Demonstration Guide

Security

Grupo T08

Setup:

1. **Obtain code from git repository (download or clone):**
   https://github.com/tecnico-distsys/T08-SD18Proj.git
2. **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.

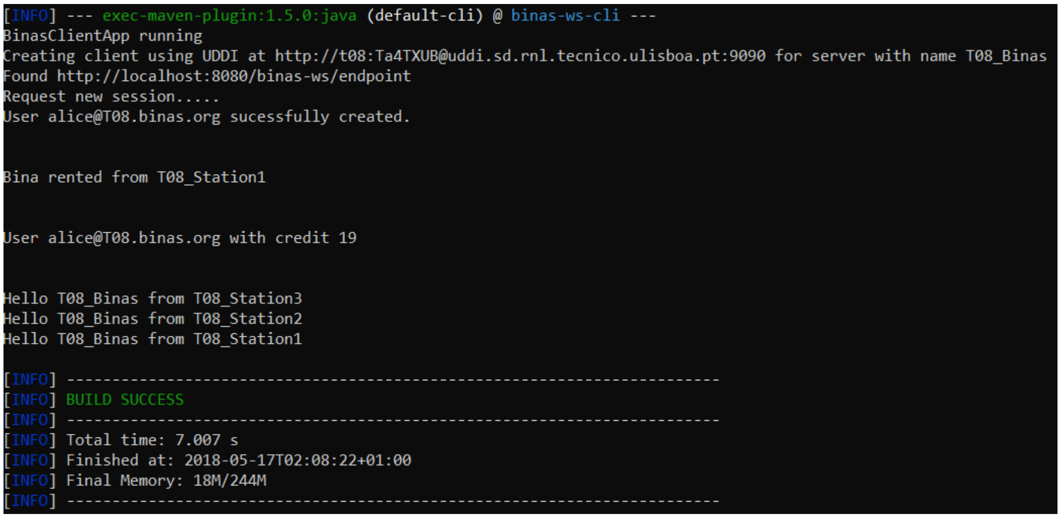
F1 - Success

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

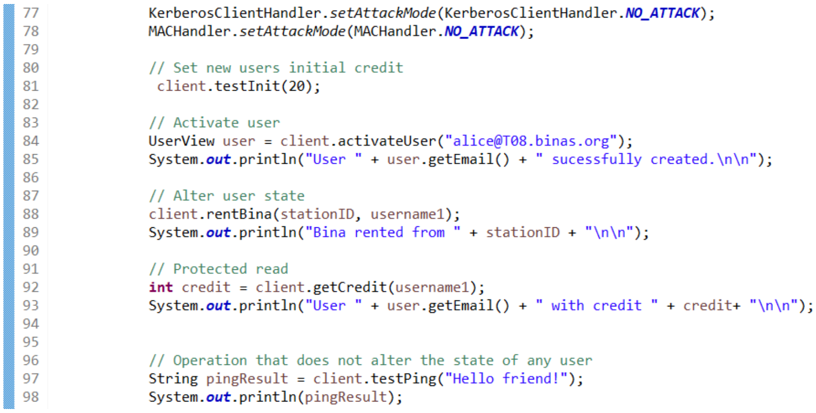
3. **Open a terminal on folder** T08-SD18Proj/binas-ws-cli **and run the following command to execute the *__binas client__*:**
   - *BinasClient:* mvn exec:java -Dws.fase=F1 -Dws.showLog=no

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



```
[INFO] --- exec-maven-plugin:1.5.0:java (default-cli) @ binas-ws-cli ---
BinasClientApp running
Creating client using UDDI at http://t08:Ta4TXUB@uddi.sd.rnl.tecnico.ulisboa.pt:9090 for server with name T08_Binas
Found http://localhost:8080/binas-ws/endpoint
Request new session.....
User alice@T08.binas.org sucessfully created.


Bina rented from T08_Station1


User alice@T08.binas.org with credit 19


Hello T08_Binas from T08_Station3
Hello T08_Binas from T08_Station2
Hello T08_Binas from T08_Station1

[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 7.007 s
[INFO] Finished at: 2018-05-17T02:08:22+01:00
[INFO] Final Memory: 18M/244M
[INFO] ------------------------------------------------------------------------
```

