

计算机科学与工程系

Department of Computer Science and Engineering

CS 315 Computer Security Course

Lab 1: Packet Sniffing and Wireshark

Introduction

The first part of the lab introduces packet sniffer, Wireshark. Wireshark is a free opensource network protocol analyzer. It is used for network troubleshooting and communication protocol analysis. Wireshark captures network packets in real time and display them in human-readable format. It provides many advanced features including live capture and offline analysis, three-pane packet browser, coloring rules for analysis. This document uses Wireshark for the experiments, and it covers Wireshark installation, packet capturing, and protocol analysis.



Figure 1: Wireshark in Kali Linux



Background

TCP/IP Network Stack



Figure 2: Encapsulation of Data in the TCP/IP Network Stack

In the Introduction to Computer Networking Course, TCP/IP network stack is introduced and studied. This background section briefly explains the concept of TCP/IP network stack to help you better understand the experiments. TCP/IP is the most commonly used network model for Internet services. Because its most important protocols, the Transmission Control Protocol (TCP) and the Internet Protocol (IP) were the first networking protocols defined in this standard, it is named as TCP/IP. However, it contains multiple layers including application layer, transport layer, network layer, and data link layer.

- Application Layer: The application layer includes the protocols used by most applications for providing user services. Examples of application layer protocols are Hypertext Transfer Protocol (HTTP), Secure Shell (SSH), File Transfer Protocol (FTP), and Simple Mail Transfer Protocol (SMTP).



- Transport Layer: The transport layer establishes process-to-process connectivity, and it provides end-to-end services that are independent of underlying user data. To implement the process-to-process communication, the protocol introduces a concept of port. The examples of transport layer protocols are Transport Control Protocol (TCP) and User Datagram Protocol (UDP). The TCP provides flowcontrol, connection establishment, and reliable transmission of data, while the UDP is a connectionless transmission model.
- Internet Layer: The Internet layer is responsible for sending packets to across networks. It has two functions: 1) Host identification by using IP addressing system (IPv4 and IPv6); and 2) packets routing from source to destination. The examples of Internet layer protocols are Internet Protocol (IP), Internet Control Message Protocol (ICMP), and Address Resolution Protocol (ARP).
- *Link Layer*: The link layer defines the networking methods within the scope of the local network link. It is used to move the packets between two hosts on the same link. An common example of link layer protocols is Ethernet.

Packet Sniffer

Packet sniffer is a basic tool for observing network packet exchanges in a computer. As the name suggests, a packet sniffer captures ("sniffs") packets being sent/received from/by your computer; it will also typically store and/or display the contents of the various protocol fields in these captured packets. A packet sniffer itself is passive. It observes messages being sent and received by applications and protocols running on your computer, but never sends packets itself.

Figure 3 shows the structure of a packet sniffer. At the right of **Figure** 3 are the protocols (in this case, Internet protocols) and applications (such as a web browser or ftp client) that normally run on your computer. The packet sniffer, shown within the dashed rectangle in **Figure** 3 is an addition to the usual software in your computer, and consists of two parts. The packet capture library receives a copy of every link-layer frame that is sent from or received by your computer. 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. In Figure 1, the assumed physical media is an Ethernet, and so all upper-layer protocols are eventually encapsulated within an Ethernet frame. Capturing all link-layer frames thus gives you access to all messages sent/received from/by all protocols and applications executing in your computer.

The second component of a packet sniffer is the packet analyzer, which displays the contents of all fields within a protocol message. In order to do so, the packet analyzer





Figure 3: Packet Sniffer Structure

must "understand" the structure of all messages exchanged by protocols. For example, suppose we are interested in displaying the various fields in messages exchanged by the HTTP protocol in **Figure** 3. The packet analyzer understands the format of Ethernet frames, and so can identify the IP datagram within an Ethernet frame. It also understands the IP datagram format, so that it can extract the TCP segment within the IP datagram. Finally, it understands the TCP segment. Finally, it understands the TCP segment. Finally, it understands the HTTP protocol and so, for example, knows that the first bytes of an HTTP message will contain the string "GET," "POST," or "HEAD".

We will be using the Wireshark packet sniffer [http://www.wireshark.org/] for these labs, allowing us to display the contents of messages being sent/received from/by protocols at different levels of the protocol stack. (Technically speaking, Wireshark is a packet analyzer that uses a packet capture library in your computer). Wireshark is a free network protocol analyzer that runs on Windows, Linux/Unix, and Mac computers.



Getting Wireshark

The Kai Linux has Wireshark installed. You can just launch the Kali Linux VM and open Wireshark there. Wireshark can also be downloaded from here:

https://www.wireshark.org/download.html



Figure 4: Download Page of Wireshark



Starting Wireshark

When you run the Wireshark program, the Wireshark graphic user interface will be shown as **Figure** 5. Currently, the program is not capturing the packets.



Figure 5: Initial Graphic User Interface of Wireshark

Then, you need to choose an interface. If you are running the Wireshark on your laptop, you need to select WiFi interface. If you are at a desktop, you need to select the Ethernet interface being used. Note that there could be multiple interfaces. In general, you can select any interface but that does not mean that traffic will flow through that interface. The



network interfaces (i.e., the physical connections) that your computer has to the network are shown. The attached **Figure** 6 was taken from my computer.

After you select the interface, you can click start to capture the packets as shown in **Figure** 7.

