Name 1:
Name 2:

COM-407: TCP/IP NETWORKING

LAB EXERCISES (TP) 0
BASIC CONFIGURATION, IP SUITE, AND PACKET INSPECTION: PING(6), TRACEROUTE(6), NETSTAT, NSLOOKUP

With Solutions

September 19, 2018

Abstract

In this lab you will practice some networking commands that enable you to obtain information about Internet machines and about the connectivity and the paths between them. You will also learn to use a GUI-based packet capture/inspection tool called Wireshark. Optionally, in research exercises, you will use tshark (command-line version of Wireshark) for packet capture/inspection.

1 ORGANIZATION OF THE TP AND WIRESHARK TOOL

1.1 TP REPORT

Type your answers in this document. We recommend you use Adobe Reader XI to open this PDF, as other readers (such as SumatraPDF, but also older versions of Adobe!) don't support saving HTML forms. That will be your TP report (one per group). When you finish, save the report and upload it on moodle. Don't forget to write your names on the first page of the report. **The deadline for report submission is September 26 (Wednesday) 11.55 PM.**

1.2 WIRESHARK

You will be using Wireshark to sniff packets. Since there are a lot of packets generated by the applications running on your machine, you may want to use filters. `http://wiki.wireshark.org/DisplayFilters` Please note that there are two types of filters: *capture* and *display*. Capture filters are used to selectively capture the traffic whereas with display filters, you capture all the traffic but the traffic is displayed as per the filter rules.

2 THE IPV4 INTERNET

Connect to the Internet in IPv4 and disable IPv6 connectivity, if needed.

To disable IPv6:

On MacOSX, you can execute the following command from the *Terminal* app

```
# networksetup -setv6off "InterfaceName"
```

If you do not know the InterfaceName, you can use the following command

```
# networksetup -listallnetworkservices
```

On Red Hat-based Linux, following commands can be used from the *Terminal* app

```
# sudo sysctl -w net.ipv6.conf.all.disable_ipv6=1
# sudo sysctl -w net.ipv6.conf.default.disable_ipv6=1
```

On Debian-based Linux, add the following in /etc/sysctl.conf file and reboot the machine.

```
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1
net.ipv6.conf.lo.disable_ipv6 = 1
```

On Windows, Click Start, and then click Control Panel. Click Network and Sharing Center. In the View your active networks area, click Local Area Connection, and then click Properties. On the Networking tab, clear the Internet Protocol Version 6 (TCP/IPv6) check box, and then click OK.

After disabling the IPv6 connectivity, in order to determine the following information:

- the IP address(es) of your machine `<my_ip>`,
- the netmask `<my_netmask>`, and
- the default gateway of your machine `<my_gateway>`.

In MacOS use following commands in *Terminal* app

```
# ifconfig
# netstat -nr
```

In Linux use following commands in *Terminal* app

```
# ip addr show
# ip route show
```

or in Windows use following commands in *cmd* app

```
> ipconfig /all
```

Q1/ List your findings here:

- IP address: 128.178.151.219
- Network Mask: 255.255.255.0
- Default Gateway: 128.178.151.1

Q2/ Is your IP address public or private? What does the netmask in IPv4 mean? Why a default gateway is configured?

Solution. *In this case, the IP address is public, which can be confirmed by navigating to the link* `http://www.myipaddress.com` *and confirming that the IP address given in the web page is the same as the one given to the Ethernet adapter. OR We have a range of private ip addresses and as we can see that this ip address does not belong to this range, it is public. The netmask or prefix is used to distinguish the "network" and the "host" parts of an IP address.*

Now, download Wireshark and install it on your computer. Start it (as administrator) and use the menu `Capture->Interfaces` to start capturing packets on the interface that you are currently using for the Internet connectivity.

Q3/ Do you see any packet captured with destination IP address of your default gateway if you navigate to a webpage through your browser? If yes/no, explain the reason behind your observation?