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



```
KerberosClientHandler.setAttackMode(KerberosClientHandler.NO_ATTACK);
MACHandler.setAttackMode(MACHandler.NO_ATTACK);

// Set new users initial credit
 client.testInit(20);

// Activate user
UserView user = client.activateUser("alice@T08.binas.org");
System.out.println("User " + user.getEmail() + " sucessfully created.\n\n");

// Alter user state
client.rentBina(stationID, username1);
System.out.println("Bina rented from " + stationID + "\n\n");

// Protected read
int credit = client.getCredit(username1);
System.out.println("User " + user.getEmail() + " with credit " + credit+ "\n\n");


// Operation that does not alter the state of any user
String pingResult = client.testPing("Hello friend!");
System.out.println(pingResult);
```

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

i. Activates the user with email “alice@T08.binas.org”

ii. The user “alice@T08.binas.org” rents a bina, changing the user state

iii. The user “alice@T08.binas.org” requests its credit info, reading the user state

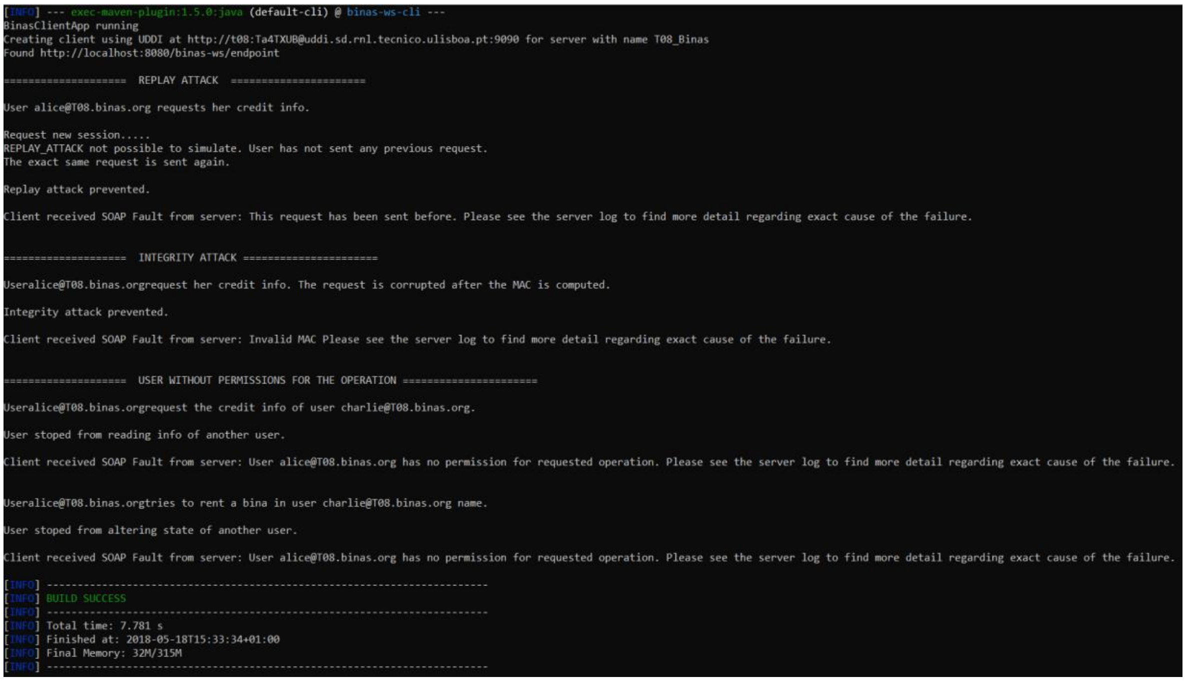
iv. Invokes an operation that does not read or modify any user’s state

.