			Wireshark: Capture Inter	aces		• • •
		Device	Description	IP	Packets	Packets/s
		eth0		172.16.108.151	0	0
	P	any		none	0	0
	_	lo		127.0.0.1	0	0
	8	bluetooth0		none	44	0
	P	nflog		none	0	0
	.	nfqueue		none	0	0
		usbmon1		none	0	0
		usbmon2		none	0	0
He	elp		Start	Stop	ions	Close

Figure 6: Capture Interfaces in Wireshark

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help					Capturing	from ethC) [Wir	eshark 1.	12.6 (Git Re	/ Unknown	from ur	iknown)]			•	•
Silter Expression Clear Apply Save 0. Time Source Destination Protocol Length Info 1 0.00000000000000000000000000172.16.108.151 172.16.108.2 DNS 72 Standard query 0x38c5 A www.kali.org 3 0.00002910000 172.16.108.151 172.16.108.2 DNS 72 Standard query 0x630c AAAA www.kali.org 4 0.0000900000 172.16.108.151 172.16.108.2 DNS 74 Standard query 0x80c AAAA tww.kali.org 5 0.0012830000 172.16.108.151 172.16.108.2 DNS 74 Standard query 0x3dbb A www.offensive-security.com Frame 4: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0 Ethernet II, Src: Ymware_6d:7a:35 (00:0c:29:6d:7a:35), Dst: Ymware_f0:1a:b5 (00:50:56:f0:1a:b5) Internet Protocol Version 4, Src: 172.16.108.151 (172.16.108.151), Dst: 172.16.108.2 (172.16.108.2) User Datagram Protocol, Src Port: 46328 (46328), Dst Port: 53 (53) Domain Name System (query) 000 00 50 55 f 0 1a b5 00 0c 29 6d 7a 35 08 00 45 00 .PV mz5E. 000 00 00 50 55 f 0 1a b5 00 0c 29 6d 7a 35 08 00 45 00 .PV mz5E. 000 00 00 00 00 00 00 00 00 20 26 5	ile E	dit View Go C	Capture Anal	yze Stati	stics Tel	ephony To	ools In	ternals H	lelp								
0. Time Source Destination Protocol Length Info 1 0.000000000 172.16.108.151 172.16.108.2 DNS 72 Standard query 0x28c5 A www.kali.org 3 0.000291000 172.16.108.151 172.16.108.2 DNS 74 Standard query 0x28c5 AAAA tww.kali.org 4 0.000290000 172.16.108.151 172.16.108.2 DNS 74 Standard query 0x28c9 AAAA tools.kali.org 4 0.000290000 172.16.108.151 172.16.108.2 DNS 74 Standard query 0x28c9 AAAA tools.kali.org 5 0.001283000 172.16.108.151 172.16.108.2 DNS 86 Standard query 0x28c9 AAAA tools.kali.org Frame 4: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0 Ethernet IT, Src: Wmware_6d:7a:35 (00:00:02:29:6d:7a:35), Dst: Vmware_f0:1a:b5 (10:00:50:56:f0:1a:b5) Internet Protocol Version 4, Src: 172.16.108.151 (172.16.108.2) (172.16.108.2) Use Datagram Protocol, Src Port: 46328 (46328), Dst Port: 53 (53) Domain Name System (query)		ک 🗖 🔪 🖲	÷ 🗇 آ	¢ م	् 🔹	► ₹	Ŧ	1	€	ର୍ ପ୍	🖭 🥁	(🗹		?			
1 0.00000000 172.16.108.151 172.16.108.2 DNS 72 Standard query 0x28c5 A www.kali.org AAAA www.kali.org AAAA www.kali.org AAAA www.kali.org AAAA www.kali.org AAAA www.kali.org AAAA www.kali.org A 0.000390000 172.16.108.151 172.16.108.2 DNS 74 Standard query 0x28a0 AAAA tools.kali.org A 0.000390000 172.16.108.151 172.16.108.2 DNS 86 Standard query 0x28a0 AAAA tools.kali.org 5 0.001283000 172.16.108.151 172.16.108.2 DNS 86 Standard query 0x28a0 AAAA tools.kali.org T4 Standard query 0x28a0 AAAA tools.kali.org A 0.000390000 172.16.108.151 172.16.108.2 DNS 86 Standard query 0x28a0 AAAA tools.kali.org 5 0.001283000 172.16.108.151 172.16.108.2 DNS 86 Standard query 0x28a0 AAAA tools.kali.org T4 Standard query 0x28a0 O AAAA tools.kali.org DNS 80 Standard query 0x28a0 AAAA tools.kali.org S 0.001283000 172.16.108.151 172.16.108.2 DNS 86 Standard query 0x38b A www.offensive-security.com Sthere at 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0 Ethernet IT, Src: Ymware_Gd7:ar35, 000:00:29:0617a:35), Dost: Ymware fo:1a:b5 (00:50:56:f0:1a:b5) Internet Protocol Version 4, Src: 172.16.108.151 (172.16.108.151), Dst: 172.16.108.2 (172.16.108.2) User Datagram Protocol, Src Port: 46328 (46328), Dst Port: 53 (53) Domain Name System (query)	ilter:					~	Expres	ssion Cl	ear Ap	ply Sa	ve						
2 0.000093000 172.16.108.151 172.16.108.2 DNS 72 Standard query 0x630c AAAA www.kali.org 3 0.000291000 172.16.108.151 172.16.108.2 DNS 74 Standard query 0x200 AAAA tools.kali.org 5 0.001283000 172.16.108.151 172.16.108.2 DNS 86 Standard query 0x3db A www.offensive-security.com Frame 4: 74 bytes on vire (592 bits), 74 bytes captured (592 bits) on interface 0 Ethernet II, Src: Vmware_dd:7a:35 (00:0c:29:6d:7a:35), Dst: Vmware_f0:1a:b5 (00:50:56:f0:1a:b5) Internet Protocol Version 4, Src: 172.16.108.151 (172.16.108.151), Dst: 172.16.108.2 (172.16.108.2) User Datagram Protocol, Src Port: 46328 (46328), Dst Port: 53 (53) Domain Name System (query)									ocoll								
