Lab VI
Capturing and monitoring the network traffic

1. Goals

- To gain general knowledge about the network analyzers and to understand their utility
- To learn how to use network traffic analyzer tools (Wireshark)
- To understand through this tool the layered architecture of TCP/IP
- To analyze the main protocols from different layers of TCP/IP: UDP, TCP, IP, Ethernet etc.

2. Network traffic analyzers

2.1 General overview

Network traffic analyzers are software tools which are able to capture and analyze the traffic transmitted or received over a network on which the computer is attached. These tools can serve to learning purposes in the "network protocols world", but they can also be reliable instruments for enhanced networks security. Most of these tools are freeware and they can be downloaded from the Internet. Some of the well known traffic analyzers (sometimes referred to as "packet sniffer") are: Ethereal (former name of Wireshark), tcpdump (command-line utility), Microsoft Network Monitor (supplementary part of Windows operating systems), ettercap (which is able to "capture" and decode passwords for a lot of TCP/IP application protocols and to conduct "man-in-the-middle" attacks), Wireshark etc.

On wired broadcast LANs, depending on the network structure, one can capture traffic on all or just parts of the traffic from a single machine within the network; however, there are some methods to avoid traffic narrowing by switches to gain access to traffic from other systems on the network (e.g. ARP spoofing). For network monitoring purposes it may also be desirable to monitor all data packets in a LAN by using a network switch with a so-called *monitoring port*, whose purpose is to mirror all packets passing through all ports of the switch.

Thus, when systems (computers) are connected to a switch port rather than a hub, the analyzer will be unable to read the data due to the intrinsic nature of switched networks. On wired broadcast and wireless LANs, in order to capture traffic other than unicast traffic sent to the machine running the sniffer software, multicast traffic sent to a multicast group to which that machine is listening, and broadcast traffic, the network adapter being used to capture the traffic must be put into promiscuous mode; some sniffers support this, others don't. On wireless LANs, even if the adapter is in promiscuous mode, packets not for the service set for which the adapter is configured will usually be ignored; in order to see those packets, the adapter must be put into monitor mode.

2.2 Capabilities

Between the capabilities of these network analyzing tools, we can mention:

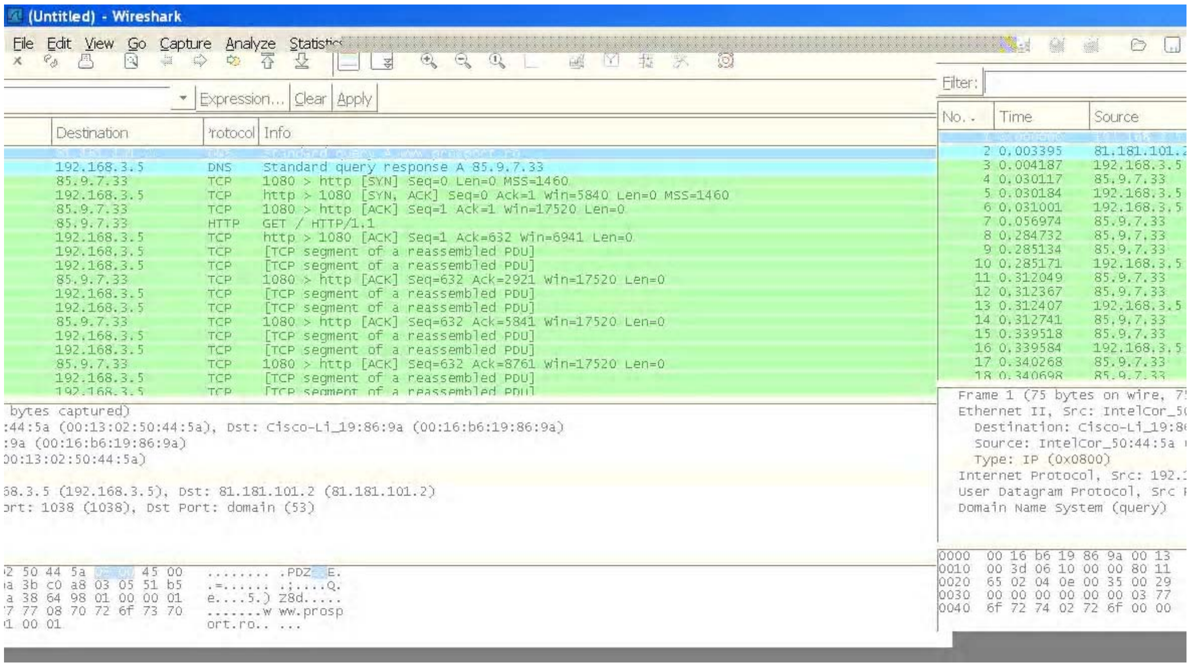
- Analyze network problems.
- Detect network intrusion attempts.
- Monitor network usage.
- Gather and report network statistics.
- Filter suspect content from network traffic.
- Spy on other network users and collect sensitive information such as passwords (depending on any content encryption methods which may be in use)
- Debug network protocol implementations.