Solution. *No, unless you are pinging your default gateway or communicating directly with it by any mean (DNS, FTP, HTTP, SCP, etc). In IP, communication is done end-to-end thus in general we should not see IP packets with destination IP address of any of the intermediate devices, including the default gateway.*

2.1 PING

PONG

The ping command uses the ICMP protocol to probe whether a host is up:

```
# ping <hostname>
```

Q4/ Start a new capture with Wireshark and then ping `www.facebook.com`. Observe the traffic generated by the ping command. Do you see only ICMP packets?. Stop the ping program and start it again after a couple of seconds. Is there a difference from the first captured packets? Explain.

```
128.178.151.139 128.178.15.227 DNS 74 Standard query 0x59d0  A www.google.com
28.178.15.227 128.178.151.139 DNS 154 Standard query response 0x59d0  A 173.194.35.17
      A 173.194.35.18 A 173.194.35.19 A 173.194.35.20 A 173.194.35.16
128.178.151.139 173.194.35.18 ICMP 74 Echo (ping) request  id=0x0001, seq=3/768, ttl=128
173.194.35.18 128.178.151.139 ICMP 74 Echo (ping) reply    id=0x0001, seq=3/768, ttl=52
```

Solution. *First a DNS query is performed, next a ping request is sent to the IP address of facebook.*

The second time the DNS request is typically not performed. The IP address was cached.

Other valid observations: the ARP request for the gateway is not performed either (ARP cache), the sequence numbers continue from where they left off during the first ping, another IP address is used (due to Facebook's load balancing system), etc.

Q5/ In a browser open `www.swisscom.ch`. Next, try pinging it. Explain.

Solution. *The server hosting the website is up, yet it is configured not to respond to ping (ICMP is disabled).*

2.2 TRACEROUTE AND NETSTAT

traceroute is a tool for displaying the route to a destination.

In MacOS and Linux:

```
# traceroute www.facebook.com
```

In Windows:

```
> tracert www.facebook.com
```

Q6/ Do you see more than one name/IP address at any of the hops? If so, why? Which OS (Linux, MacOSX, or Windows) are you running?

Solution. *Yes, on some hopes (e.g. 9 and 10) we see three different IP Addresses. For discovering hops, traceroute sends UDP packets with increasing TTL to the UDP echo port (tracert in Windows uses ICMP echo requests). For each hop, 3 packets are sent. At hops 9 and 10, the three packets are sent on 3 different paths because of a load-balancer on the way that is trying to balance loads on three different paths. The three paths are not seen on Windows machine because the load-balancer ignores ICMP traffic.*

netstat is a tool for displaying TCP connections, routing table, interfaces and network statistics. Open a web browser, go to `lca.epfl.ch`, and leave the browser open for the moment.

Look at the active TCP connections.

```
# netstat -t -n
```

The `-n` switch prevents name resolving and makes netstat display results faster (but obviously without the names of the hosts).

Q7/ Identify the TCP connections opened by visiting the `www.epfl.ch` webpage. Write them down and describe them here. Is there one, or are there several such connections? Why?

Solution. *Several connections are established, as modern browsers load in parallel the HTML document and the graphics (images, sound, etc.).*

3 NAMES IN THE INTERNET

Juliet: [...]
What's in a name? That which we call a rose
By any other name would smell as sweet.

W.S.

Replace your DNS servers by an inexisting IP address, say `1.2.3.4`. If you configured statically your DNS servers, don't forget to write them down somewhere before changing them to `1.2.3.4`.

Go to the `Properties` of your Internet connection. Click on Internet Protocol Version 4, `Properties`, choose `Use the following DNS server addresses`, and write `1.2.3.4`

Use the manual configuration in the network settings and set the DNS address to `1.2.3.4`

Switch to root mode using `su` and edit the `/etc/resolv.conf` file. Comment out the lines that begin with `nameserver` (precede them with the `#` character) and add one line `nameserver 1.2.3.4`

Q8/ Try pinging Facebook and observe the traffic with Wireshark. What happens?
Solution. *A DNS request is sent to the bogus server `1.2.3.4` with no reply back*