3 0.000291000 172.16.108.151 172.16.108.2 DNS 74 Standard query 0xbd4f A tools.kali.org 4 0.000390000 172.16.108.151 172.16.108.2 DNS 74 Standard query 0x2a0 AAAA tools.kali.org 5 0.001283000 172.16.108.151 172.16.108.2 DNS 86 Standard query 0x2a0 AAAA tools.kali.org Frame 4: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0 86 Standard query 0x2a0 A www.offensive-security.com Ethernet II, Src: Ymware_6d:7a:35 (00:02:29:6d:7a:35), Dst: YIZ:16.108.151), Dst: 172.16.108.2 (172.16.108.2) User Datagram Protocol, Src Port: 46328 (46328), Dst Port: 53 (53) Domain Name System (query) 00 00 50 56 f0 1a b5 00 0c 29 6d 7a 35 08 00 45 00 .PV)mz5E. 100 03 c 37 6f 40 00 40 11 28 7a c10 6 6 97 ac 10 .4700.0 200 00 00 00 00 00 00 00 00 00 00 57 4 6f 6f 6 73 30 00 00 11 200 00 00 00 00 00 00 00 00 74 46 6f 6 6 73 01 00 00 01 30 00 00 00 00 00 00 00 00 00 75 76 6f 6 7 73 01 00 00 01 <t< td=""><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></t<>																	
4 0.000390000 172.16.108.151 172.16.108.2 DNS 74 Standard query 0x2a0 AAAA tools.kali.org 5 0.001283000 172.16.108.151 172.16.108.2 DNS 86 Standard query 0x2a0 AAAA tools.kali.org Frame 4: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0 86 Standard query 0x3dbb A www.offensive-security.com Frame 4: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0 86 Standard query 0x3dbb A www.offensive-security.com Ethernet II, Src: Ymware_6d:77:35 (00:00:229:6d:78:35), Dst: Vmware_f0:18:b5 (00:50:56:f0:18:b5) Internet Protocol Version 4, Src: 172.16:108:151 Internet Protocol Version 4, Src: 172.16:108:151 Dst: 172.16:108:12 User Datagram Protocol, Src Port: 46328 (46328), Dst Port: 53 (53) Domain Name System (query) 00 50 56 f0 1a b5 00 0c 29 6d 7a 35 08 00 45 00 .PV Jmz5.E. 00 00 50 56 f0 1a b5 00 0c 29 6d 7a 35 08 00 45 00 .PV Jmz5.E. 10 00 50 56 f0 1a b5 00 0c 29 6d 7a 35 08 00 45 00 .PV Jmz5.E.																	
5 0.001283000 172.16.108.151 172.16.108.2 DNS 86 Standard query 0x3dbb A www.offensive-security.com Frame 4: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0 Ethernet II, Src: Wmware_6d:7a:35 (00:0c:29:6d:7a:35), Dst: Wmware_f0:1a:b5 (00:50:56:f0:1a:b5) Internet Protocol Version 4, Src: 172.16.108.151 (172.16.108.151), Dst: 172.16.108.2 (172.16.108.2) User Datagram Protocol, Src Port: 46328 (46328), Dst Port: 53 (53) Domain Name System (query) 000 00 50 56 f0 1a b5 00 0c 29 6d 7a 35 08 00 45 00 .PV)mz5E. 100 03 cs 37 6f 40 00 40 11 287 ac 10 6c 97 ac 10 .<70@.@l																	
Frame 4: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0 Ethernet II, Src: Vmware_6d:7a:35 (00:0c:29:6d:7a:35), Dst: Vmware_f0:1a:b5 (00:50:56:f0:1a:b5) Internet Protocol Version 4, Src: 172.16.108.151 (172.16.108.151), Dst: 172.16.108.2 (172.16.108.2) User Datagram Protocol, Src Port: 46328 (46328), Dst Port: 53 (53) Domain Name System (query) 000 00 50 56 f0 1a b5 00 29 6d 7a 35 08 0.9V)mz5E. 000 00 50 56 f0 1a b5 00 45 00 .PV)mz5E. 010 00 37 f6 40 04 11 d2 87 ac 10																	
Ethernet II, Śrc: Vmware_6d:7a:35 (00:0c:29:6d:7a:35), Dst: Vmware_f0:1a:b5 (00:50:56:f0:1a:b5) Internet Protocol Version 4, Src: 172.16.108.151 (172.16.108.151), Dst: 172.16.108.2 (172.16.108.2) User Datagram Protocol, Src Port: 46328 (46328), Dst Port: 53 (53) Domain Name System (query)		5 0.001283000) 172.16.10	8.151	172	.16.108.2	2	DNS		86	Standar	d query	0x3dbb	A www.offe	nsive-sec	urity.co	m
110 00 3c 37 6f 40 00 40 11 d2 87 ac 10 6c 97 ac 10 .<70@.@l 120 6c 02 b4 f8 00 35 00 28 63 53 2e a0 01 00 00 01 l5.(c5 130 00 00 00 00 00 00 5 74 6f 6f 6c 73 04 6b 61 6ctools.kal																	
10 00 3c 37 6f 40 00 40 11 d2 87 ac 10 6c 97 ac 10 .<70@.@l 20 6c 02 b4 f8 00 35 00 28 63 53 2e a0 01 00 00 01 l5.(c5 30 00 00 00 00 00 05 74 6f 6f 6c 73 04 6b 61 6ctools.kal			,														
110 00 3c 37 6f 40 00 40 11 d2 87 ac 10 6c 97 ac 10 ,<70@,@l)20 6c 02 b4 f8 00 35 00 28 63 53 2e a0 01 00 00 01 l5.(c5)30 00 00 00 00 00 00 57 4 6f 6f 6c 73 04 6b 61 6ctools.kal			,														
110 00 3c 37 6f 40 00 40 11 d2 87 ac 10 6c 97 ac 10 .<70@.@l 120 6c 02 b4 f8 00 35 00 28 63 53 2e a0 01 00 00 01 l5.(c5 130 00 00 00 00 00 00 5 74 6f 6f 6c 73 04 6b 61 6ctools.kal																	
10 00 3c 37 6f 40 00 40 11 d2 87 ac 10 6c 97 ac 10 .<70@.@l 120 6c 02 b4 f8 00 35 00 28 63 53 2e a0 01 00 00 01 l5.(c5 130 00 00 00 00 00 00 05 74 6f 6f 6c 73 04 6b 61 6ctools.kal																	
110 00 3c 37 6f 40 00 40 11 d2 87 ac 10 6c 97 ac 10 ,<70@,@l)20 6c 02 b4 f8 00 35 00 28 63 53 2e a0 01 00 00 01 l5.(c5)30 00 00 00 00 00 00 57 4 6f 6f 6c 73 04 6b 61 6ctools.kal																	
030 00 00 00 00 00 00 05 74 6f 6f 6c 73 04 6b 61 6c t ools.kal	000	00 50 56 f0 1		29 6d 7	a 35.08.	00.45.00	PV) mz	5. F.								
			a b5 00 0c				.<70	0.0	.l								
	010 020	00 3c 37 6f 4 6c 02 b4 f8 0	a b5 00 0c 0 00 40 11 0 35 00 28	d2 87 a 63 53 20	c 10 6c : e a0 01	97 ac 10 00 00 01	.<70 l	@.@ .5.(cS.	.l								
🕅 eth0: ve capture in progress> File: /t Packets: 42 · Displayed: 42 (100.0%) Profile: Default	010 020 030	00 3c 37 6f 4 6c 02 b4 f8 0 00 00 00 00 0	a b5 00 0c 0 00 40 11 0 35 00 28 0 00 57 44	d2 87 a 63 53 26 6f 6f 6	c 10 6c : e a0 01	97 ac 10 00 00 01	.<70 l	@.@ .5.(cS. t ool	.l								







