LAB WORK 06. WIRESHARK. NETWORK TRAFFIC CAPTURE AND ANALYZE

1. LAB WORK GUIDELINES

Disclaimer. This document was created on the basis of the textbook «Data Communication and Networking», 5th.Edition, 2011, -1269 pp., by Behrouz A. Forouzan.

We have created lab assignments for several layers of TCP/IP. We have only theoretical lab assignments for physical layer. We cannot use a standard packet sniffer, such as Wireshark, to capture bits. We cannot sniff management packets because we have normally no permission to act as a manager. In this document, we give an introduction to packet sniffing, introduce the Wireshark software, talk about the lab reports, and finally tell you what to do in this lab assignment.

1.1 PACKET SNIFFING

The purpose of lab assignments is to show how we can get a deeper understanding of the networking concepts by capturing and analysing the packets sent and received from our host. One way to do so is to use a packet sniffer. A packet sniffer is a piece of soft- ware that should be running in parallel with the application whose packets needed to be analysed. However, before running a packet sniffer, we need to interpret the term packet. As we discussed in Chapter 1 of the textbook, communication via the Internet is done using a five-layer suite. We can analyse the packets at four layers: application, transport, network, and data-link. There is no packet exchange at the physical layer; communication at this layer is done using bits.

Although it is useful to analyse the packets in each of the four upper layers of the TCP/IP protocol suite, should a packet sniffer software be designed to capture packets at each of these layers? The answer to this question can be found in Figure 2.8 in the textbook (encapsulation-decapsulation). In an outgoing situation, a packet created at any upper-layer is encapsulated in a frame (at the data-link layer); in an incoming situation, a packet intended for any layer is decapsulated from the received frame. This means we need to capture only outgoing or incoming frames; a packet-sniffer software can extract the packets at any layer desired to be analysed from these frames. For this reason, a packet-sniffer software is normally having two components: a packet-capturer and a packet-analyser. The packet-capturer captures a copy of all outgoing and incoming frames (at the data-link layer) and passes them to the packet-analyser. The packet- analyser can then extract different headers and the ultimate message for analysis.

Before we continue with our discussion, we need to make a point clear. Although Figure 2.8 in the textbook shows that the encapsulation starts or decapsulation ends at the application layer, a packet in the Internet can belong to any layer above the

data- link layer. As we will see in future chapters, protocols at the transport or network layer protocols also need to exchange packets. All of these packets are encapsulated in or decapsulated from the frames. A packet sniffer needs to capture all incoming and out- going frames and show the headers of all protocols used for communication. The source or the sink of a packet is not necessarily the application layer. Figure 1.1 shows two examples.



Figure 1.1 Role of frame capturing and packet analysing in a packet-sniffer

In Example I, an outgoing frames is captured. The source of the frame is the HTTP protocol at the application layer (discussed in Chapter 26 of the textbook). A copy of the frame is passed to the analyser. The analyser extracts the general information in the frame (the box marked frame), headers 2, 3, and 4, and the HTTP message for analysis. In Example II, an incoming frame is captured. The sink (final destination) is the ARP protocol at the network layer (discussed in Chapter 9 in the textbook). A copy of the frame is passed to the analyser. The analyser extracts the general information in the header (the box marked frame), header 2 and the ARP message for analysis.

1.2 WIRESHARK



In this and other lab assignments, we use a packet-sniffer called Wireshark. Wireshark (formerly known as ETHEREAL) is a free packet sniffer/analyser which is available for both UNIX-like (Unix, Linux, Mac OS X, BSD, and Solaris) and Windows operating systems. It captures packets from a network interface and displays them with detailed protocol information. Wireshark, however, is a passive analyser. It only captures packets without manipulate them; it neither sends packets to the network nor does other active operations. Wireshark is not an intrusion-detection tool either. It does not give warning about any network intrusion. It, nevertheless, can help network administrators to figure out what is going on inside a network and to troubleshoot network problems. In addition of being an indispensable tool for network administrators, Wire- shark is a valuable tool for protocol developers, who may use it to debug protocol implementations. It is also a great educational tool for computer-network students who can use it to see details of protocol operations in real time.

1.2.1 Main Window

The Wireshark main window is similar to other GUI tools as shown in Figure 1.2. The Wireshark window is made of seven sections: title bar, menu bar, filter bar, packet list pane, packet detail pane, packet byte pane, and status bar. We briefly discuss the functionality of each section below:

Title Bar

The title bar (like the one in any GUI) shows the title of the window, the closing, maximizing, and minimizing icons.

Menu Bar

The menu bar is made of several pull-down menus and tool bars used in most GUIs. We will use some of these menus in our lab assignments. We can use the File menu to perform some actions on the file itself such as saving and printing. The Capture menu is used to start and capturing frames. The View menu is useful to show or hide some of the sections in the window.



Figure 1.2 Main window of Wireshark

Filter Bar

The filter bar allows us to display packet we are interested in while hiding the rest. As we see later in this document, when we start capturing frames, Wireshark captures and analyse any outgoing and incoming frame no matter what is the source or sink protocol. Sometimes, this is not what we want. We may want to limit the analysis to a specific source or sink protocol. For example, we may want to analyse only packets sent or receive by the HTTP protocol at the application layer or the ARP protocol at the network layer. This is called filtering in the parlance of packet sniffing. After packets have been captured, we can type the name of the protocol in lowercase and click Apply.