Q9/ Try pinging the IP address of Facebook that you discovered in Sections 2.1 and 2.2. Does it work?
Solution. *Since there is no need to resolve a name, the ping to Facebook's IP address works fine.*

nslookup is a command-line tool for querying Domain Name System (DNS) name servers. Run `nslookup` with the address of the Google public DNS server.

```
# nslookup - 8.8.8.8
```

Q10/ In the > prompt, type `lca.epfl.ch`. Give the IPv4 and IPv6 addresses of `lca.epfl.ch`. Use `set type=A` for IPv4 or `set type=AAAA` for IPv6

```
icsil1noteb147:~ barreto$ nslookup
> set type=A
> lca.epfl.ch
Server: 2001:620:618:10f:1:80b2:f07:1
Address: 2001:620:618:10f:1:80b2:f07:1#53

lca.epfl.ch canonical name = lca1srv2.epfl.ch.
Name: lca1srv2.epfl.ch
Address: 128.178.156.24
> set type=AAAA
> lca.epfl.ch
Server: 2001:620:618:10f:1:80b2:f07:1
Address: 2001:620:618:10f:1:80b2:f07:1#53

lca.epfl.ch canonical name = lca1srv2.epfl.ch.
lca1srv2.epfl.ch has AAAA address 2001:620:618:19c:1:80b2:9c18:1
```

Solution. *IPv4 address:* `128.178.156.24`
IPv6 address: `2001:620:618:19c:1:80b2:9c18:1`

Q11/ Do you recognize the IPv4 address in the IPv6 address, or vice-versa?

Solution. *An IPv4 address,* `128.178.156.24`*.*

An IPv6 address, `2001:620:618:19c:1:80b2:9c18:1`*.*

There is a mapping between IPv4 and IPv6 addresses (IPv4 appears in the IPv6 address: `80b2:9c18`*).*

Restore now your initial DNS configuration.

Start a capture in `wireshark` and do a traceroute in IPv4 to `www.facebook.com`. Focus on the line:

```
swiel2 (192.33.209.33)  1.219 ms  0.968 ms  0.944 ms
```

Q12/ Look at the capture and identify the packet in which you see the name `swiel2`. How does this differ from the usual DNS response observed in previous questions? Based on the observed difference, comment on how `traceroute` works.

Solution. *This is a reverse DNS query as opposed to the previous ones.*

The `traceroute` *tool works by sending the udp packet (in case of Linux and MacOSX) and ICMP packet (in case of Windows) with increasing TTL values until it reaches the destination. When TTL expires, the intermediate routers reply and that's how it knows all the intermediate machines. By default the* `traceroute` *tool makes a reverse DNS query for the IP address of each intermediate router, and then it displays the name in the output of the traceroute command. To disable this reverse query (and thus making the command faster), when typing the traceroute command you can use the "*`-n`*" argument in Mac and Linux, or the "*`-d`*" argument in Windows*

Q13/ Analyze the capture and comment on how `traceroute` finds successive hops. ***Solution.*** *By increasing the TTL after each TTL exceeded message.*

4 THE IPV6 INTERNET

Now let's examine the situation when only IPv6 connectivity is present.

Find access to an IPv6 network and **disable IPv4 on your machine**.

To disable IPv4:

On MacOSX, you can execute the following command from the *Terminal* app

```
# networksetup -setv4off "InterfaceName"
```

You can also turn an interface off without using *Terminal* app. Go to System Preferences and then click on Network. Next, click on the interface you want to change its configuration. Then, select Advanced button. In the new window, go to tab TCP/IP. Now, in the configuration of IPv4, you can turn off IPv4 or select Using DHCP for automatic IPv4 assignment.

On Red Hat-based Linux, following commands can be used from the *Terminal* app

```
# sudo sysctl -w net.ipv4.conf.all.disable_ipv4=1
# sudo sysctl -w net.ipv4.conf.default.disable_ipv4=1
```

On Debian-based Linux, add the following in /etc/sysctl.conf file and reboot the machine.

```
net.ipv4.conf.all.disable_ipv4 = 1
net.ipv4.conf.default.disable_ipv4 = 1
net.ipv4.conf.lo.disable_ipv4 = 1
```

On Windows, Click Start, and then click Control Panel. Click Network and Sharing Center. In the View your active networks area, click Local Area Connection, and then click Properties. On the Networking tab, clear the Internet Protocol Version 4 (TCP/IPv4) check box, and then click OK.