Figure 8: Wireshark Graphical User Interface on Microsoft Windows

The Wireshark interface has five major components:

The **command menus** are standard pulldown menus located at the top of the window. Of interest to us now is the File and Capture menus. The File menu allows you to save captured packet data or open a file containing previously captured packet data, and exit the Wireshark application. The Capture menu allows you to begin packet capture.

The **packet-listing window** displays a one-line summary for each packet captured, including the packet number (assigned by Wireshark; this is not a packet number contained in any protocol's header), the time at which the packet was captured, the packet's source and destination addresses, the protocol type, and protocol-specific information contained in the packet. The packet listing can be sorted according to any of these categories by clicking on a column name. The protocol type field lists the highest-level protocol that sent or received this packet, i.e., the protocol that is the source or ultimate sink for this packet.



The **packet-header details window** provides details about the packet selected (highlighted) in the packet-listing window. (To select a packet in the packet-listing window, place the cursor over the packet's one-line summary in the packet-listing window and click with the left mouse button.). These details include information about the Ethernet frame and IP datagram that contains this packet. The amount of Ethernet and IP-layer detail displayed can be expanded or minimized by clicking on the right-pointing or down-pointing arrowhead to the left of the Ethernet frame or IP datagram line in the packet details window. If the packet has been carried over TCP or UDP, TCP or UDP details will also be displayed, which can similarly be expanded or minimized. Finally, details about the highest-level protocol that sent or received this packet are also provided.

The **packet-contents window** displays the entire contents of the captured frame, in both ASCII and hexadecimal format.

Towards the top of the Wireshark graphical user interface, is the **packet display filter field**, into which a protocol name or other information can be entered in order to filter the information displayed in the packet-listing window (and hence the packet-header and packet-contents windows). In the example below, we'll use the packet-display filter field to have Wireshark hide (not display) packets except those that correspond to HTTP messages.



Capturing Packets

After downloading and installing Wireshark, you can launch it and click the name of an interface under Interface List to start capturing packets on that interface. For example, if you want to capture traffic on the wireless network, click your wireless interface.

