# 1 Installation and Patching

MongoDB v3.6 with current patches is installed.

# 2 Authentication

Ensure that authentication is enabled for MongoDB databases

* Auth=True

Ensure that MongoDB does not bypass authentication via the localhost exception

* setParameter: enableLocalhostAuthBypass: false

Ensure authentication is enabled in the sharded cluster

* Generate a keyfile
* keyFile = </server/mongodb/keyfile location>

Ensure an industry standard authentication mechanism is used – recommend x.509 for cluster

* security:

clusterAuthMode: x509

* net:

ssl:

mode: requireSSL

PEMKeyFile <path to TLS certificate and key PEM file>

CAFile: <path to root CA PEM file>

# 3 Access Control

Ensure that role-based access control is enabled and configured appropriately.

* Dependent upon approved roles and users. Should only be service accounts and not individuals as we do not allow individuals into the system. All access will be via a mapping of a user request to the role.

Ensure that MongoDB only listens for network connections on authorized interfaces

* Configure via AWS Security Group specifically for mongodb
* Only allow approved MongoDB ports

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

* Use dedicated service account (mongos, mongod)
* Set database data files, keyfile, and tls private keys to only be readable by mongod/mongos accounts
* Set log files to only be writable by the mongod/mongos users and readable only by root and Splunk.

Ensure that each role for each MongoDB database is needed and grants only the necessary privileges

* Dependent upon approved roles and users. Should only be service accounts and not individuals as we do not allow individuals into the system. All access will be via a mapping of a user request to the role.

Review User-Defined Roles

* Dependent upon approved roles and users. Should only be service accounts and not individuals as we do not allow individuals into the system. All access will be via a mapping of a user request to the role.

Review Superuser/Admin Roles

* Dependent upon approved roles and users. Should only be service accounts and not individuals as we do not allow individuals into the system. All access will be via a mapping of a user request to the role.

# 4 Data Encryption

All network communications shall use approved TLS ciphers

* In mongos.conf:

net:

ssl:

mode: requireSSL

PEMKeyFile: </etc/ssl/mongodb.pem>

CAFile: </etc/ssl/ca.pem>

Ensure Federal Information Processing Standard (FIPS) is enabled

* net:

ssl:

FIPSMode: true

# 5 Hardening

Perform input validation on data inserted by clients

* net.wireObjectCheck: true

# 6 Auditing

Ensure that system activity is audited

* auditLog.destination file
* auditFormat (TBD with Splunk team)
* auditPath <separate filesystem – see GS notes below>

Ensure that audit filters are configured properly

* By default all events are recorded. Filters can be used provided that Tech Risk has reviewed and approved any filters.

Ensure that logging captures as much information as possible

* Ensure that /etc/mongodb.conf -> SysteLog.quiet: false

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

* /etc/mongodb.conf -> SysteLog.logAppend: true

# 7 Operating System Hardening

Mongodb Database Running with Least Privileges

* Already addressed above (MongoDB running as non-privileged account)

Ensure that MongoDB uses a non-default port

* This can be problematic for packaged applications or ones that expect MongoDB on a specific port. Ports for Shardsvr and confgsvr do not appear to be changeable. Test thoroughly before configuring.
* /etc/mongodb.conf -> port: [choose port]

Ensure that operating system resource limits are set for MongoDB

* For systemd:
* [Service]
* LimitFSIZE=infinity
* LimitCPU=infinity
* LimitAS=infinity
* LimitMEMLOCK=infinity
* LimitNOFILE=64000
* LimitNPROC=64000

Ensure that server-side scripting is disabled if not needed

* /etc/mongodb.conf -> security.javascriptEnabled: false

# 8 File Permissions

Ensure that key file permissions are set correctly

* Keyfile permissions should be: mongodb:mongodb, 600

Ensure that database file permissions are set correctly

* Database directory and file permissions should be: mongodb:mongodb, 600

# 9 TLS Key Management

Key management for TLS will follow the model established for applications. Keys will be unique to MongoDB, but the process will be largely identical:

* Root CA private key is stored in AWS CloudHSM
* A Root certificate is signed.
* Intermediate certificates are created by service and Vault and sent to CloudHSM for signing
* CloudHSM signs the certificate, and service returns the signed Intermediate Certificate to Vault
* Intermediate keys are used to sign CSRs for microservices, Consul, Mongo, etc…
* Root Certificates are installed in each instance (and as needed microservices) upon deployment
* Intermediate Certificates and signed service and system certificates are stored in Vault and collected by service or instance (as appropriate)
* Rotation of Intermediate and service TLS certs should occur on a 12 hour basis. Intermediate and service TLS certs should last 24 hours (expire after 24 hours) to ensure overlap through rotation.
* A service on MongoDB system will interface with Consul, and upon key rotation will pull/initiate new key retrieval.
* Each MongoDB will fetch a new TLS Key pair, Intermediate Certificate, and Certificate chain from Vault (and Consul as appropriate). These artifacts will be short lived – 1 day, with new keys being created every 12 hours. This Key pair is used as TLS keys and only in memory.

# 10 Items from GS Team:

Enable all auditing for mongod, mongos

Recommended separate FS:

* Audit log fs
* Server log fs
* Swap space defined
* Operating system fs
* Backup fs for dumps/backups
* Data for mongod, mongoc and mongos on separate fs (3)

Audit logs are stored on separate filesystem. Mongo doesn’t behave if logs fill up. Logs will crash if audit logs fill/block. Audit logs are stored on separate filesystem as separate user. Logs should be written from mongo userid and rotated (moved, and permissions changed) every 5 minutes to new user that mongo cannot access.

RW and Cluster admin users inherit audit role –

User Access:

* Monitor
* Read-only to see data
* RW
* User Access role – off to the side
* DB owner