

ECE Paris

Computer Networks

ING 4 – SI (Information Systems, Big Data and Cyber Security)

Rafik ZITOUNI: rafik.zitouni@ece.fr

Takvor MAGADIS: tmagadis@inseec-edu.com



LAB 4

Contents

I.	Objective.....	1
II.	Getting started	2
III.	The basic HTTP GET/response interaction	2
	1. Retrieve HTML file	2
	2. Retrieving Long Documents	3
IV.	DNS.....	4

I. Objective

After this lab, the students will have skills using Wireshark and then add the tool to their CV.

1. Wireshark packet capture
2. HTTP get/response
3. DNS

- Each group (composed of **3 students** at most) shall submit a report **in campus**.
- The report (**PDF format is accepted**) shall be uploaded on the campus page before the deadline **November 29 at 23h55**.

II. Getting started

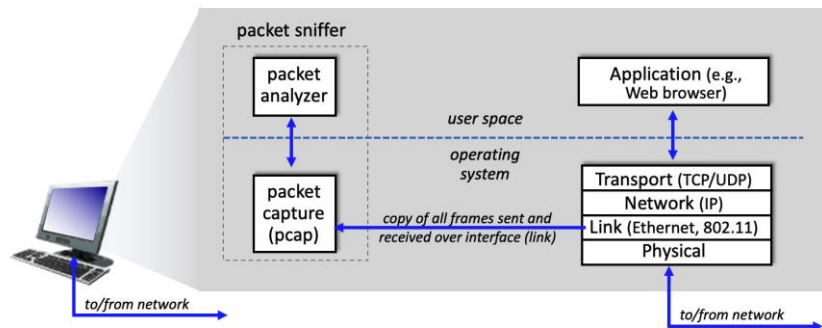


Figure 1 Structure of Packet Sniffer

Figure 1 shows the structure of a packet sniffer. At its right you have the protocols (in this case, Internet protocols) and applications (such as a web browser or email client) that normally run on your computer. **The packet sniffer**, shown within the dashed rectangle is an addition to the usual software in your computer, and consists of two parts: 1) **The packet capture library** and 2) **the packet analyzer**. The first receives a copy of every link-layer frame that is sent from or received by your computer over a given interface (link layer, such as Ethernet or WiFi). Recall that messages exchanged by higher layer protocols such as HTTP, FTP, TCP, UDP, DNS, or IP all are eventually encapsulated in link-layer frames that are transmitted over physical media such as an Ethernet cable or a WiFi interface. Capturing all link-layer frames thus gives you all messages sent/received across the monitored link from/by all protocols and applications executing in your computer. **The packet analyzer** displays the contents of all fields within a protocol message. It must “understand” the structure of all messages exchanged by protocols.

III. The basic HTTP GET/response interaction

1. Retrieve HTML file

We begin our exploration of HTTP by downloading a very simple HTML file

1. Start up your web browser (Mozilla, Chrome, etc) .
2. Start up the Wireshark and enter “**http**” (without the quotation marks) in the display-filter-specification window.
3. Wait a bit more than one minute (we’ll see why shortly), and then begin Wireshark packet capture.
4. Enter the following to your browser
<http://networktests.000webhostapp.com/HTTP-wireshark-file1.html>
Your browser should display the very simple, one-line HTML file.
5. Stop Wireshark packet capture.