Test Run

Do the following steps:

- 1. Start up the Wireshark program (select an interface and press start to capture packets).
- 2. Start up your favorite browser (ceweasel in Kali Linux).
- 3. In your browser, go to Wayne State homepage by typing www.wayne.edu.
- 4. After your browser has displayed the http://www.wayne.edu page, stop Wireshark packet capture by selecting stop in the Wireshark capture window. This will cause the Wireshark capture window to disappear and the main Wireshark window to display all packets captured since you began packet capture see image below:





- 5. Color Coding: You'll probably see packets highlighted in green, blue, and black. Wireshark uses colors to help you identify the types of traffic at a glance. By default, green is TCP traffic, dark blue is DNS traffic, light blue is UDP traffic, and black identifies TCP packets with problems — for example, they could have been delivered out-of-order.
- 6. You now have live packet data that contains all protocol messages exchanged between your computer and other network entities! However, as you will notice the HTTP messages are not clearly shown because there are many other packets included in the packet capture. Even though the only action you took was to open your browser, there are many other programs in your computer that communicate via the network in the background. To filter the connections to the ones we want to focus on, we have to use the filtering functionality of Wireshark by typing "http" in the filtering field as shown below:

	Capturing from ethO	[Wireshark 1.12	6 (Git Rev Unknown from unknown)]	• •
File Edit View Go Capture Analyze Sta	istics Telephony Tools Interna	ls Help		
● ● ▲ ■ ▲ 市 🗎 ¥ ᠀	९. ♦ ⇒ .२ ∓ ± 📃	🔹 ୧୧୦	1	
Filter: http	- Expression	Clear Apply Sa	/e	
No. Time Source		Protocol Length		
4124 25.63383500 172.16.108.152	141.217.1.160		GET /promos/1376/programs-min_1.png HTTP/1.1	
4126 25.63399800 172.16.108.152			GET /promos/1376/apply-students-2015.jpg HTTP/1.1 HTTP/1.1 200 OK (text/javascript)	
4160 25.66536200 141.217.1.160 4173 25.67091200 172.16.108.152	172.16.108.152 141.217.1.160		GET /promos/1376/winter-registration2015-3section 1.jpg HTT	D/1 1
4173 25.67288800 141.217.1.160	172.16.108.152		HTTP/1.1 200 0K (PNG)	-/1.1
4208 25.67323100 172.16.108.152			GET /promos/1380/flu-shot-wayne-edu.jpg HTTP/1.1	
4223 25.68279200 141.217.1.160	172.16.108.152		HTTP/1.1 200 OK (GIF89a)	
4233 25.68285500 141.217.1.160	172.16.108.152		HTTP/1.1 200 OK (PNG)	
4235 25.68306500 172.16.108.152	141.217.1.160	HTTP 508	GET /images/news/van-jones-news.jpg HTTP/1.1	
4236 25.68308400 172.16.108.152	141.217.1.160		GET /_resources/images/footer/give-to-wsu.gif HTTP/1.1	
4250 25.68478900 141.217.1.160	172.16.108.152	HTTP 1402	HTTP/1.1 200 OK (PNG)	
Frame 4126: 513 bytes on wire (4104	bits), 513 bytes captured (4104 bits) on ir	terface O	
Ethernet II, Src: Vmware_6d:7a:35 (
Internet Protocol Version 4, Src: 1				
Transmission Control Protocol, Src H	Port: 52099 (52099), Dst Por	t: 80 (80), Seq:	1, Ack: 1, Len: 459	
Hypertext Transfer Protocol				
)mz5E.		
010 01 f3 32 ce 40 00 40 06 5e 15 a 020 01 a0 cb 83 00 50 f1 b7 ba 26 3		^l .&.iz.P.		
030 72 10 c3 8a 00 00 47 45 54 20 2		T /promo		
040 73 2f 31 33 37 36 2f 61 70 70 6	ic 79 2d 73 74 75 s/1376/a	pply-stu		
		15.jpg H .Host: w		
		User-A		
080 67 65 6e 74 3a 20 4d 6f 7a 69 6	6c 6c 6l 2f 35 2e gént: Mo	zilla/5.		
090 30 20 28 58 31 31 3b 20 4c 69 6 0a0 38 36 3b 20 72 76 3a 33 31 2e 3		Linux i6 1.0) Gec		
1040 58 50 50 20 72 76 54 55 51 20 3 10b0 6b 6f 2f 32 30 31 30 30 31 30 3		101 Fire		
0c0 66 6f 78 2f 33 31 2e 30 20 49 6		Iceweas		
)0d0 65 6c 2f 33 31 2e 38 2e 30 0d ()0e0 74 3a 20 69 6d 61 67 65 2f 70 (OAccep /png.ima		
		.8,*/*;q		
		1 A A		
) 💅 eth0: <live capture="" in="" progress=""> File: /t</live>	Packets: 5085 · Displayed: 60 (1.2%)		Profile:	Default

Notice that we now view only the packets that are of protocol HTTP. However, we also still do not have the exact communication we want to focus on because using HTTP as a filter is not descriptive enough to allow us to find our connection to http://www.wayne.edu. We need to be more precise if we want to capture the correct set of packets.



