Department of Informatics

NETWORK COMMANDS AND SNIFFING

Network Security

Diego Sempreboni

INTRODUCTION

Welcome to the second lab of Network Security. In this lab, you will get introduced to the Linux commands used for networking purposes, and you will also learn how to conduct sniffing by using both tcpdump and Wireshark.

INTRODUCTION

At the command prompt, type in "`ifconfig`" to find out all the interfaces that your computer has been configured with.

From the output of ifconfig, can you figure out which interface is the ethernet interface, and which one is the "loopback interface"? (Hint: Loopback interface is always assigned the IP address 127.0.0.1).

Answer: ethernet is usually eth0 or em0 etc.

Google for "loopback interface" and find out what it can be useful for.

List the flags assigned to the ethernet interface. Use google to find out what each flag means.

At the command prompt, type in "`netstat -i`". This should give you an equivalent view of network interfaces available on your computer.

netstat is a generic command for printing all sorts of network information. Try typing in "`netstat -r`" and just "`netstat`" and explain the output that you see. (Hint: Use "man netstat" to understand the output you see).

ARP is used to establish the mapping between IP addresses at the network layer and the corresponding ethernet address at the data link layer. Type in "`arp`" to see the arp cache on your machine.

You can also obtain equivalent information about ARP mappings by typing in "`ip neighbour`". Try other related commands such as "`ip route`" and "`ip link`". What information did you obtain? What previous commands gave you similar information as ip route and ip link?

Answer: the command ip is new to linux. Other unices and even some linux distributions don't have the ip command. The other commands such asifconfig, netstat and arp provide similar information.

SNIFFING - TCPDUMP

Next you will listen to packets passing through your network using `tcpdump`. `tcpdump` can only be run as a superuser (i.e., as “root”. To run as root, you will need to prepend all commands with “sudo” - e.g., “`sudo tcpdump`”). If you do not have sudo permission on your system, you can try this on a virtual machine at http://linuxzoo.net (you will have to register a username. Linuxzoo gives root access by default so you do not have to use “sudo”).

Listen to all packets on the ethernet interface. You can do this by:

```
sudo tcpdump -i <ethernet-interface-name>
```

What is the name of the ethernet interface? (Hint: Can any of the commands from the previous section help here?)

Listening to all packets prints out too much information (there are a lot of packets!). For normal operation and diagnostics, you will want to filter information. This can be done by specifying BPF filters. For example, the following will restrict to only icmp (ping protocol) packets:

```
sudo tcpdump icmp
```

Now if you ping (icmp protocol) some website (say google.com) it should show up in the tcpdump window. You can do this by typing:

```
ping google.com
```

You can restrict to seeing a fixed number of packets, for example the ping and its response (pong). To do this type in

```
sudo tcpdump -c2 icmp
```

Try the ping command again. Notice that tcpdump finishes after 2 packets, whereas the ping command continues to receive responses back. What are the types of the two packets??

When you ping ‘google.com’, you expect a response from google.com. What host were you getting the response from? (Hint: use the lecture slides or “man tcpdump” to identify the source of the ICMP echo request or the destination of the echo reply (pong)). Is this the host you were expecting? Why or why not? (Hint: can a single host have multiple DNS names?)

You can avoid dealing with DNS names, by asking tcpdump to use IP addresses (use the -n switch):

```
sudo tcpdump -n —c2 icmp
```

Repeat the ping. Does the IP address for google correspond to the ping output?

Answer: Obviously, it should - if you can’t see it check with your TA.

tcpdump has lots of settings. Check them out. Maybe you can try some of them.

- -X : Show the packet’s contents in both hex and ascii.
- -XX : Same as -X, but also shows the ethernet header.
- -D : Show the list of available interfaces
- -l : Line-readable output (for viewing as you save, or sending to other commands)
- -q : Be less verbose (more quiet) with your output.
- -t : Give human-readable timestamp output.
- -tttt : Give maximally human-readable timestamp output.
- -i eth0 : Listen on the eth0 interface.
- -vv : Verbose output (more v’s gives more output).
- -c : Only get x number of packets and then stop.
- -s : Define the snaplength (size) of the capture in bytes. Use -s0 to get everything, unless you are intentionally capturing less.
- -S : Print absolute sequence numbers.
- -e : Get the ethernet header as well.
- -q : Show less protocol information.
- -E : Decrypt IPSEC traffic by providing an encryption key.

SNIFFING - WIRESHARK

You can obtain much of the above functionality in a GUI using wireshark. Type in

```
sudo wireshark
```

to start it up and explore. When wireshark starts, you need to choose a network interface to monitor. Which one would you choose, based on the above experience?

See if you can right click and follow a TCP stream.

If you only want to obtain http traffic, explore how you might do this by entering a filter expression at the top of the window. What filter expression is needed for http?

Answer: You can just type in http. This is a convenience which is not possible in tcpdump.

Try to execute some complex filters on wireshark in order to isolate traffic you might be interested.

Further details here:

https://www.wireshark.org/docs/wsug_html_chunked/ChWorkBuildDisplayFilterSection.html