If IPv6 networking is disabled (which might be the case if you used the same interface as for second section of the TP), enable it before accessing an IPv6 network.

To re-enable IPv6 for a network interface (if not already enabled):

On MacOSX, you can execute the following command from the *Terminal* app

```
# networksetup -setv6automatic "InterfaceName"
```

On Red Hat-based Linux, following commands can be used from the *Terminal* app

```
# sudo sysctl -w net.ipv6.conf.all.disable_ipv6=0
# sudo sysctl -w net.ipv6.conf.default.disable_ipv6=0
```

On Debian-based Linux, remove the lines you added in /etc/sysctl.conf file while disabling IPv6 connectivity and reboot the machine.

On Windows, Click Start, and then click Control Panel. Click Network and Sharing Center. In the View your active networks area, click Local Area Connection, and then click Properties. On the Networking tab, mark the Internet Protocol Version 6 (TCP/IPv4) check box, and then click OK.

IPv6 access is provided in or around INF019 room via a wireless access point (`SSID: lca2-tcpip-labs, password: tcp-ip-wifi-ap-2017`).

Use wireshark to observe the traffic. On your computer type

```
# ping www.facebook.com
```

Note that If it is not pinging with IPv6 by default, instead of `ping` command, you should use `ping6` on MacOSX and `ping -6` on Windows.

Q14/ Describe some differences in the observed traffic compared to the IPv4 case. Write the average RTT you get and compare it with the IPv4 case. Explain the differences if any.

Solution. *IPv6 and IPv4 packets may take different paths to reach the destination host, also at any given moment we could experience congestion in the network, thus RTT may be different. Differences are also in packet length, protocol used, etc.*

Repeat the test with the `traceroute` command from Section 2. Use:

In Linux or MacOS:

```
# traceroute www.facebook.com
```

Note that if the `traceroute` command is not done by default with IPv6, you should use `traceroute6` command.

In Windows:

```
> tracert www.facebook.com
```

Note that if the `tracert` command is not done by default with IPv6, you should use `tracert -6` command.