7. To further filter packets in Wireshark, we need to use a more precise filter. By setting the http.host==sustech, we are restricting the view to packets that have as an http host the www.wayne.edu website. Notice that we need two equal signs to perform the match "==" not just one. See the screenshot below:

http.host == sustech Wireless controls are not s . Time 1 1088 33.640931 1091 34.641026 1093 35.641764	upported in this version of Wireshark.			λ Q. Π.		Expression
Wireless controls are not s Time 1088 33.640931 1091 34.641026 1093 35.641764	Source					
Time 1088 33.640931 1091 34.641026 1093 35.641764	Source					
1088 33.640931 1091 34.641026 1093 35.641764						802.11 Preference:
1091 34.641026 1093 35.641764		Destination	Protocol	Length Info		
1093 35.641764	192.168.1.100	239.255.255.250	SSDP	217 M-SEARCH * HTTP/1		
	192.168.1.100	239.255.255.250	SSDP	217 M-SEARCH * HTTP/1		
	192.168.1.100 192.168.1.100	239.255.255.250 239.255.255.250	SSDP SSDP	217 M-SEARCH * HTTP/1. 217 M-SEARCH * HTTP/1.		
	192.168.1.100	218.30.103.56	HTTP		.i 6496&bi=1&sohuurl=http%3A%2F%2Fww	w ifens comb7E&cod=275f03a77
	s on wire (1736 bits), 217 bytes ca :83:e7:67:28:18 (a4:83:e7:67:28:18)					
	rsion 4, Src: 192.168.1.100, Dst: 2		1.13 (01.06			
	ol, Src Port: 60366 (60366), Dst Po					
Hypertext Transfer Pi						
00 01 00 5e 7f ff fa	a a4 83 e7 67 28 18 08 00 45 00	^				
10 00 cb 9d 24 00 00	0 01 11 69 f7 c0 a8 01 64 ef ff	\$ id				
20 ff fa eb ce 07 6c	: 00 b7 7d 09 4d 2d 53 45 41 52	l }.M-SEAR				
30 43 48 20 2a 20 48 40 4f 53 54 3a 20 32		CH * HTT P/1.1H OST: 239 .255.255				
50 2e 32 35 30 3a 31	1 35 39 20 52 55 55 20 52 55 55 1 39 30 30 0d 0a 4d 41 4e 3a 20	.250:190 0MAN:				
60 22 73 73 64 70 3a	a 64 69 73 63 61 76 65 72 22 Ød	"ssdp:di scover".				
		.MX: 1 ST: urn: dial-mul tiscreen				
70 0a 4d 58 3a 20 31						
70 0a 4d 58 3a 20 31 80 64 69 61 6c 2d 6d						
70 0a 4d 58 3a 20 31 80 64 69 61 6c 2d 6d 90 2d 6f 72 67 3a 73 a0 6c 3a 31 0d 0a 55	8 65 72 76 69 63 65 3a 64 69 61 5 53 45 52 2d 41 47 45 4e 54 3a	-org:ser vice:dia l:1USE R-AGENT:				
170 0a 4d 58 3a 20 31 180 64 69 61 6c 2d 6d 190 2d 6f 72 67 3a 73 1a0 6c 3a 31 0d 0a 55 1a0 20 47 6f 6f 67 6c	3 65 72 76 69 63 65 3a 64 69 61 5 53 45 52 2d 41 47 45 4e 54 3a 65 20 43 68 72 6f 6d 65 2f 37	l:1USE R-AGENT: Google Chrome/7				
70 0a 4d 58 3a 20 31 80 64 69 61 6c 2d 6d 90 2d 6f 72 67 3a 73 80 6c 3a 31 0d 0a 55 50 20 47 6f 6f 67 6c	8 65 72 76 69 63 65 3a 64 69 61 5 53 45 52 2d 41 47 45 4e 54 3a 6 52 04 43 68 72 6f 6d 65 2f 37 3 30 39 2e 31 30 30 20 4d 61 63	l:1USE R-AGENT:				

8. Now, we can try another protocol. Let's use Domain Name System (DNS) protocol as an example here.

				ethO [Wiresha	ırk 1.12.6 (Git	Rev Unknov	vn from	unknown)]		e	
ile Edit	View Go Ca	apture Analyze	Statistics Telephony T	ools Internals	Help						
• •	1	10 🗋 × 🤆	o Q ♦ ⇒ .2 ∰	± 🔳 🔹	େ ୍ ୍ ପ୍	FF 🙀	1	2			
ilter: dr	าร		-	Expression	Clear Apply Sa	ve					
		Source	Destination		otocol Length						
		172.16.108.2	172.16.108.1							cc.wayne.edu A 141.21	7.1.160
		172.16.108.15							onts.gstatic.com		
		172.16.108.15							A fonts.gstatic.com		
		172.16.108.2	172.16.108.							Issl.l.google.com A 19	
		172.16.108.2	172.16.108.							issl.l.google.com AAAA	2607:1
		172.16.108.15							nsights.hotjar.com		
		172.16.108.15							A insights.hotjar.com		-16 -
		172.16.108.2	172.16.108.							1411736383.eu-west-1.	
		172.16.108.2	172.16.108.						nsights.hotjar.com	1411736383.eu-west-1.	etp.ama
		172.16.108.15							A insights.hotjar.com A insights.hotjar.com		
							query	0x/9/2 AAA	A insights.notjar.com	1	
			1680 bits), 210 bytes								
			5 (00:50:56:f0:la:b5)					>			
			: 172.16.108.2 (172.)			152 (172.)	16.108.	152)			
			:: 53 (53), Dst Port:	40430 (40430)							
Domain	Name System	(response)									
000 000	Oc 29 6d 7a	35 00 50 56	f0 1a b5 08 00 45 00)mz5.P V.	E.						
			da ac 10 6c 02 ac 10	к.							
			50 6c 5a 81 80 00 01	l5F							
	65 64 75 00		77 05 77 61 79 6e 65 c0 0c 00 05 00 01 00	w ww							
	00 05 00 0e		32 70 72 6f 64 02 63								
	c0 10 c0 2b		00 00 00 05 00 04 8d	c+							
	01 a0 c0 10		00 00 00 05 00 06 03								
	73 32 c0 10		00 01 00 00 00 05 00	ns2							
	02 6e 73 c0		01 00 01 00 00 00 05 01 00 01 00 00 05	nsg							
			1c 00 01 00 00 00 05	U							
		00 00 00 fo	Oe 00 00 00 00 00 00								
00 00 00	02			••							
-			-								
		k_pcapng_eth0_2	 Packets: 5487 · Display 							Profile: Default	