Packet List Pane

The packet list pane displays a one-line summary for each captured packet (actually frame). The summary includes the packet (frame) number (added by the Wireshark and not part of the packet), the time when the packet was captured, the source and destination IP addresses of the packet (at the network layer), the packet source or sink proto- col, and the additional information about the packet contents. In other words, this pane shows the captured frames that will be passed for analysing to the packet analyser. For colouring packets use View \rightarrow Colorize Packet List.

Packet Detail Pane

The packet detail pane shows the detailed analysis for each frame (Figure 1.3). The information is limited to one frame, which means we need to select one of the frames in the packet list pane for analysis. This can be done by clicking on the corresponding frame in the packet list pane. Clicking on any frame in the packet list pane highlights the frame and shows the details of the frame in the packet details pane. Information exhibited in this pane for each frame is made of a tree structure. However, each top branch of the tree is shown as one line as it is common in GUI trees.



Figure 1.3 Packet detail pane

We can expand the branch (to see sub branches) by clicking on the plus box at the leftmost part of the line, which changes the plus sign to a minus sign; the branch can be collapsed again, which changes the minus sign to the plus sign. Note that the analyser first shows a general information at the data-link layer (frame). It then displays the information contained in each header from the data-link layer (H2) up to the source or sink protocol. It finally shows the whole message at the source or sink layer. Figure 1.3 shows an example of a packet details pane when the frame is expanded. It shows some general information and names of all protocols used in the frame (intermediate and source or sink).

Packet byte pane

The packet byte pane shows the entire current frame (selected in the packet list pane) in hex dump format (hexadecimal view of data) and ASCII format. The number in the left field shows the offset in the packet data; the hex dump of the packet is shown in the middle field; the corresponding ASCII characters are shown in the right field. If we need the byte (or ASCII equivalent) of any line in the packet detail pane, we can click on the line in the packet detail pane and the byte contents will be highlighted. Figure 1.4 shows an example of a packet byte pane. It shows all the bytes in the frame, but we can select the bytes in any protocol header by highlighting it in the packet detail pane section.

0020	7d	10	06	11	00	50	95	03	e9	ac	5e	b9	db	0C	50	18	}P^P.	~
0030	11	73	01	Te	00	00	47	45	54	20	21	66	6d	61	73	68	.sGE T /fmosh	
0040	61	72	72	61	66	21	45	6e	67	69	6e	65	65	72	69	6e	arraf/En gineerin	100
0050	67	21	45	4e	47	52	25	32	30	32	31	37	21	65	6e	67	g/ENGR%2 0217/eng	
0060	72	5 Ť	32	31	37	2e	68	74	6d	20	48	54	54	50	21	31	r_217.ht m HTTP/1	
0070	2e	31	od	0a	48	61	73	74	3a	20	66	61	63	75	6C	74	.1Host : facult	
0080	79	2e	72	69	6f	68	6Ť	6e	64	6f	2e	65	64	75	od	0a	y.riohon do.edu	
0090	55	73	65	72	2 d	41	67	65	6e	74	3a	20	4d	61	7a	69	User-Age nt: Mozi	
00a0	6C	6C	61	21	35	2e	30	20	28	57	69	6e	64	61	77	73	11a/5.0 (Windows	
00b0	зb	20	55	3b	20	57	69	6e	64	6f	77	73	20	4e	54	20	; U; Win dows NT	
00c0	35	2e	31	3b	20	65	6e	2d	55	53	3b	20	72	76	3a	31	5.1; en- US; rv:1	
00d0	2e	39	2e	32	2e	38	29	20	47	65	63	6b	6f	21	32	30	.9.2.8) Gecko/20	-
00e0	31	30	30	37	32	32	20	42	54	52	53	32	39	33	31	33	100722 B TRS29313	
00f0	20	46	69	72	65	66	6Ť	78	21	33	2e	36	2e	38	20	47	Firefox /3.6.8 G	
0100	54	42	37	2e	31	20	28	20	2e	4e	45	54	20	43	4C	52	TB7.1 (.NET CLR	
0110	20	33	2e	35	2e	33	30	37	32	39	29	od	0a	41	63	63	3.5.307 29)Acc	
0120	65	70	74	3a	20	74	65	78	74	2f	68	74	6d	6C	20	61	ept: tex t/html,a	
0130	70	70	6C	69	63	61	74	69	61	6e	21	78	68	74	6d	6C	pplicati on/xhtml	
0140	2b	78	6d	6C	2 C	61	70	70	6C	69	63	61	74	69	6Ť	6e	+xml,app lication	
0150	21	78	6d	6C	3b	71	3 d	30	2e	39	20	2a	21	2a	зb	71	/xml;q=0 .9,*/*;q	
0160	30	30	2e	38	od	oa	41	63	63	65	70	74	2d	40	61	6e	=0.8Ac cept-Lan	
0170	67	75	61	67	65	3a	20	65	6e	2 d	75	73	od	0a	41	63	guage: e n-usAc	
0180	63	65	70	74	2 d	45	6e	63	6f	64	69	6e	67	3a	20	67	cept-Enc oding: g	
0190	7a	69	70	2C	64	65	66	6C	61	74	65	0d	0a	41	63	63	zip, defl ate Acc	
01a0	65	70	74	2 d	43	68	61	72	73	65	74	3a	20	49	53	41	ept-Char set: ISO	Y
						-	-						-					-
Transn	nissio	n Co	ntrol	Pro	tocol		Pac	kets:	7 Dis	play	ed: 3	z Ma		Prof	ile: [Default		.;

Figure 1.4 Packet byte pane

Status Bar

The last section of the window (at the bottom) is the status bar which shows the current protocol, the total number of packets captured, and so on.

