

**Demonstration Guide**

**Security**

Grupo T08

**Setup:**

1. **Obtain code from git repository (download or clone):**

<https://github.com/tecnico-distsys/T08-SD18Proj.git>

1. **Open terminal on project root** T08-SD18Proj.  **To install project run:**
   * cd uddi-naming & mvn install -DskipTests
   * cd ../kerby & mvn install -DskipTests
   * cd .. /ws-handlers & mvn install -DskipTests
   * cd .. & mvn install -DskipTests

**Note:** If on windows, running demoTest.bat will preform steps 1 to 3.

**F1 - Success**

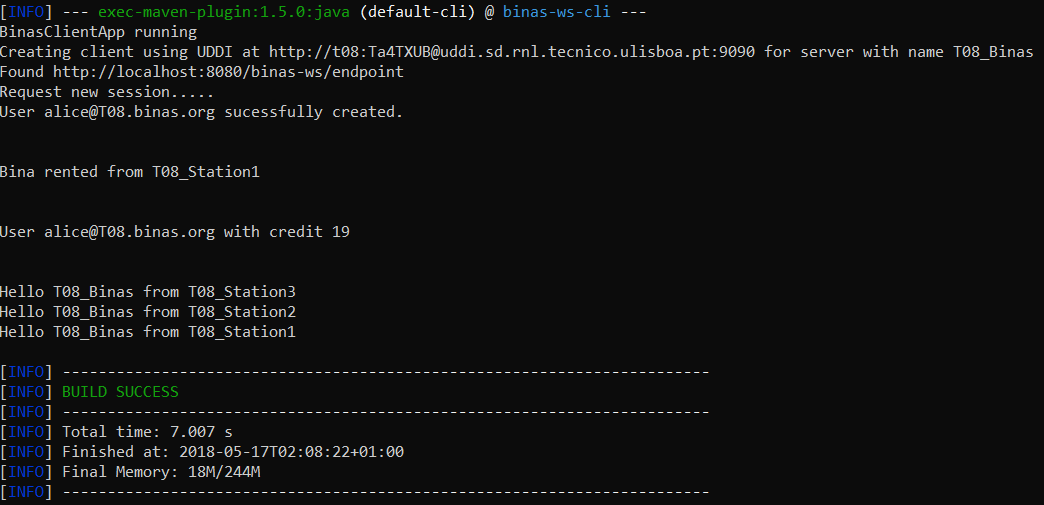
1. **Open 3 terminals on folder** T08-SD18Proj /station-ws **and run the following command to execute each *station server*:**
   * *T08\_Station1*: mvn exec:java
   * *T08\_Station2*: mvn exec:java -Dws.i=2
   * *T08\_Station3*: mvn exec:java -Dws.i=3
2. **Open a terminal on folder** T08-SD18Proj/binas-ws **and run the following command to execute the *binas server*:**

* *T08\_Binas*: mvn exec:java

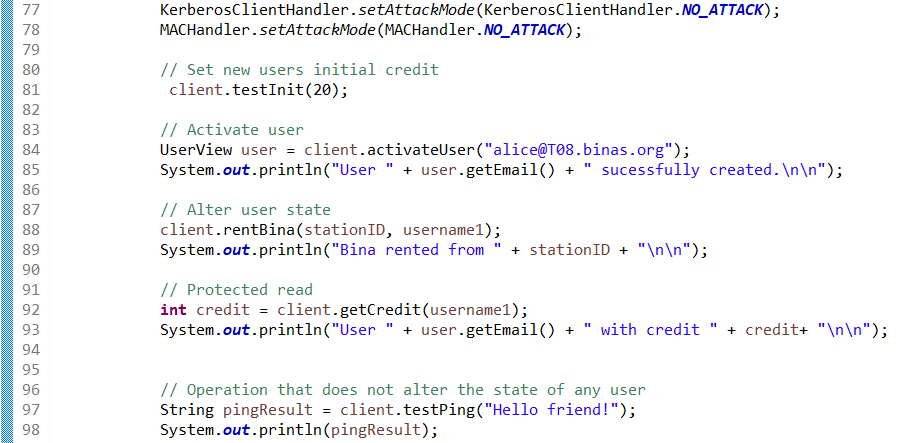
1. **Open a terminal on folder** T08-SD18Proj/binas-ws-cli **and run the following command to execute the *binas client*:**

