Exercise 10: AMQ 7 Security

One of the key differences with AMQ 7 is the addition of new permissions. AMQ 6 only had 3 permissions:

1. read
2. write
3. admin

AMQ 7 on the other hand extends to 10 permissions. They are:

- createAddress
- deleteAddress
- createDurableQueue
- deleteDurableQueue
- createNonDurableQueue
- deleteNonDurableQueue
- send
- consume
- manage
- browse

With the additional 7 permissions, we have finer grain control over assigning roles to our users.

Security Labs

Permissions / Roles Lab 1

This lab demonstrates how to setup a read-only user on A-MQ i.e. the user can only consumer from a given queue.

1. Create a new broker by executing the following command:

```
./bin/artemis create
```

Name the broker "securitybroker" and give it an admin user with the credentials admin/admin.

1. `cd` to `brokers/securitybroker/bin`
2. Execute `./artemis user add --user read-only-user --password Abcd1234 --role read-only` to create a read-only user
3. `cd` to `brokers/securitybroker/etc` and open the broker.xml file.
4. Under the security-settings section, add the following text:

```
         <security-setting match="test.#">
            <permission type="createNonDurableQueue" roles="amq"/>
            <permission type="deleteNonDurableQueue" roles="amq"/>
            <permission type="createDurableQueue" roles="amq"/>
            <permission type="deleteDurableQueue" roles="amq"/>
```

```
            <permission type="createAddress" roles="amq"/>
            <permission type="deleteAddress" roles="amq"/>
            <permission type="consume" roles="read-only,amq"/>
            <permission type="browse" roles="amq"/>
            <permission type="send" roles="amq"/>
            <!-- we need this otherwise ./artemis data imp wouldn't work -->
            <permission type="manage" roles="amq"/>
         </security-setting>
```

Notice we have a specific match for any queue starting with "test.*". Also notice that we have assigned the "read-only" role to ensure that our read-only user can only consume from our test.* queue.

5. Save the broker.xml file

6. Start up our new broker using the following command: `./brokers/securitybroker/bin/artemis run`

7. Try out our new "read only" role / user by typing the following command in a separate command windo:

```
java -jar activemq-all-5.11.0.redhat-630187.jar producer --sleep 100 --
messageCount 1000 --user read-only-user --password Abcd1234 --brokerUrl
'failover:(tcp://localhost:61616,tcp://localhost:61617)' --destination
queue://test.readonly.queue
```

Notice that this command fails because our read-only-user cannot create a durable queue. This means our user is working correctly.

8. Change the command and execute the following instead to use the admin user, which coincidently has admin permissions to create and write to queues:

```
java -jar activemq-all-5.11.0.redhat-630187.jar producer --sleep 100 --
messageCount 1000 --user admin --password admin --brokerUrl
'failover:(tcp://localhost:61616,tcp://localhost:61617)' --destination
queue://test.readonly.queue
```

9. Now lets try consuming from the test.readonly.queue using the read-only-user credentials:

```
java -jar activemq-all-5.11.0.redhat-630187.jar consumer --sleep 100 --
messageCount 1000 --user read-only-user --password Abcd1234 --brokerUrl
'failover:(tcp://localhost:61616,tcp://localhost:61617)' --destination
queue://test.readonly.queue
```

If all goes well, the client should connect and start consuming messages from our queue.

10. For the sake of testing, try the same to write messages to the queue. Again, this should fail but with a different permission error:

```
java -jar activemq-all-5.11.0.redhat-630187.jar producer --sleep 100 --
messageCount 1000 --user read-o
```