1.2.3 Working with Wireshark

When we work with Wireshark in this and other labs, there are some actions that we need to repeat over and over. We mention the details of some of this action to avoid rementioning them.

Start Capturing

To begin capturing, select the Capture from the pull down menu and click Options... to open the Wireshark capture dialog box.

There are several steps that you need to follow before you start capturing:

1. The network interfaces are shown in the Interface list at the Input box. Select the network interface (or use the default interface chosen by Wireshark). If the IP address in the dialog box is unknown, you must select a different interface; otherwise, the Wireshark will not capture any packet (Figure 1.5).

Int	erface	Traffic		Link-layer	Header	Promisci	Snaplen (B)	Buffer (MB)	Monitor
V	Wi-Fi: en0			Ethernet		 Image: A set of the set of the	default	2	
	Addresses: fe80::bae8:56ff:fe04	4:8a4a, 192.168	.111.4						
⊧	awdl0			Ethernet			default	2	_
	Thunderbolt Bridge: bridge0			Ethernet			default	2	_
	Thunderbolt 1: en1			Ethernet			default	2	_
	p2p0			Raw IP			default	2	_
⊧	Loopback: Io0			BSD loop	back	\checkmark	default	2	_
	Cisco remote capture: cisco			Remote of	apture dependent DLT	_	_	_	_
	Random packet generator: randpkt			Generato	or dependent DLT	_	_	_	_
	SSH remote capture: ssh			Remote of	capture dependent DLT	_	_	_	_
<u>_</u>	Enable promiscuous mode on all inte	erfaces		Remote	capture dependent DLI	_	_	 Manage Inte	
:ap	oture filter for selected interfaces: 📕	Enter a captur	re filter				× •	Comp	ile BPFs



