University of Bamberg
Professorship for Computer Science

Foundations of Internet Communication
KTR-GIK-M
Laboratories Summer Term 2020

Assignment 1

Fundamentals of Internet Packet Sniffing

Marcel Großmann, Duy Le, Jopaul John, Markus Wolff

Issued:	April 27, 2020
Deadline:	May 10, 23:55, 2020

Prelab

1. Work through the Network Protocol Analyzer Tutorial (available in the VC-course).
2. Make yourself familiar with the *brctl* command, which is used to configure a normal Linux PC as switch.
3. Refresh your knowledge about IP networks (e.g. IP addresses, sub-netting, etc.) and make yourself familiar with the Linux network configuration commands and files, like *ifconfig*, *ip*, etc.
4. Make yourself familiar with Kathará.

1 Taking Wireshark for a Test Run

The best way to learn about any new piece of software is to try it! Perform the following experiment:

1. Start the Wireshark software. To begin packet capture, select the *Capture* pull down menu and select *Options*. You can use all of the default values in this window, but uncheck "Hide capture info dialog" under Display Options. Select the network interface that is being used to send and receive packets (mostly likely the wired interface which counts frames). Now start the packet capture.

2. While Wireshark is running, visit the URL http://neverssl.com/ with your browser. After your browser has displayed the page stop the Wireshark packet capture. The HTTP message exchanges with the web server should appear in the listing of packets captured. There will probably be many other types of packets displayed as well.

3. Set the wireshark display filter to "http". Select the first HTTP message in the packet-listing window. This should be the HTTP GET message that was sent from your computer to *http://neverssl.com/*. When you select the HTTP GET message, the header information of different network layers will be displayed in the packet-header window. Expand the information about the HTTP protocol.

4. Congratulations, you made your first packet capture experience.

2 Wireshark Handling

The goal of this tutorial is primarily to introduce you to Wireshark. Further information about Wireshark and protocol analyzers in general can be found in the VC and the online documentation of the program.
Answer the following questions, based on your Wireshark experimentation and **provide screenshots** where it is annotated.

1. Provide a short documentation that explainand its actions.
2. How long was the period between the sending of the HTTP GET message until the receipt of the HTTP OK reply? Provide screenshots
3. What are the IP addresses of the servers www.fau.de and www.denic.de?

3 Wireshark Packet Filtering

The following experiments can be performed with assistance from the tutors. Wireshark provides a set of man pages directly accessible via Help ManualPages in the program. You can also find a complete documentation on the homepage of the program. Answer the following questions and **provide screenshots** where specified.

1. Use the tool `ping` to check the connection between your host and another in the network. Capture the traffic and explain **briefly** how the tool works. State a `display filter` expression you can use to display the specific traffic. Provide screenshots.
2. Display only the packets sent by your host if you visit the URL http://www.caida.org/tools/visualization/mapnet with your browser. Do this by filtering IP addresses to display only packets with the IP address of http://www.caida.org/tools/visualization/mapnet. Provide screenshots.
3. What is the MAC address of your host? Display only the packets sent by your host in the previous experiment. Do this by filtering MAC addresses.
4. Explain the difference between a MAC and an IP address.

5. Does your PC need MAC or layer 2 addresses? If yes explain why, if no describe the system that works without them.

6. Display only the packets that have been received by your host. Do this by filtering IP addresses. What is your IP address? Provide scrlinux start telnet servereenshots.

7. Write the syntax of a `tcpdump or capture filter expression` that captures packets containing IP datagrams with a source or destination IP address equal to your IP address.

8. Write the syntax of a `tcpdump or capture filter expression` that captures packets containing IP datagrams between two hosts with IP addresses 10.0.0.3 and 10.0.0.12 both on interface *eth0*. Write the command for a capture on host 10.0.0.3.

9. Write a `tcpdump or capture filter expression` that captures TCP packets using port number 22.

10. Write the syntax for a `display filter` that shows only IP datagrams with a destination IP address equal to 192.168.178.1 and frame size greater than 350 bytes.

4 Taking Kathará for a Test Run

Kathará, which originates from the Greek Καθαρὰ is an implementation of the notorious Netkit using Python and Docker. Ten times faster than Netkit and more than 100 times lighter, allows easy configuration and deploy of arbitrary virtual networks featuring SDN, NFV and traditional routing protocols such as BGP and OSPF.
Kathará comes with P4, OpenVSwitch, Quagga, Bind, FRRouting and more, but can also be extended with your own container images.
Thanks to Docker, the framework has the performances to run in production and our images can emulate most network equipments

Installation of Kathará

Linux

- Install Docker. We suggest installing Docker from this script.

