Week 10 – TCP/UDP

A pic here…

**Class:**

**Student numbers:**

**Student names: Sebastian**

Date: Aug 2019  
Version 1.0

# TCP/UDP

Task 1: TCP in Netcat

To do this assignment we will use the Netcat tool which is provided in the Netkit. Netcat makes it possible to create and use TCP/UDP connections. If you want more info about Netcat you can consult Internet. To make this assignment we will reuse the net\_routing lab from the previous assignments. Let’s start a chat session by connecting 2 netcat instances via a TCP connection.

To listen to the TCP connections, go to one of your simulated nodes (e.g. PC1A) and issue the following command:

**nc –l –p <port\_nr>**

This will make netcat listen to port number that you have specified in port\_nr and accept connections.

Note: Any port number would be ok, as long as it is not used by another application.

To establish a TCP connection you can issue the following command from another simulated node (e.g. PC1C)

* **nc <IP address of the “listening” node> <port\_nr of the “listening node”>**

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

This will make a TCP connection with the listening netcat instance.Now you can chat from one netcat instance to the another. Try it out!

Your task:

* Netcat can also be used to copy the contents of a file from one place (file, folder, computer) to another. Find out how and try it out.

Provide screenshots of the sending and receiving command.

Let’s now build a basic one-page webserver using netcat.

As a first step create a textfile ‘response.txt’ with following content:

**HTTP/1.1 200 OK**

**Content-Type: text/html;**

**Content-Length: 12**

**Connection: close**

**Hello World!**

This is a proper HTTP response.

You are going to simulate HTTP server and HTTP client (browser) using netcat. You are going to use Gateway node for running HTTP client (browser), see section below about how to install links browser on Gateway before you start the exercise. You are going to use any node of your network (e.g. PC2C) to simulate HTTP server.

Your task:

Construct an appropriate netcat command to listen to a port on your HTTP server (e.g. PC2C) and send the contents of the file response.txt to the HTTP client (Gateway) when a connection is made to this port. That is roughly what a webserver does too.  
You can test it by entering the following URL in links browser on Gateway.

**http://<IP\_ADDRESS\_OF\_WEBSERVER>:<port\_nr>**If everything works well the links should show the “Hello World” webpage.

Links installation:

Before starting this task, you have to install links text-mode web browser. Follow these steps:

* Go to your lab’s Gateway directory. Create etc subdirectory and create there a resolv.conf text file with the following contents:

nameserver 8.8.8.8  
search localdomain

* Start your lab. Now you should be able to connect to the Internet, so do the following installation on the Gateway node:

apt-get install links

Now the links web browser should be available on your Gateway node. Watch out, once you stop your Gateway node, you have to do ‘apt-get install links’ command again to install links.

Note1: You can start links browser by issuing this command:

links

To be able to enter the URL in links browser press “G”.

Note2 : You can put the response.txt file in your net\_routing directory **before** start of the lab (lstart command). **After** starting the lab, you can find this file in the /hostlab directory of your node.

Provide a screenshot of the netcat command you used and of the links browser output.

Task 2: Find 2 TCP uses

Think about two different scenarios for TCP use that you can simulate (you can do this on your own PC, so you don’t need Ubuntu for this). Start a Wireshark trace for both scenarios.

Describe the chosen scenarios and a proof of TCP use in them by attaching a Wireshark trace showing TCP packets.

Choose one of the 2 scenarios traces and browse it in the Wireshark. Select ‘Statistics > Flow Graph’ and then choose flowtype ‘TCP flow’ to draw a Sequence Diagram of the TCP message interaction that you see in Wireshark.

Provide a screenshot of this Flow Graph.

Explain what is happening duringvarious stages (begin, middle, end) of the communication. Explain SYN, SYNACK and ACK. Explain the Len, Seq and Ack numbers.

Task 3: Find 2 UDP uses

Think about two different scenarios for UDP use that you can simulate (you can do this on your own PC, so you don’t need Ubuntu for this). Start a Wireshark trace for both scenarios.

Describe the chosen scenarios and a proof of UDP use in them by attaching a Wireshark trace showing UDP packets.

Task 4 (Optional): TCP SYN Flooding

Read an explanation of TCP SYN Flooding at <http://en.wikipedia.org/wiki/SYN_flood> or from some other source.

In this task you’re going to simulate this kind of DDOS attack that uses vulnerability of TCP protocol.

For this experiment you can reuse net\_routing lab. You can use for example the PC1B node as the victim and RouterAC node as an attacker.

To be able to wait for the TCP connections, use netcat command to wait for the TCP connections on a specific port at the victim node.

To simulate TCP SYN flood traffic from the attacker node, you can use the “hping3” tool which is part of your netkit nodes.

Before you start the attacker command, don’t forget to sniff the traffic with tcpdump command and write the output to a pcap file:

tcpdump –w <filename>.pcap –s 0

Tip: If you want to run tcpdump or any other command in the background you can do it by specifying “&” at the end of the command. In this way you can use your Linux prompt again. To see all your background processes use “jobs” command and to put a job in foreground again use : “fg <job\_number>”.

A command to be issued at the attacker node can look like:

hping3 --rand-source <IP\_ADDRESS\_OF\_VICTIM> --flood –S –L 0 –p <PORT\_NR\_OF\_VICTIM>.

Wait about 10 seconds, stop hping3 and tracing.

Now you should be able to analyze the trace. You should be able to see spoofed source IP address.

Analyze your trace. Find out how many SYNs, SYN+ACKs and ACKs you can see. Explain what do these numbers tell you about SYN attack.

Consult internet to find out how another transport protocol – SCTP - solves TCP SYN flooding problem. Give a short explanation of how it is implemented in SCTP.