2. It is possible to configure packet filtering using the window Capture Filters (Figure 1.6).

Name Filter Ethernet address 00:00:5e:00:53:00 ether host 00:00:5e:00:53:00 Ethernet type 0x0806 (ARP) ether proto 0x0806 No Broadcast and no Multicast not broadcast and not multicast No ARP not arp IPv4 only ip IPv4 address 192.0.2.1 host 192.0.2.1 IPv6 only ip6 IPv6 address 2001:db8::1 host 2001:db8::1 IPx only ipx TCP only udp TCP or UDP port 80 (HTTP) port 80 HTTP TCP port (80) tcp port http No ARP and no DNS not arp and port not 53 Non-HTTP and non-SMTP to/from www.wireshark.org not port 80 and not port 25 and host www.wireshark.org	•	🔘 🔵 Wireshark ·	Capture Filters
Name Filter Ethernet address 00:00:5e:00:53:00 ether host 00:00:5e:00:53:00 Ethernet type 0x0806 (ARP) ether proto 0x0806 No Broadcast and no Multicast not broadcast and not multicast No ARP not arp IPv4 only ip IPv4 address 192.0.2.1 host 192.0.2.1 IPv6 only ip6 IPv6 address 2001:db8::1 host 2001:db8::1 IPX only ipx TCP only udp TCP or UDP port 80 (HTTP) port 80 HTTP TCP port (80) tcp port http No ARP and no DNS not arp and port not 53 Non-HTTP and non-SMTP to/from www.wireshark.org not port 80 and not port 25 and host www.wireshark.org			
Ethernet address 00:00:5e:00:53:00 ether host 00:00:5e:00:53:00 Ethernet type 0x0806 (ARP) ether proto 0x0806 No Broadcast and no Multicast not broadcast and not multicast No ARP not arp IPv4 only ip IPv4 address 192.0.2.1 host 192.0.2.1 IPv6 only ip6 IPv6 address 2001:db8::1 host 2001:db8::1 IPX only ipx TCP only tcp UDP only udp TCP or UDP port 80 (HTTP) port 80 HTTP TCP port (80) tcp port http No ARP and no DNS not arp and port not 53 Non-HTTP and non-SMTP to/from www.wireshark.org not port 80 and not port 25 and host www.wireshark.org		Name	Filter
Ethernet type 0x0806 (ARP) ether proto 0x0806 No Broadcast and no Multicast not broadcast and not multicast No ARP not arp IPv4 only ip IPv4 address 192.0.2.1 host 192.0.2.1 IPv6 only ip6 IPv6 address 2001:db8::1 host 2001:db8::1 IPX only ipx TCP only tcp UDP only udp TCP or UDP port 80 (HTTP) port 80 HTTP TCP port (80) tcp port http No ARP and no DNS not arp and port not 53 Non-HTTP and non-SMTP to/from www.wireshark.org not port 80 and not port 25 and host www.wireshark.org		Ethernet address 00:00:5e:00:53:00	ether host 00:00:5e:00:53:00
No Broadcast and no Multicast not broadcast and not multicast No ARP not arp IPv4 only ip IPv4 address 192.0.2.1 host 192.0.2.1 IPv6 only ip6 IPv6 address 2001:db8::1 host 2001:db8::1 IPX only ipx TCP only udp TCP or UDP port 80 (HTTP) port 80 HTTP TCP port (80) tcp port http No ARP and no DNS not arp and port not 53 Non-HTTP and non-SMTP to/from www.wireshark.org not port 80 and not port 25 and host www.wireshark.org		Ethernet type 0x0806 (ARP)	ether proto 0x0806
No ARP not arp IPv4 only ip IPv4 address 192.0.2.1 host 192.0.2.1 IPv6 only ip6 IPv6 address 2001:db8::1 host 2001:db8::1 IPX only ipx TCP only tcp UDP only udp TCP or UDP port 80 (HTTP) port 80 HTTP TCP port (80) tcp port http No ARP and no DNS not arp and port not 53 Non-HTTP and non-SMTP to/from www.wireshark.org not port 80 and not port 25 and host www.wireshark.org		No Broadcast and no Multicast	not broadcast and not multicast
IPv4 only ip IPv4 address 192.0.2.1 host 192.0.2.1 IPv6 only ip6 IPv6 address 2001:db8::1 host 2001:db8::1 IPX only ipx TCP only tcp UDP only udp TCP or UDP port 80 (HTTP) port 80 HTTP TCP port (80) tcp port http No ARP and no DNS not arp and port not 53 Non-HTTP and non-SMTP to/from www.wireshark.org not port 80 and not port 25 and host www.wireshark.org		No ARP	not arp
IPv4 address 192.0.2.1 host 192.0.2.1 IPv6 only ip6 IPv6 address 2001:db8::1 host 2001:db8::1 IPX only ipx TCP only tcp UDP only udp TCP or UDP port 80 (HTTP) port 80 HTTP TCP port (80) tcp port http No ARP and no DNS not arp and port not 53 Non-HTTP and non-SMTP to/from www.wireshark.org not port 80 and not port 25 and host www.wireshark.org		IPv4 only	ip
IPv6 only ip6 IPv6 address 2001:db8::1 host 2001:db8::1 IPX only ipx TCP only tcp UDP only udp TCP or UDP port 80 (HTTP) port 80 HTTP TCP port (80) tcp port http No ARP and no DNS not arp and port not 53 Non-HTTP and non-SMTP to/from www.wireshark.org not port 80 and not port 25 and host www.wireshark.org		IPv4 address 192.0.2.1	host 192.0.2.1
IPv6 address 2001:db8::1 host 2001:db8::1 IPX only ipx TCP only tcp UDP only udp TCP or UDP port 80 (HTTP) port 80 HTTP TCP port (80) tcp port http No ARP and no DNS not arp and port not 53 Non-HTTP and non-SMTP to/from www.wireshark.org not port 80 and not port 25 and host www.wireshark.org		IPv6 only	ip6
IPX only ipx TCP only tcp UDP only udp TCP or UDP port 80 (HTTP) port 80 HTTP TCP port (80) tcp port http No ARP and no DNS not arp and port not 53 Non-HTTP and non-SMTP to/from www.wireshark.org not port 80 and not port 25 and host www.wireshark.org		IPv6 address 2001:db8::1	host 2001:db8::1
TCP only tcp UDP only udp TCP or UDP port 80 (HTTP) port 80 HTTP TCP port (80) tcp port http No ARP and no DNS not arp and port not 53 Non-HTTP and non-SMTP to/from www.wireshark.org not port 80 and not port 25 and host www.wireshark.org + - Pa		IPX only	ipx
UDP only udp TCP or UDP port 80 (HTTP) port 80 HTTP TCP port (80) tcp port http No ARP and no DNS not arp and port not 53 Non-HTTP and non-SMTP to/from www.wireshark.org not port 80 and not port 25 and host www.wireshark.org + - Pb		TCP only	tcp
TCP or UDP port 80 (HTTP) port 80 HTTP TCP port (80) tcp port http No ARP and no DNS not arp and port not 53 Non-HTTP and non-SMTP to/from www.wireshark.org not port 80 and not port 25 and host www.wireshark.org + - Pa Help Capacity		UDP only	udp
HTTP TCP port (80) tcp port http No ARP and no DNS not arp and port not 53 Non-HTTP and non-SMTP to/from www.wireshark.org not port 80 and not port 25 and host www.wireshark.org + - Holp Capacity		TCP or UDP port 80 (HTTP)	port 80
No ARP and no DNS not arp and port not 53 Non-HTTP and non-SMTP to/from www.wireshark.org not port 80 and not port 25 and host www.wireshark.org + - Pb Help Capacity		HTTP TCP port (80)	tcp port http
Non-HTTP and non-SMTP to/from www.wireshark.org not port 80 and not port 25 and host www.wireshark.org		No ARP and no DNS	not arp and port not 53
		Non-HTTP and non-SMTP to/from www.wireshark.org	not port 80 and not port 25 and host www.wireshark.org
	l		
		+ – Pa	
	l		
Cancel UK		Help	Cancel



3. You normally will use the default values in the capture options dialog box, but there are some options that you may need to override the default (Figure 1.7).