Some practical examples related to the above-mentioned features could be:

- A packet analyzer for a token ring network could detect that the token has been lost or the presence of too many tokens (verifying the protocol).
- A packet analyzer could detect that messages are being sent to a network adapter; if the network adapter did not report receiving the messages then this would localize the failure to the adapter.
- A packet analyzer could collect statistics on the amount of traffic (number of messages) from a process detecting the need for more bandwidth or a better method.
- A packet analyzer could be used to diagnose operating system connectivity issues like web,ftp,sql,active directory,etc.
- A packet analyzer could be used to analyze data sent to and from secure systems in order to understand and circumvent security measures, for the purposes of penetration testing or illegal activities.
- A packet analyzer can passively capture data going between a web visitor and the web servers, decode it at the HTTP and HTML level and create web log files as a substitute for server logs and page tagging for web analytics.

3. Wireshark (Ethereal) overview

Wireshark (whose older name is Ethereal) is a free software package which can be used to interactively dump and analyze the network traffic. This software tool provides a simple graphical user interface to work with. Like other protocol analyzers, **Wireshark**'s main window shows 3 views of a packet. First, a summary line describes what the packet is. A packet details display is shown, allowing navigating to exact protocol or field that you are interested in. Finally, a hex dump shows exactly what the packet looks like when it goes over the wire. A snapshot of Wireshark's main window can be seen below:


(Untitled) - Wireshark
File Edit View Go Capture Analyze Statistics
Expression... Clear Apply
Filter:

```
No. .   Time        Source
      2 0.003395    81.181.101.2
      3 0.004187    192.168.3.5
      4 0.030117    85.9.7.33
      5 0.030184    192.168.3.5
      6 0.031001    192.168.3.5
      7 0.056974    85.9.7.33
      8 0.284732    85.9.7.33
      9 0.285134    85.9.7.33
     10 0.285171    192.168.3.5
     11 0.312049    85.9.7.33
     12 0.312367    85.9.7.33
     13 0.312407    192.168.3.5
     14 0.312741    85.9.7.33
     15 0.339518    85.9.7.33
     16 0.339584    192.168.3.5
     17 0.340268    85.9.7.33
     18 0.340698    85.9.7.33
```

```
Destination       'rotocol Info
[illegible]       DNS      Standard query A www.prosport.ro
192.168.3.5       DNS      Standard query response A 85.9.7.33
85.9.7.33         TCP      1080 > http [SYN] Seq=0 Len=0 MSS=1460
192.168.3.5       TCP      http > 1080 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1460
85.9.7.33         TCP      1080 > http [ACK] Seq=1 Ack=1 Win=17520 Len=0
85.9.7.33         HTTP     GET / HTTP/1.1
192.168.3.5       TCP      http > 1080 [ACK] Seq=1 Ack=632 Win=6941 Len=0
192.168.3.5       TCP      [TCP segment of a reassembled PDU]
192.168.3.5       TCP      [TCP segment of a reassembled PDU]
85.9.7.33         TCP      1080 > http [ACK] Seq=632 Ack=2921 Win=17520 Len=0
192.168.3.5       TCP      [TCP segment of a reassembled PDU]
192.168.3.5       TCP      [TCP segment of a reassembled PDU]
85.9.7.33         TCP      1080 > http [ACK] Seq=632 Ack=5841 Win=17520 Len=0
192.168.3.5       TCP      [TCP segment of a reassembled PDU]
192.168.3.5       TCP      [TCP segment of a reassembled PDU]
85.9.7.33         TCP      1080 > http [ACK] Seq=632 Ack=8761 Win=17520 Len=0
192.168.3.5       TCP      [TCP segment of a reassembled PDU]
192.168.3.5       TCP      [TCP segment of a reassembled PDU]
```

```
Frame 1 (75 bytes on wire, 75 bytes captured)
Ethernet II, Src: IntelCor_50:44:5a (00:13:02:50:44:5a), Dst: Cisco-Li_19:86:9a (00:16:b6:19:86:9a)
  Destination: Cisco-Li_19:86:9a (00:16:b6:19:86:9a)
  Source: IntelCor_50:44:5a (00:13:02:50:44:5a)
  Type: IP (0x0800)
Internet Protocol, Src: 192.168.3.5 (192.168.3.5), Dst: 81.181.101.2 (81.181.101.2)
User Datagram Protocol, Src Port: 1038 (1038), Dst Port: domain (53)
Domain Name System (query)
```

```
0000  00 16 b6 19 86 9a 00 13  02 50 44 5a 08 00 45 00   ........ .PDZ..E.
0010  00 3d 06 10 00 00 80 11  9a 3b c0 a8 03 05 51 b5   .=...... .;....Q.
0020  65 02 04 0e 00 35 00 29  5a 38 64 98 01 00 00 01   e....5.) Z8d.....
0030  00 00 00 00 00 00 03 77  77 77 08 70 72 6f 73 70   .......w ww.prosp
0040  6f 72 74 02 72 6f 00 00  01 00 01                  ort.ro.. ...
```

Fig. 1: Snapshot of Wireshark GUI

The frames are captured by means of a rich collection of capture filters. These filters are either pre-defined, or they can be defined by the user, who can choose what protocols to be displayed during the capture. Furthermore, the packets flowing through the transmission environment during a TCP/IP "conversation" can be displayed in an ASCII "user-friendly" format.