9. Let's try now to find out what are those packets contain by following one of the conversations (also called network flows), select one of the packets and press the right mouse button (if you are on a Mac use the command button and click), you should see something similar to the screen below:

	*ethO [Wireshark	1.12.6 (Git Rev Unknown from unknown)]	• •	C
ile Edit View Go Capture Analyze Statistics Tele	phony Tools Internals Help			
● ● ▲ ■ ▲ 前 ■ × ハ へ ◆ ⇒		. Q 🝸 📓 🕅 🛅 🕄 💈		
Filter: dns	Expression Clear Appl	y Save		
	ination Protocol Le			
	16.108.152 DNS	219 Standard query response 0xd529 CNAME insights-1411736383.eu-west-1.elb.amazona		
	16.108.152 DNS	386 Standard query response 0x2bd1 CNAME insights-1411736383.eu-west-1.elb.amazona	ws.com	A
4822 26.13119000 172.16.108.152 172. 4823 26.13129600 172.16.108.152 172.		tandard query 0xbbe2 A insights.hotjar.com tandard query 0x7972 AAAA insights.hotjar.com		
4823 26.13129600 172.16.108.152 172. 4824 26.13378100 172.16.108.2 172.		tandard query 0x/9/2 AAAA insignts.notjar.com tandard query response 0x7972 CNAME insights-1411736383.eu-west-1.elb.amazona		
4824 26.13378100 172.16.108.2 172. 4826 26.13538500 172.16.108.2 172.		tandard query response 0x79/2 CNAME insights 1411/36383.eu-west 1.elb.amazona		
4820 26.22835200 172.16.108.2 172.		tandard query 0x2bfd A login.wayne.edu	ws.com	A
4833 26.22846200 172.16.108.152 172.		tandard query 0x45c6 AAAA login.wayne.edu		
4834 26.22862000 172.16.108.152 172.		itandard query 0xd660 A parents.wayne.edu		
4835 26.22871500 172.16.108.152 172.	EUILFACKEL	tandard query 0x242b AAAA parents.wayne.edu		
4836 26.22886300 172.16.108.152 172.	16 Packet Comment	tandard query 0x287c A alumni.wayne.edu		
Frame 4814: 386 bytes on wire (3088 bits), 38 Ethernet II, Src: Vmware_f0:1a:b5 (00:50:56:f		arface 0)c:29:6d:7a:35)		
Internet Protocol Version 4, Src: 172.16.108. User Datagram Protocol, Src Port: 53 (53), Ds		> ³² (172.16.108.152)		
Domain Name System (response)	Prepare a Filter	>		
	Conversation Filter	>		
	Colorize Conversation	>		
	SCTP	>		
	Follow TCP Stream			
	Follow UDP Stream			
0000 00 0c 29 6d 7a 35 00 50 56 f0 1a b5 08 0 0010 01 74 bf 5f 00 00 80 11 49 5e ac 10 6c 0				
020 6c 98 00 35 c8 46 01 60 4e d2 2b d1 81 8 030 00 04 00 04 00 04 08 69 6e 73 69 67 68 7	0 4 Copy	>		
040 68 6f 74 6a 61 72 03 63 6f 6d 00 00 01 0 050 0c 00 05 00 01 00 00 00 05 00 2e 13 69 6	e Protocol Preferences	>		
060 67 68 74 73 2d 31 34 31 31 37 33 36 33 3 070 65 75 2d 77 65 73 74 2d 31 03 65 6c 62 0				
080 61 7a 6f 6e 61 77 73 c0 1c c0 31 00 01 0				
090 00 00 05 00 04 34 1f 8f 37 c0 31 00 01 0 0a0 00 00 05 00 04 34 1e 80 ff c0 31 00 01 0				
0b0 00 00 05 00 04 36 ab ca 9a c0 45 00 02 0	0 Show Packet in New Window			
00c0 00 00 05 00 13 06 6e 73 2d 33 34 31 09 6 00d0 64 6e 73 2d 34 32 c0 1c c0 45 00 02 00 0				
10d0 64 66 73 20 34 32 00 10 00 45 00 02 00 0				
ofo 6e 73 2d 32 38 03 6e 65 74 00 c0 45 00 0	2 00 01 ns-28.ne tE			
File: "/tmp/wireshark_pcapng_eth0_2 Packets: 548	7 · Displayed: 260 (4.7%) · Dropped: 0 (0.0	0%) Profile: Default		

Click on Follow UDP Stream, and then you will see following screen.