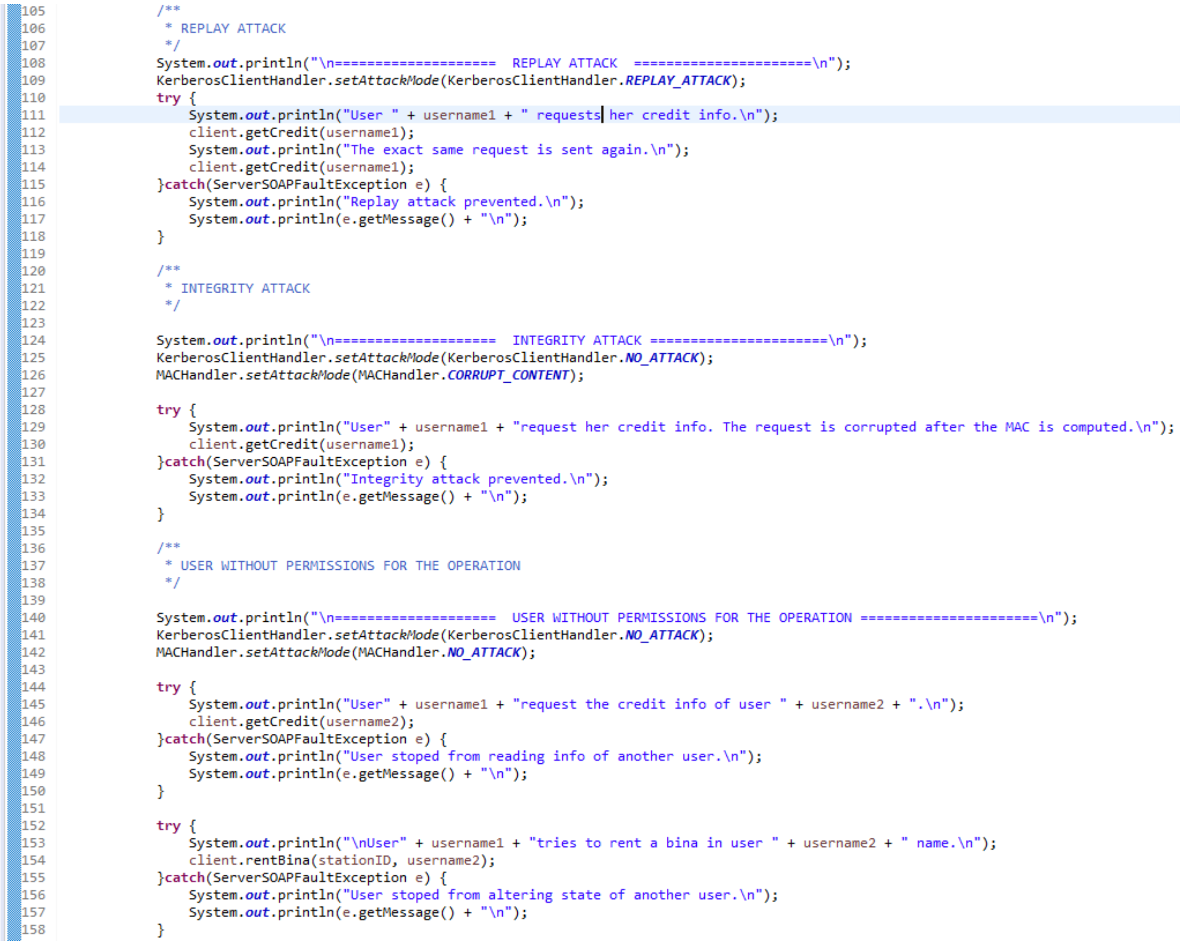
F2 – Resistance to Attack

5. **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

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



```
[INFO] --- exec-maven-plugin:1.5.0:java (default-cli) @ binas-ws-cli ---
BinasClientApp running
Creating client using UDDI at http://t08:Ta4TXUB@uddi.sd.rnl.tecnico.ulisboa.pt:9090 for server with name T08_Binas
Found http://localhost:8080/binas-ws/endpoint

====================  REPLAY ATTACK  ======================

User alice@T08.binas.org requests her credit info.

Request new session.....
REPLAY_ATTACK not possible to simulate. User has not sent any previous request.
The exact same request is sent again.

Replay attack prevented.

Client received SOAP Fault from server: This request has been sent before. Please see the server log to find more detail regarding exact cause of the failure.


====================  INTEGRITY ATTACK ======================

Useralice@T08.binas.orgrequest her credit info. The request is corrupted after the MAC is computed.

Integrity attack prevented.

Client received SOAP Fault from server: Invalid MAC Please see the server log to find more detail regarding exact cause of the failure.


====================  USER WITHOUT PERMISSIONS FOR THE OPERATION ======================

Useralice@T08.binas.orgrequest the credit info of user charlie@T08.binas.org.

User stoped from reading info of another user.

Client received SOAP Fault from server: User alice@T08.binas.org has no permission for requested operation. Please see the server log to find more detail regarding exact cause of the failure.


Useralice@T08.binas.orgtries to rent a bina in user charlie@T08.binas.org name.

User stoped from altering state of another user.

Client received SOAP Fault from server: User alice@T08.binas.org has no permission for requested operation. Please see the server log to find more detail regarding exact cause of the failure.

[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 7.781 s
[INFO] Finished at: 2018-05-18T15:33:34+01:00
[INFO] Final Memory: 32M/315M
[INFO] ------------------------------------------------------------------------
```



```
/**
 * REPLAY ATTACK
 */
System.out.println("\n====================  REPLAY ATTACK  ======================\n");
KerberosClientHandler.setAttackMode(KerberosClientHandler.REPLAY_ATTACK);
try {
    System.out.println("User " + username1 + " requests her credit info.\n");
    client.getCredit(username1);
    System.out.println("The exact same request is sent again.\n");
    client.getCredit(username1);
}catch(ServerSOAPFaultException e) {
