Ensure that the REST API is disabled*

**Audit:**

Verify the REST API is disabled (parameter value set to false) by running the following

command:

**COMMAND: cat /etc/mongod.conf |grep –A12 “net” | grep –A10 “http” | grep “RESTInterfaceEnabled”**

**Remediation:**

Set the parameter value to false to disable the REST API.

**Default Value:**

false

*7.1 Ensure that key file permissions are set correctly*

**Audit:**

To verify the permissions for the MongoDB key file, run the following command:

**COMMAND: cat /etc/mongod.conf | grep “keyFile=”**

**WINDOWS: type mongod.conf | findstr “keyFile”**

**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

*7.2 Ensure that database file permissions are set correctly*

**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:

**COMMAND: cat /etc/mongod.conf |grep “dbpath”**

**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:

**COMMAND: ls –l /var/lib/mongodb**

**WINDOWS: type mongod.conf | findstr “keyFile”**

**Remediation:**

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

the following commands:

chmod 660 /var/lib/mongodb

sudo chown mongodb:mongodb /var/lib/mongodb

**Default Value:**

Not configured

**Disable the REST interface**

The MongoDB REST interface is not recommended for production. It does not support any authentication and is turned off by default. If you’ve turned it on using the “rest” configuration option, you should turn it off for production systems.

|  |  |
| --- | --- |
|  | rest = false |

**Use key files to setup the replica set**

Specify a shared key file to enable communication between your MongoDB instances in a replica set. To enable this, add the keyfile parameter to the config file as outlined below. The contents of the file need to be the same on all the machines:

|  |  |
| --- | --- |
|  | keyFile = /srv/mongodb/keyfile |