Networking Lab Day 1

Kevin Neumann, Alain C. Mendy

April 29, 2014, Goettingen

Contents

1 Prelab

In order to have version control and making changes of files parallely possible, a git repository (https://www.github.com/Phiology/networkingLab) was initialized. In the repo, the results of each week's lab will be deposited, using the directory structure labX/exercise where the files that were asked to be generated will be put into. Also, during the lab a bash script will be generated in which all commands exercised in a terminal are collected, so that repetition of the commands is easier in case of errors and so that one can track the commands issued during the lab course.

The Prelab questions were answered before the course, and the results can be found in lab1/prelab/answersLab1.

2 Part 1: Becoming familiar with the equipment

The network was set up according to the information given in the lab wiki. The connectivity was tested, and all PCs were connected to the network, except for PC3. This was due to an "error" in the file /etc/network/interfaces, in which all information necessary for the interfaces were outcommented. The interfaces file was fixed and the networking deamon restarted by entering '/etc/init.d/networking restart'. After that, telnet communication was possible from and to all 4 PCs.

3 Part 2: Using the Linux Operating System

It was asked to use some basic linux commands. This exercise was skipped, because we are already familiar using linux based operating systems.

4 Part 3

The content of /etc/ was saved on three different ways (using redirection, view and save at the same time, and via copy paste). The files etcfile_1-3 can be found in lab1/exercise3/.

5 Part 4

5.1 Saving data to a USB stick

The etcfile_1 was saved to a USB stick. Also, the output of df was redirected into a file df_output. The file can be found in lab1/exercise4/.

5.2 Convention for saving data on a USB stick

FTP was used to acquire each etcfile_1-3 from PC1,PC3 and PC4. These files can be found in lab1/exercise4/PC[1,3,4]/.

6 Part 5

6.1 Using the more command

The output of the command 'more /etc/hosts' was saved and can be found in lab1/exercise5/etc_hosts.

6.2 Exercise 5

Which files must be edited to change the name of a Linux PC (e.g. from PC1 to machine1)?

The file /etc/hostname has to be edited in order to permanently change the hostname of any Linux PC.

Which files include information that determines whether a Linux PC performs IP forwarding?

Whether or not a linux PC performs IP forwarding can be seen in /proc/sys/net/ipv4/ip_forward. When the file contains a 0, no forwarding takes place, 1 means forwarding is enabled.

It is also possible that permanent forwarding is enabled in /etc/sysctl.conf. If that's the case, it contains a line net.ipv4.ip_forward = 1.

Attach the standard Ubuntu network interface configuration file.

The /etc/network/interfaces file can be found in lab1/exercise5/.

7 Part 6

7.1 Issuing Ping commands

From PC2, the commands 'ping -c 5 10.0.1.11' (ping PC1) and ping -c 5 127.0.0.1 (ping lo interface) were issued and the results can be found in lab1/exercise6/ in the files ping_1 and ping_lo.

7.2 Exercise 6

Explain the difference between pinging the local Ethernet interface and the loopback interface. Specifically, on PC1, what is the difference between typing ping 10.0.1.11 and ping 127.0.0.1. (This is a conceptual question on the role of the loopback interface. The response of the ping command does not provide you with the answer to this question.)

The loopback interface (lo) is a special, virtual network interface that a computer uses to communicate with itself. It's purpose is that applications

on a PC can always connect to servers or communicate with other tools on the same PC, even when the ethernet port is unplugged or Wi-Fi is shut down. The lo interface is mainly used for troubleshooting and diagnostics.

From PC1 sight: The difference between 'ping 10.0.1.11' and 'ping 127.0.0.1' is that the second one is always going to work, no matter what. However, pinging yourself using your eth0 adress is not going to work when there is no network connection. Because the eth0 adress ping is issued via the network, it is slower than pinging the loopback. But when trying to find out whether or not the network is working, it should be refrained from issuing the loopback command.

8 Part 7

8.1 Simple tcpdump exercise

On PC2, 'tcpdump -n host 10.0.1.11' was issued as root and a ping was executed to PC1. The output of tcpdump was saved and can be found in lab1/exercise7/tcpdump_PC1.

