Skills that should be learned in this course:

* Install Virtualbox, Ubuntu 16.04
* Bash – useful for all kinds of things, grep, ldconfig, cat, nano, | operator, find, ls, cd, git, uname, ssh, scp, df, du, mv, cp, ping, nc, shutdown, reboot, kill, pkill, killall, rsync, ps, history, chmod, screen, rm, if [[ ]], for, watch
* SSH skills – tunnel, keyless login
* Linux – Ubuntu + Raspbian, apt, dpkg
* Basic gcc, Makefiles, make, make install
* Networking in Python
* Networking in C++
* Libboost
* Wireshark, tcpdump, tshark, filters (display + capture), editcap, tcpreplay
* Basic understanding TCP, UDP, DHCP, IP, ICMP, ARP, Ethernet, DNS
* Packet crafting - Libtins, libpcap, Pypacker
* Wireshark plugins
* What is a socket, types of socket
* Mininet + POX
* Iperf, with multicasting
* Basic understanding of Linux kernel modules
* What happens when a packet gets to an interface?
* Tap / TUN devices
* iptables
* Routing?
* Promiscuous listening
* CLI Wireless tools – wpa\_cli, iw, ifconfig, ip tools, route, iwconfig
* Omnet++, NS-3, ?

ACTIVITY 1: Bash primer

1. grep, ldconfig, cat, nano, | operator, find, ls, cd, git, uname, ssh, scp, df, du, mv, cp, ping, nc, shutdown, reboot, kill, pkill, killall, rsync, ps, history, chmod, screen, rm, if [[ ]], for, watch
2. Installing Virtualbox, XUbuntu 18.04
3. Start with simple single commands
   1. Status -> ls, cd , ps, history, ifconfig, find, reboot, shutdown, pkill, killall, & operator, fg, cat, ldconfig, watch –n0 ifconfig, watch –n0 cat
4. Piping commands, single line commandline
   1. ls | grep
   2. find | grep
   3. ldconfig | grep
5. Cmdline editing vim vs nano
6. Scripting, set goals to accomplish
7. For and if in scripts

Assigment 1: Using BASH scripts to send, receive

and manipulate files

This week’s assigment require that you write a BASH script that performs the

following tasks:

1. Use a for-loop to create a text file from cc\_example\_text.txt that is

roughly 40 MB in size.

2. Transfer the large file using nc and scp. Time the transfer times using

time command. Whatever comes after the time command will be timed,

for example

time ls

The script should accept three commandline arguments: the IP address

and port of where nc should connect to and the third argument is the

username scp should use. You only need to write the sending side in the

script that you submit, but it is suggested that you test the script on a

interface on your PC.

3. Check file correctness by calculating the differences with diff. Read man

entries.

4. Send only the first 10 lines of the cc\_example\_text.txt over nc. Hint use

head. Repeat, but send only the last 10 lines.

5. Send only lines containing the word “have”.

6. Finally, submit your work to a GitHub repository using git from the

commandline. You should create your own public GitHub repository, and

send the link to the lecturers of this course.

Activity 2 : Networking in Linux

1. Installing programs in Linux using apt-get – Wireshark, tcpreplay
2. Installing programs in other ways: Sublime Text 3 https://www.sublimetext.com/docs/3/linux\_repositories.html, libtins <https://libtins.github.io/download/> , pypacker <https://github.com/mike01/pypacker>
3. Write a Python Hello World and running
4. Writing a Hello World and compiling with g++.
5. Write a simple Makefile to run g++. Maybe very very simple Makefile improvements...?
6. Wireshark – basic + display filters – filter on IP, udp, icmp type, etc etc. See what happens to ping, nc, nc -u, scp
7. tcpdump, tshark, capture filters, reading dumps with Wireshark
8. Create a socket in Python. Send a group of bytes. Also specifically send a handcrafted bytes array, i.e. not using a packet crafting tool. Use wireshark to check what happened.
9. Create a libboost socket in C++. Send a random byte group as well as a handcrafted bytes array. Use wireshark to check what happened.
10. Record a packet dump from sending the packets, and replay it using tcpdump.

Assignment 2: Write your own ping request + ping responder. If a packet is sent to the localhost, it should not be answered by a ping response. However, write a program that crafts a ping response and sends it back to the original sender. Also craft a UDP counting the ping requests and send this to a separate nc process.

*Ping packets crafted and sent to local address will not be answered: you need to write a program that answers them yourself. ping commands will be answered, but only because loopback is set up to allow for ping requests.*