Q15/ Write the result. Does the path to Facebook in the IPv6 Internet cross the same routers as in IPv4?

```
icsil1noteb157:~ mohiuddi$ traceroute6 www.facebook.com
traceroute6 to star-mini.c10r.facebook.com (2a03:2880:f11c:8083:face:b00c::25de) from
 2001:620:618:197:1:80b2:97d7:1, 64 hops max, 12 byte packets
 1  cv-ic-dit-v151-ro  0.477 ms  0.281 ms  0.433 ms
 2  cv-gigado-v100  0.388 ms  0.466 ms  0.378 ms
 3  c6-ext-v200  0.511 ms  0.469 ms  0.441 ms
 4  swiel1-10ge-0-0-0-2.switch.ch  1.168 ms  1.156 ms  1.055 ms
 5  swiel2-10ge-5-3.switch.ch  0.898 ms  1.000 ms  0.755 ms
 6  swice2-10ge-4-1.switch.ch  1.671 ms  1.651 ms  1.552 ms
 7  swice3-p23.switch.ch  1.647 ms  1.724 ms  1.712 ms
 8  2001:7f8:1::a500:6762:1  17.286 ms  17.302 ms  17.253 ms
 9  lo0.franco32.fra.seabone.net  30.441 ms  35.908 ms  30.567 ms
10  2001:41a8:600:2::162  17.802 ms  23.524 ms
    2001:41a8:600:2::15e  23.361 ms
```

```
11  po111.asw04.fra2.tfbnw.net  18.316 ms
    po114.asw01.fra2.tfbnw.net  19.627 ms
    po121.asw01.fra2.tfbnw.net  19.966 ms
12  po203.psw01c.frt3.tfbnw.net  18.722 ms
    po204.psw01c.frt3.tfbnw.net  23.821 ms
    po201.psw01b.frt3.tfbnw.net  19.562 ms
13  po3.msw1ac.01.frt3.tfbnw.net  19.592 ms
    po2.msw1ai.01.frt3.tfbnw.net  25.719 ms
    po3.msw1ad.01.frt3.tfbnw.net  28.020 ms
14  edge-star-mini6-shv-01-frt3.facebook.com  24.834 ms  18.878 ms  18.851 ms

tsf-484-wpa-4-040:~ mohiuddi$ traceroute www.facebook.com
traceroute to star-mini.c10r.facebook.com (31.13.64.35), 64 hops max, 52 byte packets
 1  cv-gigado-v484 (128.179.184.1)  72.081 ms  2.642 ms  1.149 ms
 2  c6-ext-v200 (128.178.200.1)  1.126 ms  3.746 ms  1.599 ms
 3  swiel2 (192.33.209.33)  2.093 ms  1.914 ms  1.825 ms
 4  swiel2-10ge-5-3.switch.ch (130.59.36.78)  3.270 ms  2.346 ms  2.210 ms
 5  swice2-10ge-4-1.switch.ch (130.59.37.65)  2.832 ms  3.245 ms  3.616 ms
 6  swice3-p23.switch.ch (130.59.36.210)  3.708 ms  3.118 ms  4.326 ms
 7  br02.ams1.tfbnw.net (80.249.209.164)  19.682 ms  18.969 ms  18.255 ms
 8  be2.bb01.ams3.tfbnw.net (204.15.20.10)  26.151 ms  27.995 ms  26.756 ms
 9  ae21.bb02.ams2.tfbnw.net (31.13.27.66)  25.323 ms  31.258 ms  26.080 ms
10  ae1.pr02.ams2.tfbnw.net (74.119.79.195)  18.956 ms  18.960 ms  18.236 ms
11  po102.psw01c.amt2.tfbnw.net (157.240.32.17)  18.176 ms  18.973 ms  18.399 ms
12  msw1am.01.amt2.tfbnw.net (173.252.66.217)  18.807 ms
    msw1ac.01.amt2.tfbnw.net (173.252.65.1)  22.155 ms
    msw1al.01.amt2.tfbnw.net (173.252.66.219)  19.376 ms
13  edge-star-mini-shv-01-amt2.facebook.com (31.13.64.35)  20.706 ms  20.008 ms  21.127 ms
```

Solution. *There are some routers with the same name in the two cases. It is not impossible that they are dual-stack routers. The path is however not identical!*

Now, open the web browser (new window), go to `lca.epfl.ch`.

Q16/ Do you notice a difference between two versions of `lca.epfl.ch` pages? Can you imagine by which mechanism such a difference may occur ?
Hint: Which device (default gateway, intermediate routers, the web server, etc) do you think is in charge of displaying the web content for IPv4 if you are connected to an IPV4 network or for IPv6 otherwise?

Solution. *There is an IPv6 logo at the bottom of the page. The Route Rank widget does not work in IPv6.*

Who put this logo on the page we received ? The web server did it. In this case the same web server is reached over IPv4 and IPv6 (in other settings they might be different) but the web server itself, when it is contacted by a client, knows on which network (IPv4 or IPv6) the HTTP request arrives (based on sockets, as we will see later in the course). The web server then runs a script that puts the IPv6 logo in the page when the request arrived over IPv6. Intermediate systems are of course not involved in this.

Look at the active connections.

```
# netstat -t -n
```

Q17/ Compare the output that is related to `www.epfl.ch` with the one that you wrote down for IPv4. Comment about it

Solution. *We can see that the transport layer (TCP) connections are different for IPv4 and IPv6 networks.*

Q18/ Try pinging `www.swisscom.ch` again. Did it work? Explain. ***Solution.*** *It works. IPv4 and IPv6 configurations are run separately in routers. It is likely that ICMP is not disabled in the IPv6 interface of* `www.swisscom.ch` *website).*

5 IPv4 AND IPv6

Let's see what happens when both IPv4 and IPv6 Internet connectivities are present. **Stay connected in IPv6, but enable IPv4.**

From your computer do a traceroute in IPv4 and IPv6 to `www.switch.ch`

Q19/ Does it work in both cases?. Write down any difference in the traceroutes

Solution. *Traceroute works in both cases, and they traverse different routers.*

Now, start a new `Wireshark` capture, open a browser and type `www.switch.ch`.