8.2 Exercise 7A

Explain the meaning of each field in the captured data.

10:17:47.131263 IP 10.0.1.12 ¿ 10.0.1.11: ICMP echo request, id 57629, seq 1, length 64
10:17:47.131390 IP 10.0.1.11 ¿ 10.0.1.12: ICMP echo reply, id 57629, seq 1, length 64
10:17:52.130635 arp who-has 10.0.1.12 tell 10.0.1.11
10:17:52.130654 arp reply 10.0.1.12 is-at 00:18:8b:1a:be:17

The first line describes the echo request carried over by the internet control message protocol (ICMP).
The second line sends back the reply from PC1.
The third line shows that 10.0.1.11 wants to send some data. It knows it must send this data to 10.0.1.12, but it does not know the MAC address to send it to, so it has sent an Address Resolution Protocol (ARP) request to find that out.
The fourth line gives the wanted information to PC1 so it can send the data.

8.3 Another tcpdump traffic capture

On PC2, packet capturing was started by issuing 'tcpdump -n' as root. A ping was issued to a nonexistant IP adress (111.111.111.111). Also a ping was issued to the broadcast adress 10.0.1.255. The outputs of ping commands are saved in files ping_nonexistant and ping_broadcast and the

tcpdump output was saved to the file tcpdump_nonexistant_and_broadcast. All files can be found in lab1/exercise7/.

8.4 Exercise 7B

Interpret the results. How many of the Linux PCs responded to the broadcast ping?

10:51:55.173197 arp who-has 10.0.1.254 tell 10.0.1.12

This is the output of tcpdump when pinging an unreachable host. Arp can't resolve the host, so it tries three times to ask the gateway/router where to continue searching. Since there is no further route, the search remains unsuccessful and the ping fails.

10:52:05.300743 IP 10.0.1.12 ¿ 10.0.1.255: ICMP echo request, id 36384, seq 1, length 64
10:52:05.300886 IP 10.0.1.11 ¿ 10.0.1.12: ICMP echo reply, id 36384, seq 1, length 64
10:52:05.300928 IP 10.0.1.14 ¿ 10.0.1.12: ICMP echo reply, id 36384, seq 1, length 64
10:52:05.300934 IP 10.0.1.13 ¿ 10.0.1.12: ICMP echo reply, id 36384, seq 1, length 64
10:52:06.300724 IP 10.0.1.12 ¿ 10.0.1.255: ICMP echo request, id 36384, seq 2, length 64
10:52:06.300825 IP 10.0.1.11 ¿ 10.0.1.12: ICMP echo reply, id 36384, seq 2, length 64
10:52:06.300834 IP 10.0.1.14 ¿ 10.0.1.12: ICMP echo reply, id 36384, seq 2, length 64
10:52:06.300846 IP 10.0.1.13 ¿ 10.0.1.12: ICMP echo reply, id 36384, seq 2, length 64
10:52:10.297260 arp who-has 10.0.1.12 tell 10.0.1.11
10:52:10.297290 arp reply 10.0.1.12 is-at 00:18:8b:1a:be:17
10:52:10.300667 arp who-has 10.0.1.12 tell 10.0.1.14
10:52:10.300679 arp reply 10.0.1.12 is-at 00:18:8b:1a:be:17
10:52:10.314716 arp who-has 10.0.1.12 tell 10.0.1.13
10:52:10.314727 arp reply 10.0.1.12 is-at 00:18:8b:1a:be:17

The output of the broadcast ping is similar to the one in Exercise 7A. The ping goes to 10.1.0.255 (broadcast address) and all three connected PCs answer the echo request. ARP again resolves the MAC-Address of PC2 for each other PC. However, the broadcast only reaches all three other computers after the command 'sysctl -w net.ipv4.icmp_echo_ignore_broadcasts=0' was issued as root on PC3, because PC3 was ignoring broadcasts before.

9 Part 8

Wireshark was started and traffic of different ping commands captured.