    System.out.println("Replay attack prevented.\n");
    System.out.println(e.getMessage() + "\n");
}

/**
 * INTEGRITY ATTACK
 */

System.out.println("\n====================  INTEGRITY ATTACK ======================\n");
KerberosClientHandler.setAttackMode(KerberosClientHandler.NO_ATTACK);
MACHandler.setAttackMode(MACHandler.CORRUPT_CONTENT);

try {
    System.out.println("User" + username1 + "request her credit info. The request is corrupted after the MAC is computed.\n");
    client.getCredit(username1);
}catch(ServerSOAPFaultException e) {
    System.out.println("Integrity attack prevented.\n");
    System.out.println(e.getMessage() + "\n");
}

/**
 * USER WITHOUT PERMISSIONS FOR THE OPERATION
 */

System.out.println("\n====================  USER WITHOUT PERMISSIONS FOR THE OPERATION ======================\n");
KerberosClientHandler.setAttackMode(KerberosClientHandler.NO_ATTACK);
MACHandler.setAttackMode(MACHandler.NO_ATTACK);

try {
    System.out.println("User" + username1 + "request the credit info of user " + username2 + ".\n");
    client.getCredit(username2);
}catch(ServerSOAPFaultException e) {
    System.out.println("User stoped from reading info of another user.\n");
    System.out.println(e.getMessage() + "\n");
}

try {
    System.out.println("\nUser" + username1 + "tries to rent a bina in user " + username2 + " name.\n");
    client.rentBina(stationID, username2);
}catch(ServerSOAPFaultException e) {
    System.out.println("User stoped from altering state of another user.\n");
    System.out.println(e.getMessage() + "\n");
}
```

The code above is executed by binas client main class:

- **Replay Attack**
  - **i.** The user "alice@T08.binas.org" requests her credit info. Then the exact same request message is sent.
  - **ii.** The KerberosClientHandler of uses the same request time from the previous request to create a new authenticator.
  - **iii.** The BinasServerHandler, when checking whether the request time is fresh, throws a RuntimeException.

- **Integrity Attack**
  - **i.** After the MACHandler, on the client's side, generates the MAC, the message body is modified.
  - **ii.** 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**
  - **i.** The user "alice@T08.binas.org" requests its credit info of user "charlie@T08.binas.org".
  - **ii.** The BinasAuthorizationHandler, when checking whether Alice has permission to read Charlie's state, throws a RuntimeException.
  - **iii.** Then Alice tries to request a bina for Charlie.
  - **iv.** The BinasAuthorizationHandler, when checking wether Alice has permission to modify Charlie's state, throws a RuntimeException.