Q20/ Check the capture in Wireshark, your connection to the webpage is done in IPv4 or in IPv6? ***Solution.*** *On Mac, it prefers IPv6 if available*

Q21/ Explain how do you think your machine could decide whether it uses IPv4 or IPv6.

Solution. *It depends on your machine but in general IPv6 is preferred over IPv4 and decision is based on the DNS query. If the target host has an `AAAA` record, your machine tries an IPv6 connection; if not it goes for IPv4. However, some vendors have decision-making algorithms that tracks the latency on the IPv4 or IPv6 network and based on that decide which network they will use.*

RESEARCH EXERCISES (OPTIONAL)

6 NETWORK PACKET INSPECTION

We need to see or inspect the packets leaving or coming to our computer or other computers for various reasons. These reasons vary depending on the person and his motivations. For example, network administrators need this for troubleshooting network-related problems, software developers for debugging network-related code and newtork protocol implemtations, and security engineers for analyzing the network traffic for security purposes. In general, we all can use these tools to understand how machines actually communicate with each other, i.e., to understand the internals of the network protocols.

There exists many tools for network packet inspection. Under the hood, all these tools use pcap library on linux and winpcap on windows but they differ in the way users can interface with them and the featueres they provide. For example, Wireshark is a powerful sniffer which can decode lot of protocols. Wireshark filter syntax lets you capture only the packets you are interested in. It provides a nice GUI. There also exists a command line version of wireshark, called tshark. Depending on one's needs, abilities, and familiarity, one may sometimes find tshark more handy than wireshark or vice-versa.

tcpdump is another tool, that comes pre-installed with almost all unix distributions, to capture the live traffic. However, tcpdump has limited protocol decoding features, compared to tshark and wireshark. Most of the times, as a network packet sniffer and decoder, the best bet is to use either tshark or wireshark.

In this section, we introduce you with tshark. The capture and display filters, we are going to use in tshark, can also be used in wireshark.

6.1 TSHARK

tshark lets you capture packet data from a live network, or read packets from a previously saved capture file. The captured packets are decoded by tshark and then, can either be printed to the standard output or written to a file. TShark's native capture file format is pcap format, which is also the format used by wireshark and tcpdump.

6.1.1 A SHORT TUTORIAL ON TSHARK

To capture all the traffic passing through `eth0` interface and save it in `captured_packets.pcap` file, the following command can be used:

```
# tshark -i eth0 -w captured_packets.pcap
```

where `-i` should be followed by the name of the interface and `-w` with the name of the file for captured data.

Now, using a web browser, visit few web pages like facebook.com or cnn.com. Once you're done, stop the packet capture by pressing Ctrl + C.

To read the packets captured in `captured_packets.pcap` file, use the `-r` option. Following should read all the packets captured in the `captured_packets.pcap` file:

```
# tshark -r captured_packets.pcap
```

If you want only http request packets to be displayed, please do:

```
# tshark -r captured_packets.pcap -Y http.request
```

where `-Y` option lets you specify display filters (using the same syntax as in Wireshark).

Now, let's display the hosts you connected through http. To specify that, you need to use `-T` option to specify that we want to extract fields and `-e` option to specify the field you want to be displayed. Therefore, the whole commands becomes:

```
# tshark -r capture.pcp -Y http.request -T fields -e http.host
```

If you want to check whole list of available options in tshark, you can do:

```
# tshark -help
```

or the help page can be accessed through web with this link

`https://www.wireshark.org/docs/man-pages/tshark.html`

The capture and display filters used in tshark are the same as in Wireshark and can be accessed with below links.