	🚄 Wireshark · Capture Interfaces		
	Input Output Ontions	🚄 Wireshark · Capture	Interfaces
Capture to a permanent file	input Output Options	Input Output	Options
File: Leave blank to use a ten	nporary file	Display Options	Name Resolution
		🔽 Update list of packets in real-time	🗹 Resolve MAC Addresses
Output format:	ncan	Automatically scroll during live capture	Resolve network names
	li star	Show extra capture information dialog	Resolve transport names
Create a new file automatica	ily atter		
1 kilobytes	٥	Stop capture automatically after	
		1 C packets	
Seconds	<u> </u>	□ 1 C files	
	10	□ 1 🗘 kilobytes ᅌ	
Use a ring buffer with 2	ĴĴ files	1 C seconds C	

Figure 1.7 Capture Output and Options window

After the above three steps, click Start. Wireshark starts to captures packets that are exchanged between your computer and the network. If, after a minute, Wireshark does not capture any packet, there must be a problem; check for possible reason and troubleshooting.

Stop Capturing

Whenever you feel you have captured all the packets (frames) that you need to do your lab report, you can stop capturing. To do so, you need to use the Capture pull-down menu and click Stop. Wireshark stops capturing the frames.

Saving the Captured File

After you have stopped capturing, you may want to save the captured file for future use.

1.2.4 Incoming and Outgoing Frames

When we see the list of the captured frames, we often wonder which frames are the incoming and which ones are outgoing. This can be found by looking at the frame in packet list pane. The packet list pane shows the source and destination addresses of the frame (generated and inserted at the network layer). If the source address is the address of the host you are working with (shown on the Capture window when you start capturing), the frame is the outgoing frame; if the destination address is the address of your host, the frame is the incoming frame.

1.2.5 Analyze and Statistics

In addition, Wireshark has several convenient and useful functions. For example:

1. View \rightarrow Coloring Rules (see Figure 1.8)

am	е	Filter
~	Bad TCP	tcp.analysis.flags && !tcp.analys
<u>⁄</u>	HSRP State Change	hsrp.state != 8 && hsrp.state !=
~	Spanning Tree Topology Change	stp.type == 0x80
<u><</u>	OSPF State Change	ospf.msg != 1
<u><</u>	ICMP errors	icmp.type eq 3 icmp.type eq 4
\checkmark	ARP	arp
\checkmark	ICMP	icmp icmpv6
✓	TCP RST	tcp.flags.reset eq 1
✓	SCTP ABORT	sctp.chunk_type eq ABORT
✓	TTL low or unexpected	(!ip.dst == 224.0.0.0/4 && ip.tt
<u><</u>	Checksum Errors	eth.fcs.status=="Bad" ip.checl
\checkmark	SMB	smb nbss nbns nbipx ipx
	НТТР	http tcp.port == 80 http2

Figure 1.8 Coloring Rules window

2. Analyze \rightarrow Expert Information (see Figure 1.9) will show a list of the main events that occurred during the capture - the opening of new sessions, not quite good protocol behaviour (repeated receipts in TCP, segment retransmissions, etc.).

● ● Wireshark · Expert Information · smallFlows								
Pad	ket 🔻	Summary	Group	Protocol	Count			
►	Error	Invalid Version	Malformed	QUIC	7			
▶	Error	Malformed Packet (Exception occurred)	Malformed	QUIC	5			
►	Error	Malformed Packet (Exception occurred)	Malformed	SSL	34			
►	Error	Malformed Packet (Exception occurred)	Malformed	RTCP	1			
►	Warning	Connection reset (RST)	Sequence	TCP	78			
►	Warning	ACKed segment that wasn't captured (common at capture start)	Sequence	TCP	151			
Þ	Warning	Previous segment(s) not captured (common at capture start)	Sequence	TCP	109			
Þ	Warning	Ignored Unknown Record	Protocol	SSL	125			
Þ	Warning	This frame is a (suspected) out-of-order segment	Sequence	TCP	54			
Þ	Warning	TCP window specified by the receiver is now completely full	Sequence	TCP	2			
7	Warning	No response seen to ICMP request	Sequence	ICMP	2			
	13186	Echo (ping) request id=0x0001, seq=3926/22031, ttl=32 (no r						
	13266	Echo (ping) request id=0x0001, seq=3929/22799, ttl=32 (no r						
Þ	Note	This frame is a (suspected) spurious retransmission	Sequence	TCP	31			
Þ	Note	This frame is a (suspected) retransmission	Sequence	TCP	119			
Þ	Note	Duplicate ACK (#1)	Sequence	TCP	117			
Þ	Note	HTTP body subdissector failed, trying heuristic subdissector	Malformed	HTTP	8			
Þ	Note	This session reuses previously negotiated keys (Session resu	Sequence	SSL	22			
Þ	Note	"Time To Live" only 1	Sequence	IPv4	20			
Þ	Note	Duplicate ACK (#2)	Sequence	TCP	25			
Þ	Note	Duplicate ACK (#3)	Sequence	TCP	1			
►	Note	The acknowledgment number field is nonzero while the ACK fl	Protocol	TCP	1			
►	Chat	GET /complete/search?client=chrome&hl=en-US&q=cr HTTP/	Sequence	HTTP	1			
•	Chat	HTTP/1 1 200 OK\r\n	Sequence	HTTP	486			
0	lisplay filter set.							
	Limit to Displ	ay Filter 🗹 Group by summary Search:			Sho	w		

Figure 1.9 Expert Information window

3. Statistics \rightarrow Capture File Properties allows you to view some statistics for the capture session in general - including the average number of packets per second and the amount of data transferred (Figure 1.10).