Your Wireshark window should look similar to the window shown in Figure 2

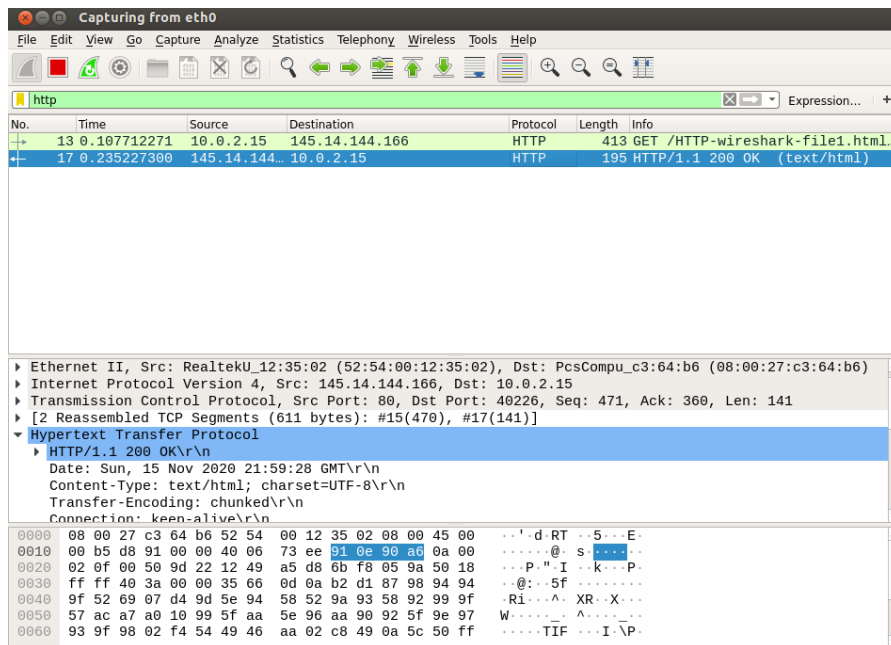


Figure 2 Wireshark packet capture

- Q1) Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server running?
- Q2) What languages (if any) does your browser indicate that it can accept to the server?
- Q3) What is the IP address of your computer? Of the 000webhostapp.com server?
- Q4) What is the status code returned from the server to your browser?
- Q5) When (hour and date) was the HTML file that you are retrieving has been received?
- Q6) How many bytes of content are being returned to your browser?
- Q7) By inspecting the raw data in the packet content window, do you see any headers within the data that are not displayed in the packet-listing window? If so, name one.

2. Retrieving Long Documents

Most web browsers perform object caching and thus perform a conditional GET when retrieving an HTTP object. Before performing the steps below, make sure your browser's cache is empty. (To do this under Firefox, select *Tools->Clear Recent History* and check the Cache box, or for Internet Explorer, select *Tools->Internet Options->Delete File*; these actions will remove cached files from your browser's cache.) Please do the same if your navigator is Chrome or IE, then follow the steps below:

1. Start up your web browser, and make sure your browser's cache is cleared, as discussed above.
2. Start up the Wireshark packet sniffer
3. Enter the following URL into your browser
http://networktests.000webhostapp.com/US_Bill_Rights.html
4. Your browser should display the rather lengthy US Bill of Rights.
5. Stop Wireshark packet capture, and enter "http" in the display-filter-specification window, so that only captured HTTP messages will be displayed. Do the same for "tcp". In the packet-listing window, you should see your HTTP GET message, followed by a multiple-packet TCP response to your HTTP GET request.

Answer the following questions:

Q8) How many HTTP GET request messages did your browser send? Which packet number in the trace contains the GET message for the US Bill or Rights?

Q9) Which packet number in the trace contains the status code and phrase associated with the response to the HTTP GET request?

Q10) What is the status code and phrase in the response?

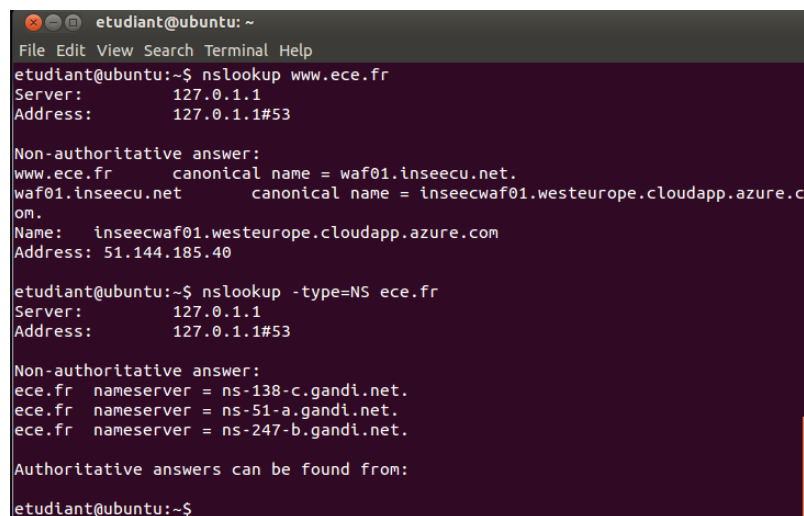
Q11) How many data-containing TCP segments were needed to carry the single HTTP response and the text of the Bill of Rights?