Capture Filters: `https://wiki.wireshark.org/CaptureFilters`

Display Filters: `https://wiki.wireshark.org/DisplayFilters`

6.1.2 EXERCISE 1

Every computer on Internet is assigned an address, called ip address. To send a packet to a remote machine, you need its ip address. While you use your computer, you want to know how your computer finds the ip addresses of other machines. For example, when you type `facebook.com` in your browser, before you connect to the facebook server, your computer needs to find the ip address of the facebook and therefore, your machine contacts the local DNS server, whose ip address your machine knows. The DNS server responds to your request and sends the ip address of the requested machine in its response.

In this exercise, your goal is to use tshark and capture only the packets that are related to the interaction of your machine with the DNS server. You should display the DNS server(s) your machine talks to, the request it makes, and finally, the response it receives from the DNS server. Once you come up with the tshark command, run the command in the terminal, and then, visit some web pages you have never visited in the recent past. In the terminal, your tshark command should display the dns server contacted, the server for which the DNS query is made, and the response from the DNS server, i.e., the ip address of the requested server, as you browse the web.

Hint: Caputre only DNS response packets from the DNS server as it will contain both the request query and the response.

Q22/ Please write below all the tshark commands (and their corresponding results) you tried in the order you typed in, even if you did not succeed. We are interested in your thought process, how did you proceed, the commands you tried, and how close you finally could come to the solution even if you did not succeed.

Solution. *This tshark command should print what is required but tshark says dns.resp.addr is not valid (even if this field is present on wireshark's display filter list).*

```
tshark -i en0 -f "src port 53 && dst host 192.168.8.103" -n -T fields
-e ip.dst -e dns.qry.name -e dns.resp.addr
```

Another possible command:

```
tshark -i en5 -Y 'dns && dns.flags.response==1'
```

6.1.3 EXERCISE 2

Alice is soon going to have her holidays. She is searching for holiday offers on the web. She finds a very interesting and cheap offer at a website and therefore, she hurries up to book it. She enters all her details in a html form, including her name, date of birth, phone numbers, email addresses, home address, and registers for this offer. After registration, when she wants to pay for this offer, she realizes that her connection to this website (until now) is not encrypted. So she stops the online payment.

The pcap file, named alice.pcap, stores all the above-mentioned activities of Alice, captured by tshark at her network interface. Now, your job is find the packet in the pcap file that contains all her information. You should use tshark command to get hold of all her details she typed in for reserving this trip.

You are not allowed to use Wireshark.

Hint: The details are filled by Alice in a html form. Therefore, an http post request body should contain her details.

Q23/ Please write below all the tshark commands you tried in the order you typed in, even if you did not succeed to get her details. We are interested in your thought process, how did you proceed, the commands you tried, and how close you finally could come to the solution even if you did not succeed. ***Solution.*** *This tshark command prints Alice's details.*

```
tshark -r alice.pcap -Y http.request.method==POST -Tfields -e
http.file_data | grep Alice
```

where -Y is for display filter options, -T is for specifying how you want to see the headers, and -e is for what field you want to get displayed. As the http response body should contain the name 'Alice', among all http post bodies, we are only interested in the body that contains the word 'Alice' and therefore, the grep command in the end.

Another possible command:

```
tshark -r alice.pcap -Y 'http.request.method == POST' -T fields -e text
```

6.1.4 EXERCISE 3

Alice uses telnet to log into her remote machine. As you might probably already know that telnet traffic is not encrypted and therefore, Alice's telnet communication can be read at any intermediate machines. While

Alice connects to her remote machine, her internet connection interface packets are captured using tshark in a file called telnet.pcap.

Now you job is to find the password of Alice. Her login to the remote machine is `testuser`. So you need to find the password for the user `testuser`.

Hint: The captured telnet session is in raw (per-character) mode. You will see each character of password in a seperate line.

Q24/ Please write below all the tshark commands in the order you tried, even if you did not succeed to get her password. We are interested in your thought process, how did you proceed, the commands you tried, and how close you finally could come to the solution even if you did not succeed. ***Solution.*** *This tshark command prints testuser password.*

```
tshark -r telnet.pcap -Y telnet -Tfields -e telnet.data
```