4.3 Simulation 3: The DCS student in the Middle

Topics: Man-in-the-middle attacks: intercepting and modifying communication

DCS spies have made a significant breakthrough that should simplify our acquisition of intelligence on hostile departments. The spies have infiltrated a data centre belonging to the the Warwick Megalomaniacs Groups and have installed a device which can intercept communications between a server in the data centre and clients wishing to connect to it. You can obtain access to this device through three separate programs which you can obtain by running the following command:

```
$ git clone https://github.com/dcs-cs263/lab3
```

Running the programs (without Docker)

The three programs can be invoked as follows (and should be invoked in the following order):

1. `./start-server.sh` allows clients to connect to the server on a random port via TCP. The port on which the server listens for connections is printed to the standard output and can be overriden by setting an environment variable named `SERVER_PORT` for the process and setting it to the desired port.
2. Open a terminal in the `listener` folder and run `gradle build` followed by `gradle run` to start a program which can intercept data between clients and the server. It does this by connecting to the server and listening for data from clients on a random port. All intercepted messages from the clients are passed to the server and vice-versa. The intercepted messages will be printed to the standard output by the listener.
3. `./start-client.sh` starts the client which thinks it is connecting to the server, but it is actually connecting to the listener which then intercepts messages between the client and server.

Running the programs (with Docker)

If you have Docker and `docker-compose` installed on your personal machine, you can start the programs more easily by running `docker-compose up` in the root directory of the lab. You can use the `dockerise.sh` script in the listener folder to build the image for the listener (and you will have to repeat this every time you change its code). `docker-compose` should automatically create the images for the server and client the first time it starts up, but you can also force this by running `docker-compose up --build`.

The simulation and task

Therefore, by running the three programs in the given order, the listener will intercept messages between the clients and server. By default, all messages will be printed to the terminal and marked with their respective direction ("Client->Server" or "Server->Client"). You have access to the source code for the listener and can modify what it does with the intercepted messages before passing them on (or choosing not to pass them on to their intended destination).

Ex8 As expected, traffic between the server and its clients is encrypted. Your first task is to break this encryption so that you can read the contents of the messages that sent across the network. You will need to determine:

- Whether the same encryption key(s) are used by the server/client for every connection and, if not, how they are negotiated.
- What encryption method is used.
- How exactly the key is used/modified by the clients/server to be a suitable encryption key for the encryption method that is used.
- How encrypted data is transmitted.

Remember to ask for help during the simulation if you feel stuck. Some additional advice:

- The simulation is set up with the expectation that you will extend the Java code that you are given in the `listener` folder. In particular, you should note that Java is big-endian while other languages may not be.
- Programs written in Java can easily be decompiled and you can decompile the server/client programs given to you for this exercise, but that is not the intended way of solving this task and it trivialises the exercise. You should try to complete it without doing this and seek help if you get stuck.

Ex9 Once you are able to read messages and manipulate them, you have a new target: steal the secret files from the server in order to acquire *vital* intelligence.