IV. DNS

The Domain Name System (DNS) translates hostnames to IP addresses. In this part, you will take a closer look at the client side of DNS. Recall that the client's role in the DNS is to send a *query* to its local DNS server, and receives a *response* back.

1. nslookup

nslookup tool or command line is available in most Linux/Unix and Windows. To run *nslookup* in Linux/Unix, you just type the *nslookup* command on the command line. The same for Windows, open the Command Prompt cmd and run *nslookup*.



```
etudiant@ubuntu: ~  
File Edit View Search Terminal Help  
etudiant@ubuntu:~$ nslookup www.ece.fr  
Server:      127.0.1.1  
Address:     127.0.1.1#53  
  
Non-authoritative answer:  
www.ece.fr   canonical name = waf01.inseecu.net.  
waf01.inseecu.net canonical name = inseecwaf01.westeurope.cloudapp.azure.com.  
Name:   inseecwaf01.westeurope.cloudapp.azure.com  
Address: 51.144.185.40  
  
etudiant@ubuntu:~$ nslookup -type=NS ece.fr  
Server:      127.0.1.1  
Address:     127.0.1.1#53  
  
Non-authoritative answer:  
ece.fr nameserver = ns-138-c.gandi.net.  
ece.fr nameserver = ns-51-a.gandi.net.  
ece.fr nameserver = ns-247-b.gandi.net.  
  
Authoritative answers can be found from:  
etudiant@ubuntu:~$
```

Figure 3 Screenshot of the result of nslookup

This command is saying “please send me the IP address for the host *www.ece.fr*”. As shown in the screenshot, the response from this command provides two pieces of information: (1) the name and IP address of the DNS server that provides the answer; and (2) the answer itself, which is the host name and IP address of *www.ece.fr*.

Q12) Explain briefly the second command line *nslookup -type=NS ece.*

Q13) Run *nslookup* to obtain the IP address of google web server for .fr, .de and .com. Comment the obtained result?

2. Tracing DNS

In this part, you capture the DNS packets that are generated by ordinary Web-surfing activity. Please follow these steps:

- 1) If you are on Windows, use *ipconfig* to empty the DNS cache in your host.
- 2) Open your browser and empty your browser cache.
- 3) Open Wireshark and enter “ip.addr == your_IP_address” into the filter, where you obtain your_IP_address with ipconfig (or ifconfig on Linux). This filter removes all packets that neither originate nor are destined to your host.
- 4) Start packet capture in Wireshark.
- 5) With your browser, visit the Web page: **<http://www.ece.fr>**
- 6) Stop packet capture.

Q14) Use your “ipconfig/all” (windows) to get more information about your network. If you are on Linux you can use the command line “nmcli dev list”

Q15) Locate the DNS query and response messages for www.ece.fr . To filter the query and response add in your filter the expression (dns.qry.name contains www.ece.fr). Are these messages sent over UDP or TCP?

Q16) What is the destination port for the DNS query message? What is the source port of DNS response message?

Q17) To what IP address is the DNS query message sent? Use ipconfig to determine the IP address of your local DNS server. Are these two IP addresses the same? Use the result of ipconfig/all to answer.

Q18) Examine the DNS query message. What “Type” of DNS query is it? Does the query message contain any “answers”?

Q19) Examine the DNS response message. How many “answers” are provided? What do each of these answers contain?

Q20) Consider the subsequent TCP SYN packet sent by your host. Does the destination IP address of the SYN packet correspond to any of the IP addresses provided in the DNS response message? Propose a filter expression to capture only SYN packets.