4. Practical work

The students will have to follow the steps below:

1. Check the network interface card (NIC) whose traffic will be captured and analyzed using Wireshark

The menu *Capture-Interface-Details* should be employed. The students must check the capabilities of the NIC which is used for packet capturing purposes. At the end of this step, the students should be able to answer to the following questions:

a. What media type is supported by the NIC?
b. Who is the vendor of the NIC?
c. Which is the MAC address of the network card?
d. What is the size of the packets supported by this interface?
e. What is the link speed?

2. Define the capture options and start the capture.

The menu used is *Capture-Options*. The Ethernet card must be selected as the capture interface and then the pre-defined capture filters should be checked. Before starting the capture using the right-corner start button, the students will write down the answer to the following questions:

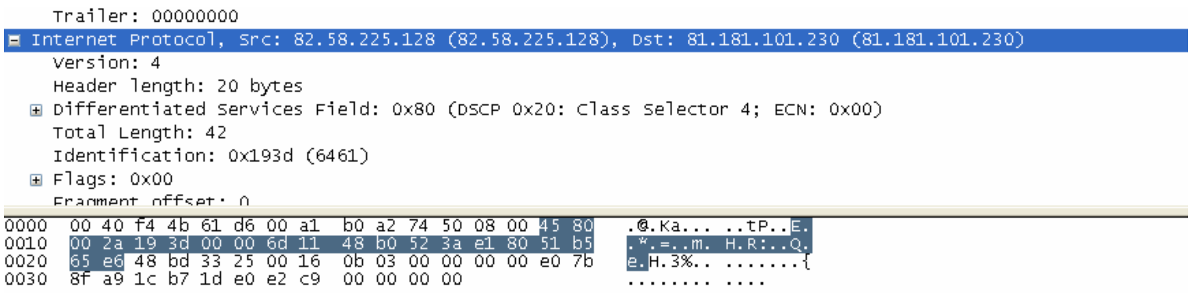
a. How many interfaces can be selected from, for capturing purposes and what are the properties of these interfaces?
b. List five of the capture filters which are pre-defined and try to explain their meaning. These filters can be browsed from the menu Capture, the button named Filter.

3. Start a capture which will be automatically stopped after 20 seconds. The capture duration can be set from the same menu (*Capture-Options*). For this first capture, no filtering option will be used. Try to identify the TCP and UDP packets and answer the following questions:

a. What are the protocols analyzed in the lower part of the window (when you select a packet in the upper part using the arrows or the mouse)?

b. What is the length of the packets? Is this length the same for all the upper-layer protocols (TCP, UDP etc)?

c. Browse the protocols in the downward direction. Try to identify and to write down the header lengths for all the protocols involved and to establish the position that each header occupies in the captured frame. An example is given below for the IP header:



```
    Trailer: 00000000
⊟ Internet Protocol, Src: 82.58.225.128 (82.58.225.128), Dst: 81.181.101.230 (81.181.101.230)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x80 (DSCP 0x20: Class Selector 4; ECN: 0x00)
    Total Length: 42
    Identification: 0x193d (6461)
  ⊞ Flags: 0x00
    Fragment offset: 0
0000  00 40 f4 4b 61 d6 00 a1  b0 a2 74 50 08 00 45 80   .@.Ka... ..tP..E.
0010  00 2a 19 3d 00 00 6d 11  48 b0 52 3a e1 80 51 b5   .*.=..m. H.R:..Q.
0020  65 e6 48 bd 33 25 00 16  0b 03 00 00 00 00 e0 7b   e.H.3%.. .......{
0030  8f a9 1c b7 1d e0 e2 c9  00 00 00 00               ........ ....
```

Fig. 2: Identifying header positions with Wireshark.

c. What is the payload length (data filed)? Compare it with the total length of the packet.

4. Check some statistical information about the packets transferred during the capture. Thus, browsing the submenus of the menu Statistics (Summary, Conversation list), try to find out and to write down: the total number of captured packets, the average throughput, the average packet size. What are the protocols from the conversation list which were not used during the capture?

5. Build a new capture filter which is enabled to capture only HTTP/IP packets (for both TCP and UDP employed as transport layer protocols), for a duration of 30sec. Start the capture and then try to open a web page. When the capture stops, try to identify and to write down the following information: the protocol hierarchy, the conversations (IP addresses involved in packet transmission), the source and destination port numbers for TCP and UDP packets. What protocol between the two transport layer protocols is the most used?