INR Lab Assignment
Network Tools

C. Dumitru Jeroen van der Ham Arno Bakker
(Arno.Bakker@os3.nl)

Feedback deadline:
November 10, 2017 10:00 CET

Abstract

The goal of this lab session is to get experience with popular Linux network-related tools. At the end of the session you will know how to collect network information about your system and start troubleshooting network issues beginning with the physical connection and ending with the application. The tasks are grouped under their corresponding layers from the OSI Model.

*For each task besides answering the question, provide the command used and **explain** all the parameters that you've used. **Do not copy-paste** text from man pages or documentation but give a brief and concise answer!*

The Linux networking stack is a fairly complicated part of the Linux kernel. Knowing the basic principles and having some knowledge of the path taken by a network packet from the sending to receiving application will help you in solving network issues. A brief overview of the network stack can be found in Bhuiyan et al. [2008].

Layers 1 and 2 - Physical and Data Link

The network interface card (NIC) is the device that allows a machine to communicate with other devices connected to the same physical layer. Currently, the most popular local area network technology is Ethernet, so most of the focus in this lab will be on Ethernet related tools.

Task 1. Find out what network cards your server has. To what type of computer expansion bus are they connected? What is the speed of this interconnecting bus in mebibytes per second?
Hint: lspci

Task 2. What is the current speed of the network interface? What offload features are enabled? Briefly explain the purpose of the `tcp-segmentation-offload` feature.
Hint: ethtool

Task 3. What is the MAC address of the OS3 router facing your server? Can you infer the manufacturer of the network card? What about the MAC address of `eth0`/`eno1` and its manufacturer?
Hint: arp

Task 4. Assuming that you have completed the previous lab, what interfaces are part of the `xenbr0` bridge? What MAC addresses has this bridge learned so far?
Hint: brctl

Task 5. How many bytes did your `eth0`/`eno1` interface receive since boot? The kernel uses an unsigned long long variable for the RX bytes counter. How much traffic (in GB) must the server receive for this value to overflow?
Hint: ifconfig

Task 6. What is the MTU setting for `eth0`/`eno1`? When do you think it should be increased? When do you think it should be decreased?
Hint: ip link, ifconfig

Layer 3 - Network

The routing table is used by the kernel to decide where to forward received packets. You can inspect the current routing table using the `route` and `ip route` commands.

Task 7. What is the default gateway on your server? Why is there an explicit route to the OS3 network? If you would delete this latter route will you be able to send traffic to your default gateway? Why?

Task 8. Perform a traceroute to `bad.horse`. Why does it stop after 30 hops? How can you increase this number? Provide the full traceroute output.
Hint: mtr, traceroute

The Linux kernel provides a very powerful and flexible packet filtering framework called `netfilter`. One of the user space tools that provides interaction with it is called `iptables`.

Task 9. What are the three built-in chains in the `netfilter` 'filter' table? Briefly explain what is the purpose of each chain.

Layer 4 - Transport

Task 10. What ports are currently open on your machine? What services do they belong to?
Hint: netstat

Task 11. How many **unix sockets** are currently created on your server? What are unix sockets used for?
Hint: lsof

Layer 7 - Application

Task 12. How can you test that a machine is listening on a specific TCP port? Can you do the same for UDP? Why?
Hint: nc, telnet

Task 13. What is the type and version of the webserver that serves `www.os3.nl`?
Hint: curl, wget

Packet capturing

Packet capturing is very useful to gain insight on the behavior of networked devices. `tcpdump` is a popular packet analyzer based on the `libpcap` library.

Task 14. Make sure that the **Guest-01** VM is turned off. On your Dom0 start `tcpdump` and make sure it listens only on the `xenbr0` interface. All captured packets should be saved to a file called `capture.pcap`. Start **Guest-01** and after it boots ping `www.os3.nl` with a packet count of 10, each packet having a payload of 1024 bytes. Once the ping process exits, stop `tcpdump` and inspect the captured file using `tcpdump` or Wireshark. What is the size of each ICMP packet? Why is it not 1024?

References

H. Bhuiyan, M. McGinley, T. Li, and M. Veeraraghavan. TCP Implementation in Linux: A Brief Tutorial. `http://www.ece.virginia.edu/cheetah/documents/papers/TCPlinux.pdf`, 2008.