Chapter 1: Authentication

Homework 1.3: Enabling Internal Authentication using X.509

‹ Back to the Question

This is another great real world example. This is the first replica set that you set up on your own in this course, and it is your first time using helpful utilities like `openssl`.

One of the hardest parts of this lab is getting all of your `mongod` options correct. The first thing you need to do is get three servers up and running with the correct options for doing internal auth with X.509.

COPY

```
$ mkdir -p ~/M310-HW-1.3/{r0,r1,r2}

$ mongod --sslMode requireSSL --sslClusterFile ~/shared/certs/server.pem \
         --sslCAFile ~/shared/certs/ca.pem --sslPEMKeyFile ~/shared/certs/server.pem \
         --clusterAuthMode x509 --replSet HW-1.3 --dbpath ~/M310-HW-1.3/r0 \
         --logpath ~/M310-HW-1.3/r0/mongo.log --port 31130 --fork
$ mongod --sslMode requireSSL --sslClusterFile ~/shared/certs/server.pem \
         --sslCAFile ~/shared/certs/ca.pem --sslPEMKeyFile ~/shared/certs/server.pem \
         --clusterAuthMode x509 --replSet HW-1.3 --dbpath ~/M310-HW-1.3/r1 \
         --logpath ~/M310-HW-1.3/r1/mongo.log --port 31131 --fork
$ mongod --sslMode requireSSL --sslClusterFile ~/shared/certs/server.pem \
         --sslCAFile ~/shared/certs/ca.pem --sslPEMKeyFile ~/shared/certs/server.pem \
         --clusterAuthMode x509 --replSet HW-1.3 --dbpath ~/M310-HW-1.3/r2 \
         --logpath ~/M310-HW-1.3/r2/mongo.log --port 31132 --fork
```

After getting three nodes up and running with certificates you then need to connect to one of the members and initiate the replica set.

```
$ mongo --host database.m310.mongodb.university --port 31130 --ssl \
        --sslPEMKeyFile ~/shared/certs/client.pem --sslCAFile
~/shared/certs/ca.pem

rs.initiate({
  _id: 'HW-1.3',
  members: [
    { _id: 0, host: 'database.m310.mongodb.university:31130'},
    { _id: 1, host: 'database.m310.mongodb.university:31131'},
    { _id: 2, host: 'database.m310.mongodb.university:31132'},
  ]
});
```

You now have a replica set up and running with internal authentication using X.509 certificates. The last step is to create a user for `client.pem`. Here we're going to need to use `openssl` to get the subject of the certificate.

```
openssl x509 -in ~/shared/certs/client.pem -inform PEM -subjec
```

From here we know that `client.pem` has a subject of:

```
C=US,ST=New York,L=New York City,O=MongoDB,OU=University2,CN=M
```

And we can easily create an account by performing:

```
use $external
db.runCommand({
  createUser: "C=US,ST=New York,L=New York
City,O=MongoDB,OU=University2,CN=M310 Client",
  roles: [{role: 'root', db: 'admin'}]
})
```

Proceed to next section