Exercise 1, basic networking tools and TLS

T-110.6220 course staff

November 10, 2008

Introduction

The aim of this exercise is to bring all the students up to speed on the basic Unix tools on networking. The exercise is very similar to excercises found on courses T-110.2100 and T-110.5200.

You should complete the excercises in pairs and fill an answer sheet together. *Remember to put both your student numbers on the sheet*, we will use the sheets to control presence at the excercises.

1 Networking basic tools

Use the *ifconfig(8)* command to find all the active interfaces on your workstation? How many separate physical interfaces are there? Which interface is not a real physical interface?

Use the command command *netstat(8)* to find out the default gateway for your workstation and *arp(8)* to retrieve the associated Ethernet-address for your default gateway. What is it?

Look at *resolv.conf(5)*. What is the IP of your default DNS server?

Use *dig(1)* to find out the authoritative nameservers for the cse.tkk.fi domain. Which former laboratory do they belong to? (If you don't have dif, you can run this part in your virtual machine later on).

2 *nc(1)* and communications protocols by hand

Use netcat (*nc(1)*) to capture the version number of the ssh daemon running on your workstation. What is this version number and the name of the workstation you are using?

Craft a valid HTTP request that requests the front page of `www.cse.tkk.fi`.

A simple example from another course where we tried to read There is a very good reason for the white space at the end of the *cat* output. What is it?

```
$ cat foo.txt
GET / HTTP/1.0
Host: www.kasvi.org
Connection: close


$ cat foo.txt |  nc www.kasvi.org 80
```

Use netcat to create a server that listens on port 8080 and responds to any connections with some text of your choice.

The easiest way to do this is to use an unending while loop of the form

```
while true; do commands ; done
```

from the command line. This has the unfortunate side effect that you must kill your shell to kill the loop and end this kind of a server.

Try to connect to the port 8080 with a web browser? Why doesn't your

Which parameter is necessary in order to close the connection in a reasonable time if the client requests a keep-alive connection??

Listen on another socket with *nc(1)* and connect to it with ssh (the -p handle allows you to connect to a non-standard port). Reply to the connection attempt with the ssh version string that you obtained previously. What kind of an answer do you receive?

Try to listen to port number 1 and notice that you can't. What is the first port number you can bind to as a regular user?

3 Wireshark and TLS handshake

Use the program *wireshark(1)* to capture a TLS handshake. You could easily generate a TLS handshake by going to almost any web page that uses the HTTPS protocol. For the sake of excercise, please go to the Oodi web page `https://oodi.tkk.fi/w/`.

Unfortunately listening to all the network traffic requires root priviledges and getting root priviledges on regular school machines is not possible. Therefore you need to start a virtual machine. First, run

```
$ ./restore-machines.sh
$ vmplayer &
```

and start one of the virtual machines in the recently used list (preferably number 1). The username and password is root/root. Install Firefox and start an X server by giving the following commands

```
$ apt-get update
$ apt-get install firefox
$ startx
```

Then you can start wireshark and firefox from an X terminal. Capture the traffic that takes place during the first page load from Oodi.

Find a Server Hello message and look at the certificate that is being transpordet. Can you discern a company name from the hex dump of the message? Then view the certificate in your browser. You can get the certificate in Firefox by clicking on the padlock icon in the lower right corner of the screen. What was the role of the company whose name was in the certificate message? Which public key algorithm does Oodi use?

Use the wonderful[1] tool at `http://www.websequencediagrams.com` to generate a sequence diagram about the TLS handshake based on the data you captured (no cheating and looking at wikipedia!). It is recommended to do this on the workstation, not on the virtual machine.

Send the diagram and the answer sheet to the courde T-110.6220(at)tml.hut.fi and have a nice day!

[1] course staff recommends it for making all the MSC diagrams you must submit to them