COMPUTER NETWORKS

LAB LIST 8

1 General remarks

Before you begin run `netmode lab` command. During today's workshops `eth1` card are connected to the hub, and `eth0` to the switch. In exercise no. 1 will use `eth1` cards, and for all subsequent exercises – `eth0` cards.

2 Exercises

Exercise 1. Assign `10.0.0.i` IP address to `eth1` card of computer number *i*. By pining broadcast address `10.0.0.0/8`, verify that you are connected to other computers in the lab room.

Organize yourselves into groups of three people (we will use the terms *Person A*, *Person B* and *Person C*). *Person A* should change the password of user `student` with following command:

```
$> passwd
```

Choose a password that is non-trivial, so it cannot be easily guessed. Pass the password to *person B*, making sure it's not revealed to *person C*. Now, *person B* should log into `student` account on the computer of *person A*.

```
$> telnet  ip-address-of-person-A-computer
```

In the meantime, *person C* should try to eavesdrop traffic between *person A* and *person B* with *wireshark*, and discover the new password. For this purpose, "Follow TCP Stream" option in *wireshark*'s context menu may prove useful, as it shows all the data transmitted on selected connection.

Note: The default password of user `student` (i.e. cisco) can be restored with command:

```
#> restore_passwd
```

Carry out the experiment again using following command instead of `telnet`:

```
$> ssh  ip-address-of-person-A-computer
```

Is it still possible to intercept your password?

Deconfigure `eth1` card and activate `eth0` card and configure it using DHCP protocol:

```
#> ifconfig eth1 down
#> ifconfig eth0 up
#> dhclient eth0
```

Exercise 2. In this exercise, we'll configure `ssh` so that it can connect without typing in a password.

Use following command to generate private and public keys:

```
$> ssh-keygen
```

Save them in the default locations (`.ssh/id_rsa.pub` and `.ssh/id_rsa`). The tool will ask you for a private key protecting password, please leave it empty. Display generated files' contents and file system permission.

Now simply add the public key to `.ssh/authorized_keys` file on the `SSH` server, which we'll connect to. This server is your neighbor's computer, let *A* denote his *IP address*. Firstly, copy the key on the computer *A* with command:

```
$> scp .ssh/id_rsa.pub A:destination_file
```

Then, using `ssh` please log in on the computer providing `cisco` as a password:

```
$> ssh A
```

Append the public key, that just has been copied onto computer *A*, to `.ssh/authorized_keys` file, and then log off:

```
$> cat destination_file >> .ssh/authorized_keys
$> rm destination_file
$> exit
```

Make sure the action became effective, i.e. you can log in to computer *A* without a password. The following command displays subsequent steps while the connection is being established:

```
$> ssh -v A
```

Exercise 3. Create an `SSH` tunnel connecting port 2525 of the local computer to port 25 of `eagle-server.example.com` server:

```
$> ssh -f -N -L 2525:localhost:25 ccnai@eagle-server.example.com
```

where *i* is the number of your computer. Check what kind of service is responding on the other side of the tunnel:

```
$> telnet localhost 2525
```

Also check what is sent over the tunnel by using `wireshark`. Configure *Evolution* (or other e-mail application) by setting `SMTP` server address for outgoing mail to `localhost` and port 2525, and `POP3` server for incoming mail to `eagle-server.example.com` and port 110. Check the settings by sending a test e-mail.

Marcin Bieńkowski
Translation: *Krystian Bacławski*