10. If we close this window and change the filter back to "http.host==www.wayne.edu" and then follow a packet from the list of packets that match that filter, we should get the something similar to the following screens. Note that we click on **Follow TCP Stream** this time.

۲	••				📕 Wi-Fi	: en1		
		🧟 💿 📘	💻 🗋 🖹 🙆 🔍 👄 🔶 🚞	₮ ± 🗔 🔳	Θ, Θ			
II.	nttp.hos	st == sustech						Expression +
	Wireles	s controls are no	at supported in this version of Wireshark.					802.11 Preferences
No.		Time	Source	Destination	Protocol	Length	Info	
	1088	33.640931	192.168.1.100	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1	
	1091	34.641026	192.168.1.100	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1	
	1093	35.641764	192.168.1.100	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1	
		36.642013	192.168.1.100		SSDP	217	M-SEARCH * HTTP/1.1	
	5415	60.960002	192,168.1.108	213-20-103-125 Mark/Umank Packet Ignore/Unignore Packet Set/Unset Time Referen Time Shift Packet Comment Edit Resolved Name Apply as Filter Prepare a Filter Conversation Filter Colorize Conversation SCTP	0°₩1	n D T	6ET /answer?3d=536496	Sbi=l6sohuurl=http%3A42F%2Fwww.ifeng.com%2F6rnd=225f93a720
b.	Frame	5415: 905 byt	tes on wire (7240 bits), 905 bytes capt	Follow			CP Stream	
•	Ethern	net II, Src: a	a4:83:e7:67:28:18 (a4:83:e7:67:28:18),	Сору	(54:75:		JDP Stream SL Stream	
•	Transm		Version 4, Src: 192.168.1.100, Dst: 218 ol Protocol, Src Port: 62072 (62072), D Protocol	Protocol Preferences Decode As Show Packet in New Wir			n: 839	

0000	54	75	95	37	5b	39	a4	83	e7	67	28	18	08	00	45	00	Tu.7[9	.g(E.	
0010	03	7b	00	00	40	00	40	06	34	1a	с0	a8	01	64	da	1e	.{@.@.	4d	
0020	67	38	f2	78	00	50	c7	dd	af	dd	e7	3a	58	a4	80	18	g8.x.P	:X	
0030	08	0a	94	Зc	00	00	01	01	08	0a	56	c4	44	Ød	66	3c	<	V.D.f<	
0040	fa	26	47	45	54	20	2f	61	6e	73	77	65	72	3f	69	64	.&GET /a	nswer?id	
0050						39					3d							bi=1&soh	
0060						68					33							p%3A%2F%	
0070						2e					67							eng.com%	
0080						64					66							25f93a72	
0090						61					3d							if=1&w=3	
00a0						31					73							&js=c&z=	
00b0						32					36							436b8da3	
00c0						36					70							6&ps=156	
00d0	37	31	34	32	35	38	35	37	38	37	26	69	74	3d	30	26	71425857	87⁢=0&	

%2F&rnd=225f93a720922da2&if=1&w=360&h=112&js=c&z: left=440⊤=1259&op=100&csp=2560,1417&bcl=365,1		87⁢=0&vs=0&ft=0&vt=0&
Host: eff.inte.sogou.com	,,,	
Connection: keep-alive Cache-Control: max-age=0		
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS	X 10 14 6) AppleWebKit/537 36 (KHTML]i	ke Gecko) (brome/
76.0.3809.100 Safari/537.36	x 10_14_0/ Appreneukit/557.50 (kinic, ti	ike deekdy enrolley
Accept: */*		
Referer: http://www.ifeng.com/a_if/190312/weicc/	testv2.html	
Accept-Encoding: gzip, deflate		
Accept-Language: en-US,en;q=0.9		
Cookie: SUID=5DCDD98D566C860A5627C152000A48A2; wi		
<pre>CXID=015D78A945F88808CD4D26336B54FE93; SUV=00B22 ad=KZllllllll2NQaI@lllllVCo7uclllll0N6aBZllll911</pre>		
du=N2++++++++++++++++++++++++++++++++++++		
HTTP/1.1 200 OK		
Server: nginx		
Date: Fri, 30 Aug 2019 05:23:07 GMT		
Content-Type: text/plain; charset=UTF-8		
Content-Length: 0		
Connection: keep-alive Last-Modified: Mon, 10 Sep 2012 10:27:21 GMT		
ETag: "8008008d-0-4c956671a7440"		
Accept-Ranges: bytes		
client pkt(s), 1 server pkt(s), 1 turn.		
	Show data as ASCII	Stream 144
Entire conversation (1.096 bytes)		
Entire conversation (1,096 bytes)		
Entire conversation (1,096 bytes)		Find Next



Questions for the Lab

- 1. Carefully read the lab instructions and finish all tasks above.
- 2. If a packet is highlighted by black, what does it mean for the packet?
- 3. What is the filter command for listing all outgoing http traffic?
- 4. Why does DNS use Follow UDP Stream while HTTP use Follow TCP Stream?
- 5. Using Wireshark to capture the FTP password.

There is a FTP server installed on the Kali Linux VM. You need to use a terminal to log into the server and use Wireshark to capture the password. The username for the FTP server is csc5991-student, and the password is [WSU-csc5991.] without the brackets. You will user the username and password to login the FTP server while Wireshark is running. Note that the FTP server is installed on the localhost, make sure you select the right interface for the capturing. You need to explain to me how you find the password and a screenshot of the password packet. Have fun!