ENSEA
Ecole Nationale Supérieure de l'Electronique et de ses Applications

3RTS – Architecture & Protocols

TP #2 - room D055

C. BARÈS

Objectives : Use advanced network commands and study them with Wireshark.

Preliminary Notes

The computers in the room D055 are networked using the sepcific network interface (grey cable). Some of the operations described in these labs require you to be an administrator of the machine. For this reason, these sessions can only take place in room D055, with with a special login.

To connect in room D055, you will use the login "tpresau" associated with the password "sethisis". To access the superuser mode, use the sudo command, and to open a root shell the sudo -s <command>.

To access the internet (no matter which tool you use), you will first need to authenticate yourself on the ucopia captive portal.

If a command <command> is missing, install it by typing :

```
$ sudo apt install <commande>
```

Information about an command can be obtained by typing :

```
$ <command> -h
$ man <command>
```

1 – Capture FTP session

Set up the filters to capture an FTP session between PC100 (192.168.28.100) and yourself :

- Using an ftp filter, perform the capture on Wireshark and comment out an entire sequence of ftp traffic (establishing the connection, sending/receiving a file...)
- Using a tcp filter (tcp and ip.addr == ftpServerAddress). Perform the capture again, and analyse the result. Analyse the TCP connection opening (TCP signalling, port number, mss negotiation, sequence number, windows). You can use the Wireshark tool to display connection graphs (Statistics > Flow Graph > TCP Flow).

Focus your attention on the evolution of the sequence and acknowledgement numbers in order to understand it well. Analyse the different commands/responses exchanged (sequencing, explanation of the different commands).

By the way, how do you make an FTP request ?

```
$ ftp 192.168.28.1
```

username : tpreseau, pass : sethisis. From there, we can use ls, get, send...

2 – MTU (Maximum Transfer Unit)

- Identify the default value assigned to the MTU parameter on your main interface
- Explain the role of this parameter
- MTU and ICMP :
  - Change the MTU value from enpXsY to 100 using ifconfig or ip addr
  - Ping a neighbouring machine by sending ICMP packets with 50 bytes of data (-s option) and run the capture on Wireshark. Can an ICMP packet of this size be sent in a single Ethernet frame of MTU 100 ?
  - Can an ICMP packet of 80 bytes be sent in a single Ethernet frame of MTU 100 ?
- MTU and ftp : We are now going to measure the impact of the modification of the MTU parameter on the file transfer in ftp towards the PC1 machine.
  - Monitor the number of blocks transferred on a large file according to the MTU using Wireshark (MTU=1500, then MTU=100)
  - What can you conclude ?
- Use the tracepath software to determine the best MTU value to access the Internet from the ENSEA network.

3 – TCP Window size

In this section, you will demonstrate the influence of TCP Window Size.

Create a python script called slow_tcp.py containing the following code (be careful to respect the indentation, use only spaces) :

```
1  import socketserver
2  import time
3
4  class MyTCPHandler(socketserver.StreamRequestHandler) :
5      def handle(self) :
6          while True :
7              data = self.rfile.readline().strip()
8              if not data : return
9              print("{} wrote : ".format(self.client_address[0]), data.decode
                   ('utf8'))
10             # time.sleep(1)
11
12 if __name__ == "__main__" :
13     HOST, PORT = "localhost", 9999
14     server = socketserver.TCPServer((HOST, PORT), MyTCPHandler)
15     server.serve_forever()
```

What does this script do ?

Then, after starting an acquisition with Wireshark (be careful to choose your filters !), run the following commands in **3 different terminals** :

- python slow_tcp.py
- while true; do echo "foo" > stream.txt; done
- tail -f stream.txt | nc localhost 9999

What can you say about the value of the Window fields ?

Repeat the same procedure after uncommenting line 10 of the previous program. Now, what can you tell about the value of the Window fields ? (the observation is quite long – up to 2 minutes)

4 – Capturing a Web session with Telnet

Telnet allows you to emulate a remote terminal. Using telnet you can see if the server is running on a port number you define and thus examine exactly what is returned to your browser when a web request for a particular URL is made. This allows you to see the headers and gather other information about the web server.

To determine the syntax of an HTTP GET request, you can look it up on the internet.

Once you have determined the syntax, open a communication socket with the command : telnet name.of.target.site 80 Be careful, in some cases, you will have a limited time to type your request. If this happens, you will have to prepare a copy/paste of the query.

5 – Ssh protocol

The ssh (secured shell) protocol is a modern variant of telnet, allowing you to open a shell remotely, using asymmetric encryption. This allows you to open a remote shell window on a machine in the room and gain root access. This manipulation then allows you to do various tricks. Try for example the command "wall". **DO NOT** do a "reboot" command !

Analyse the TCP segments exchanged in an ssh dialogue. What do you observe ?