😑 🔵 Wireshark	· Capture File	Properties	• wireshar	k_en0_20171	010010942_f	Fqbba	ar
Details							5
Time							~
First packet: Last packet: Elapsed:	2017-10-10 (2017-10-10 (00:10:38	01:09:42 01:20:20					
Capture							
Hardware: OS: Application:	Intel(R) Core Mac OS X 10 Dumpcap (W	(TM) M-5Y 12.6, build /ireshark) 2	51 CPU @ 16G29 (Da .4.1 (v2.4.1	1.10GHz (with arwin 16.7.0) I-0-gf42a0d2	SSE4.2))	- 1	
Interfaces							
Interface en0	<u>Dropped pac</u> Unknown	<u>kets</u> <u>Captu</u> none	<u>re filter</u>	<u>Link type</u> Ethernet	<u>Packet</u> 52428	<u>size limit</u> 8 bytes	
Statistics							
Measurement Packets Time span, s Average pps Average packet si B	<u>Capture</u> 1725 638.297 2.7 ze, 300.5	<u>d</u>	<u>Display</u> 1725 (1 638.29 2.7 300.5	<u>red</u> 00.0%) 07	<u>Marked</u> 		
Bytes	517999		517999	9 (100.0%)	0		

Figure 1.10 Capture File Properties window

4. Statistics \rightarrow Protocol Hierarchy - statistics on the protocols used (Figure 1.11).

rotocol 🔻	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s
Frame	100.0	4347	100.0	1494517	6345
Ethernet	100.0	4347	4.1	60858	258
Internet Protocol Version 6	0.1	3	0.0	525	2
User Datagram Protocol	0.1	3	0.0	24	0
Multicast Domain Name System	0.1	3	0.0	381	1
Internet Protocol Version 4	99.1	4310	5.8	86200	365
User Datagram Protocol	23.2	1009	0.5	8072	34
Simple Service Discovery Protocol	20.5	892	21.3	318526	1352
QUIC (Quick UDP Internet Connections)	1.6	69	3.1	45624	193
Network Time Protocol	0.1	4	0.0	192	0
NetBIOS Name Service	0.4	19	0.1	1112	4
Multicast Domain Name System	0.1	3	0.0	381	1
Domain Name System	0.4	18	0.1	1148	4
Bootstrap Protocol	0.1	4	0.1	1200	5
Transmission Control Protocol	75.9	3301	64.9	969421	4115
SSH Protocol	38.7	1682	49.5	739264	3138
Malformed Packet	0.2	10	0.0	0	0
Secure Sockets Layer	8.2	357	8.2	122306	519
Data	0.2	8	0.0	616	2
Address Resolution Protocol	0.8	34	0.1	952	4

Figure 1.11 Protocol Hierarchy window

5. Statistics \rightarrow Conversations shows information about the participants in the communication, who sent packets and data to whom to whom and how (Figure 1.12).

			Ethernet · 5	IPv4 · 13	IPv6 · 1	TCP · 16	UDP · 79			
Address A 10.10.1.75 10.10.1.75 10.10.1.75 10.10.1.75 10.10.1.75 10.10.1.75 10.10.1.75 10.10.1.75 10.10.1.75 10.10.1.75 10.10.1.75 10.10.1.75 10.10.1.75	*	Address B 239.255.255.250 10.10.1.255 85.254.142.227 192.168.68.249 54.165.177.196 52.206.41.133 17.253.38.253 216.58.211.142 216.58.209.142 17.252.92.20 17.172.238.201 224.0.0.251 17.248.150.111	Packets 892 19 2,540 22 28 3 4 659 86 6 6 3 42	Bytes 355 k 1910 913 k 3272 8114 206 360 151 k 40 k 704 704 507 14 k	Packets A → B	892 19 1,568 11 16 2 2 363 40 4 4 3 22	Bytes A → B 355 k 1910 413 k 1419 1884 132 180 87 k 7062 466 466 507 4954	Packe	Bluetooth ✓ Ethernet FC FDDI IEEE 802.11 ✓ IPv4 ✓ IPv6 IPX JXTA MPTCP NCP RSVP SCTP ✓ TCP Token-Ring	rtes B →



6. Statistics \rightarrow IO Graphs allows to you build an almost arbitrary statistical graph of the captured data (Figure 1.13).