* *BinasClient:* mvn exec:java -Dws.fase=F1 -Dws.showLog=no

1. **Verify that the *binas client* returned the following output:**



* The contents of the messages exchanged between the client and the server can be seen in the output of the Binas server

****

**The code above is executed by *binas client* main class:**

1. Activates the user with email “alice@T08.binas.org”
2. The user “alice@T08.binas.org” rents a bina, changing the user state
3. The user “alice@T08.binas.org” requests its credit info, reading the user state
4. Invokes an operation that does not read or modify any user’s state

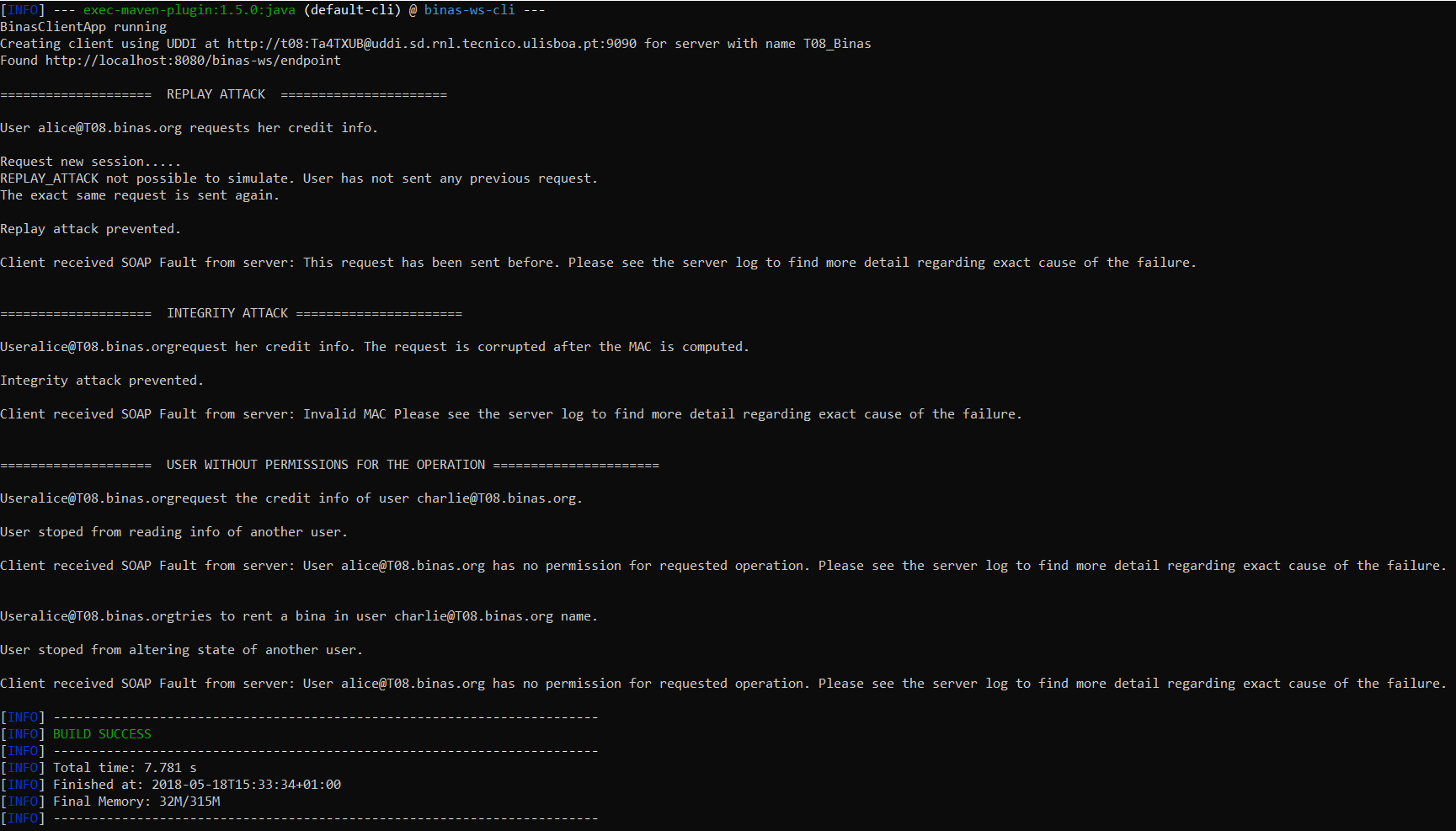
.

