Distributed Systems and Security Lab

Graham White

Week 2

This exercise will get you working with the Java networking classes. A useful reference is the Networking Trail in the Oracle Java tutorials: `http://docs.oracle.com/javase/tutorial/networking/TOC.html` This exercise sheet tells you to do what these tutorials describe, up to and including the section All About Sockets.

1. Create two Java programs, a `Client` program (using the `java.net.Socket` class) and a `Server` program, (using the `java.net.ServerSocket` class), so that, if you do the following

   (a) Run `Server`

   (b) Run `Client` with a command line argument of `xyz`

   then the `Server` program will print out `xyz` and then terminate. This will be with both programs running on the same machine.

2. Now modify your client program so that it will take an additional command line argument which is a hostname, and so that you can start the server on Machine A, run the client on Machine B, and the server will print out the string on Machine B which you have given to the client on Machine A.

3. Modify your server so that it will keep on accepting messages and printing them out. You will have to shut it down from the machine it's running from, by terminating the process.

4. Now modify the server so that it will count votes: you should send it strings which are colour names, and it should keep count of how many people have voted for each colour. If you send it a secret string (which is chosen not to be a colour name) then it should terminate and print out the cumulative total of the votes for each colour.

You should, in the course of these, read the tutorial, and particularly the part about multithreaded server processes, and try the examples. This applies especially if you have difficulty doing what I ask above.

Hints The `Socket` and `SocketServer` classes have methods and constructors which take `InetAddress` arguments. `InetAddress` does not have constructors, but it does have static factory methods, `INetAddress.getByName(Strings)` and `InetAddress.getByAddress(byte[] bytes)`. Note that `InetAddress.getByName(null)` returns the address of localhost.

If you want to be able to communicate from machine to machine in the ITL using the `java.net` classes, you must use ports in the range 20000–20010 (inclusive). The hostname of an ITL machine will be of the form `itlXXX`, where `XXX` is a number; it will be written on a label on the front of the machine in question (or you can find it out by typing `hostname` at a command prompt).