7. Statistics \rightarrow Packet Lengths allows to you finding a very short and very long frames (Figure 1.14).

	Count	Average	Min vai	Max vai	Rate (ms)	Percent	Burst rate	Burst start	
Packet Lengths 0-19	27483 0	808.52 -	42 -	1514 -	0.0895 0.0000	100% 0.00%	4.5500 -	92.370 -	
20-39	0	-	-	-	0.0000	0.00%	-	-	
40-79	10263	57.97	42	78	0.0334	37.34%	0.9200	92.396	
80-159	1578	125.59	81	159	0.0051	5.74%	0.6500	270.565	
160-319	653	201.31	161	319	0.0021	2.38%	0.4100	270.576	
320-639	258	438.79	323	639	0.0008	0.94%	0.1300	1.268	
640-1279	328	934.70	646	1279	0.0011	1.19%	0.0700	61.725	
1280-2559	14403	1449.43	1284	1514	0.0469	52.41%	3.1700	92.370	
2560-5119	0	-	-	-	0.0000	0.00%	-	-	
5120 and greater	0	-	-	-	0.0000	0.00%	-	-	



8. File \rightarrow Exporting Objects from HTTP, SMB, SMTP, TFTP Traffic. (Figure 1.15).

	extracting-objects-from-pcap-example-02.pcap																												
Eil	e <u>E</u> di	t ⊻i	ew	<u>G</u> 0			<u>A</u> na	alyze	<u>S</u> ta	tistics	Te	lepho	ny-	Wiro	lace	Tor	le Ho	ln			_				-	N			
	<u>O</u> per	1						С	trl+0			-	1					W	iresha	rk∙	Expo	ort · HT	ТР о	bject lis	st			+ - D	×
-	Oper	<u>R</u> ece	ent								•			Pack	ce 🕶	Host	name						Co	ntent Ty	/pe		Size	Filename	
1	<u>M</u> erg	e												185		www	paypal	acco	untslo	ginn	.myd	dns.co	m te	xt/html			15 kB	websrc	
-	Impo	rt fro	m He	ex Du	imp.						p	ort		389		www	paypal	acco	untslo	ginn	.myd	dns.cor	m ap	plication	n/x-www		175 bytes	verifycha	
	Close	9						С	trl+w		5	165	9	627		www	paypal	acco	untslo	ginn	.myd	dns.cor	n ap	plication	n/x-www		1,205 bytes	webscr?d	c
-	Save								trl+S		- 5	1022		636		www	paypal	acco	untslo	ginn	.myd	dns.cor	m tex	xt/html		-	2,146 bytes	webscr?o	c
	Save	As						C	trl+S	hift+S	2	30	9	644		www	paypal	acco	untslo	ginn	.myd	dns.cor	n te	xt/html			328 bytes	faviron.i	
-	File S	et						-			- 4	923	9	648		www	paypa	acco	untslo	ginn	.myd	dns.cor	n te	xt/html			261 bytes	my 🔨	
-	-				-1						- 4	923	9	4		_													\geq
	Ехро	rt Sp	ecifie	ed Pa	ckets	S					8	30		Text	Filte	r:													
	Ехро	rt Pa	cket	Disse	ection	าร						20		H	Help									Sa	ave All		Close	Save	-
2	Ехро	rt Pa	cket .	Byte	5			C	trl+S	hift+X				_	N	\mathbf{b}													
	Ехро	rt PD	Us to	o File	•••						S), (95	byte	ľ	ρ	tured	d (1	760 b)it:	S)			. 64 (64.5	
	Ехро	rt TLS	S Ses	ssion	Keys	5					.e	(00	9:0	8:0			:ae)), [UST:	Ne	tgea	ar_b6	:93	: 71 (20:e5	:2	a:b6:93	T1)	
	Ехро	rt Ob	jects								•	DIC	OM		10	rt:	53	.0.2	29.1										
	<u>P</u> rint							С	trl+P			HT	ſP																
	Quit							С	trl+Q			IMF																	
	_						_		-			SM	в																
00	000	20	e5	2a	b6	93	f1	00	08	02	1	TFT	Ρ		0	45	00		*		· · G	}···E							-
00	010	00 1 d	51	07	21 ch	00	00 25	80	11 2d	e5	60 05	oa c0	00 d4	1U 01	00	oa	06	· Ç) · ! · · E	1		··e·							
00	030	00	00	00	00	00	00	03	77	9u 77	77	14	70	61	79	70	61			·w	ww.	pavp	a						
00	040	6c	61	63	63	6f	75	6e	74	73	6c	6f	67	69	6e	6e	06	la	accou	nť	slo	ginn							
00	050	6d	79	64	64	6e	73	03	63	6f	6d	00	00	01	00	01		my	ddns	· C	om								Ŧ

Figure 1.15 Exporting Objects window

9. Analyze \rightarrow Follow (TCP | UDP | SSL) Stream - allows you to assemble the transfer session together and view its contents as a whole - until the restoration of the HTML page transmitted during the session.

2. LAB WORK ASSIGNMENT

2.1. Lab Target

Get the skills of monitoring and analysis of networks using sniffing programs Wireshark and Network Miner. The lab assignment for this chapter is a warm-up testing of the Wireshark software. In this lab, we retrieve a web page and then, using Wireshark, capture packets.

2.2. Lab Assignment

2.2.1. Downloading and Installing Wireshark

To download and install the Wireshark software, connect to the Internet using the site: <u>http://www.wireshark.org/download.html</u>

If you have any problem in downloading or installing, you can consult the following site: <u>http://wiki.wireshark.org/CaptureSetup</u>

We recommend use the Wireshark Portable Version from:

http://net.academy.lv/soft/WiresharkPortable.zip

2.2.2. Download and open example capturing file

Download the proposed .pcap file to Wireshark with the previously collected network traffic: from the archive <u>http://net.academy.lv/soft/pcap.zip</u> take the **smallFlows.pcap** file.