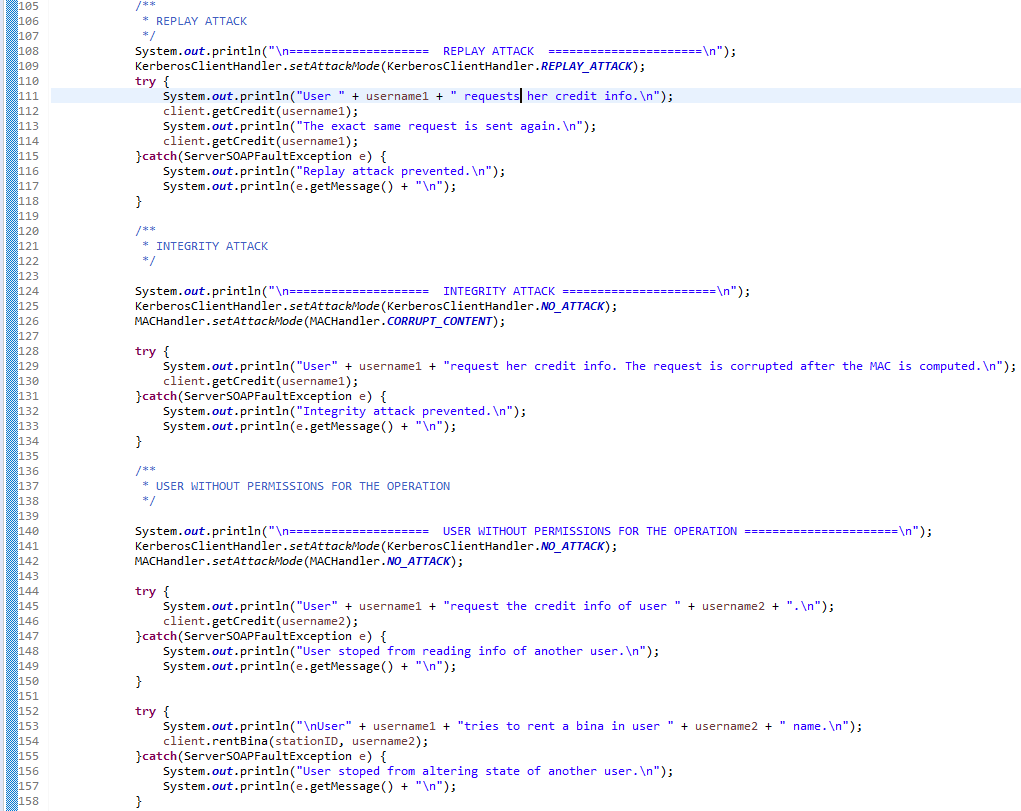
**F2 – Resistance to Attack**

1. **On the terminal at the folder** T08-SD18Proj/binas-ws-cli **and run the following command to execute the *binas client*:**

* *BinasClient:* mvn exec:java -Dws.fase=F2 -Dws.showLog=no

1. **Verify that the *binas client* returned the following output:**



****

**The code above is executed by binas client main class:**

* **Replay Attack**

1. The user “alice@T08.binas.org” requests her credit info. Then the exact same request message is sent.
2. The KerberosClientHandler of uses the same request time from the previous request to create a new authenticator.
3. The BinasServerHandler, when checking whether the request time is fresh, throws a RuntimeException.

* **Integrity Attack**

1. After the MACHandler, on the client’s side, generates the MAC, the message body is modified.
2. The MACHandler, on the server’s side, after computing the MAC of the incoming request, verifies it does not match the MAC in the header and throws a RuntimeException.

* **Unauthorized Acess**

1. The user “alice@T08.binas.org” requests its credit info of user “charlie@T08.binas.org”.
2. The BinasAuthorizationHandler, when checking whether Alice has permission to read Charlie’s state, throws a RuntimeException.
3. Then Alice tries to request a bina for Charlie.
4. The BinasAuthorizationHandler, when checking wether Alice has permission to modify Charlie’s state, throws a RuntimeException.