- (suggested) Install xterm terminal emulator (usually `sudo apt install xterm`), that is used by default. You can also specify a different terminal emulator by using the `kathara settings` command.
- Add Kathará public key to your keyring:
  `sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 21805A48E6CBBA6B991ABE76646193862B759810`
- Add the Kathará repo to your repos on **Debian** based systems:
  `sudo add-apt-repository ppa:katharaframework/kathara`
- Update your apt cache by running `sudo apt update`
- Install Kathará running `sudo apt install kathara`

Windows

This guide applies only on Windows 10 Pro, if you have a Home version you can either install a virtual machine with Linux or buy a Pro license for your PC. Technically Kathará can be configured to work with Docker-Machine on Windows Home, but it has not been tested.

- Install Docker from here.
  - During the installation procedure **do not** flag "Use Windows Containers", Kathará works only with default (Linux) containers.
  - NB: On Windows 10 this will also download and enable Hyper-V. If you later want to use another hypervisor like VMware or Virtual Box, than you will have to disable Hyper-V and restart your PC. If you later need to use Docker or Kathará again, you'll have to re-enable Hyper-V and restart. This will require that Virtualization technology is enabled in your system BIOS. For more information and an example check out this link.
- Download the setup file to a directory of your choice from a Release.
- You may also need to share the drive that will contain the labs and the drive with your user folder (it can be done from Docker settings, from the tray icon), as shown here (note that you may have/need different drives).
- Run the setup wizard and follow the instructions.
- Remember to run Docker before using Kathará.

Please note that Docker runs on Windows inside a virtual machine running Linux and managed by Hyper-V. The virtual hard drive usually takes a variable amount of space depending on the number of Docker images that you compile and can go up to about 60 GBytes. However the space occupation can be drastically reduced by removing unused Docker images with the command `docker system prune` and by following this procedure. This can be used every time your virtual disk space goes up because of some Docker Images you later decide to remove

Mac

- Install Docker from here.
- Download the setup file to a directory of your choice from a Release.
- Run the setup wizard and follow the instructions.
- Remember to run Docker before using Kathará.

Please note that Docker runs on MacOS inside virtual machine running Linux and managed by a proprietary hypervisor.

4.1 Experiment

After you have successfully installed Kathará, which you can verify by calling `kathara check`, consider a small experiment to get a simulation up and running:

1. By calling `kathara --help` in the console, you should be able to see all functionalities Kathará offers.
2. Now let us create a container with the help of Kathará by calling

   ```
   kathara vstart --name hello_world
   ```

3. A console to the container should occur, where you can perform all neccessary Linux commands. Let us briefly display our network interfaces with `ip link show`
4. You successfully started your first virtual network node. Now let us tear it down again with `kathara vclean -n hello_world`.
5. Congratulations, you set up your first single node network.

5 Basic Linux network administration

Setting up a small emulated LAN will be the first task you have to do. You are going to emulate the topology of Figure 1 by the help of Kathará. By working on the following subtasks you will get familiar with very basic Linux network functionality and the first usage of Kathará.

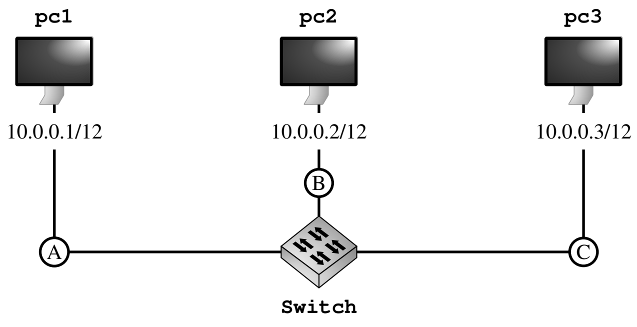

Figure 1: LAN setup

Linux PC	IP Address	Interface	CD
pc1	10.0.0.1/12	eth0	A
pc2	10.0.0.2/12	eth0	B
pc3	10.0.0.3/12	eth0	C
switch		eth0	A
		eth1	B
		eth2	C

Table 1: LAN switching configuration

1. As depicted in Figure 1
   (a) Create three emulated PC's with the help of `kathara vstart` and assign a network interface to each pc, which belongs to a unique collision domain (CD).
   (b) Create another PC. It acts as a switch and has three network interfaces attached, where each interface is connected to a different collison domain.
2. As shown in Table 1
   (a) Configure the three PCs with the command `ip` or `ifconfig` according to the IP addresses of Table 1.
   (b) Configure the switch with the command `brctl`, create a *bridge* named `switch`

and add all interfaces to it.

3. After the configuration, check the connectivity between all hosts with `ping`.
4. Capture the *ICMP* messages you exchange (to check the connectivity) with `tcpdump` on the *switch*.

Generate your jointly edited report using these answers, tables and figures of all subtasks as input. Furthermore, report all problems you encountered when working with Wireshark and Kathará.