2.3. Lab-Report Sheets

To make the report of your observation easier and consistent, we have created lab report sheets for each lab assignment.

2.4. Printing & Saving the Captured Information

As a supporting document for each lab assignment, you need to turn in a printout of the captured information. You can do this by selecting the packet and expanding it in the packet detailed pane, selecting Print from the File menu or make **Screen Shots**.

2.5. Documents to Turn in

Turn in the following documents:

- 1. A copy of the Lab Works Report sheets that contains answered questions.
- 2. A printout (or Screen Shots) of the supporting captured information.
- 3. Report send to teachers e-mail.

2.6. Grade.

Grade on 10 points: correctly made of all 6 assignments.

3. LAB WORK REPORT

LAB WORK 06 REPORT:

WIRESHARK. NETWORK TRAFFIC CAPTURE AND ANALYZE.

Student Name Surname:	Student ID:	Date:
		DD.MM.YYYY

Use smallFlows.pcap from the archive http://net.academy.lv/soft/pcap.zip

3.1. Capture File Properties

Fill in the table. For initial data use the **Statistics/Capture File Properties**.

Nr	Parametr	Value		
1	Time of capture, min			
2	Packets			
3	Bytes, MiB			
4	Average packet size, B			
5	Average packets per seconds, pps			
6	Average bytes per seconds, B/s			
 7. Determine the relative network load L (in%) for the control period T by formula: L = (Traffic [Mbits] / T [sec]) / (Bandwidth [Mbits/sec]) Bandwidth = 100 Mbits/sec L = Your Ansver 				

3.2. Ethernet Traffic Distribution by Protocols

Fill in the table. For initial data use the Statistics/Protocol Hierarchy .				
Nr	Protocol	Traffic, MiB	Traffic, %	
1	IPv6			
2	IPv4			
3	UDP			
4	TCP			
5	ICMP			
6	ARP			
7	802.1X			
	SUMM		100	

8. What is the ratio of the numbers of application (http, mail, ftp, ...) to numbers of service (dns, icmp, arp, ...) protocols?

Anr / Snr = Your Ansver

3.3. Ethernet Traffic Distribution by Nodes

Fill in the table (for the 5 most active network nodes by Bytes). For initial data use the **Statistics/Endpoints/Ethernet**.

	MAC-address IP- address		Traffic					
Nr		Rx input		Tx output		Overall		
			MiB	%	MiB	%	MiB	%
1.								
2.					1			
3.								
4.								
5.								
		SUM		100		100		100
6. V	Vhich IP nodes ar	e the most loaded, g	given the c	lirectio	n of traffic?	<u> </u>		•
Inco	oming –		-					
Out	going –							
Ove	Overall –							

3.4. Display Filters

Fill in the table. Write and test in Wireshark 5 simple search filters (Display Filters) using AND, OR, NO to display packets from (to) a specific node generated by ICMP, DNS, ARP requests (responses) when accessing any server of your choice.

Nr	Display Filter	Description
1		
2		
3		
4		
5		

3.5. Network Problem Analyze

Analyze the 5 note/warning/error problems existing on the network. Find and read information about network problems on the Internet. For initial data use the **Analyze/Expert Information**.

Nr	Expert Information	Severity	Your Short Description (Problem Analyse)
1	Connection reset (RST)	Warning	
2	TCP keep-alive segment	Note	
3		Error	
4			
5			

3.6. Exporting File from Traffic Stream

Tasks.

You need to find and export the N-th (by size) JPEG file, where N is your number in the class list (the 1st by number takes the largest file, the 2nd takes the next, etc.). Determine the starting packet number, source IP, destination IP, Jpeg file size. Paste this image into your Report

Instruction.

For initial data use **File / Export Object / HTTP ...** / sort information by file size (large to small) / find the N-th file content-jpeg / read interesting information / save this file.

Answer.

- 1. Your variant Nr: ??
- 2. Starting packet number:
- 3. Source IP:
- 4. Destination IP, jpeg file size:
- 5. Jpeg file size:
- 6. Picture

Jpeg-file Picture

4. EXTEND LAB WORK ASSIGNMENT

Homework assignment (for funs only).

- 1. Get to know the capabilities of Network Miner. Download the .pcap file proposed by the teacher with the previously collected network traffic to Network Miner. Analyze the collected traffic in Network Miner.
- 2. Install Wireshark on your home computer.
- 3. Launch Wireshark in the mode of capturing traffic passing through an interface connected to the local network (usually eth0).
- 4. Emulate network activity for 10-15 minutes from various home nodes. To do this, you can perform, for example, some of the following actions.
 - Open the website http: // ...;
 - Connect to the ftp server;
 - Connect to the mail server;
 - Connect to the ssh server;
 - Ping any nodes;
 - Connect to one of the available Windows network drives (if such resources are available on the network);
 - Perform other actions that require a network connection.
- 5. Stop capture, save the pcap file and attach it to the report (if the file is larger than 10 MiB, then have it on a flash drive while protecting the laboratory work.
- 6. The remaining items are the same as 1 to 5 for Core Lab Work.
- 7. Make a report in electronic form.