



COLLEGE OF ENGINEERING

CSC 5991 Cyber Security Practice

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.

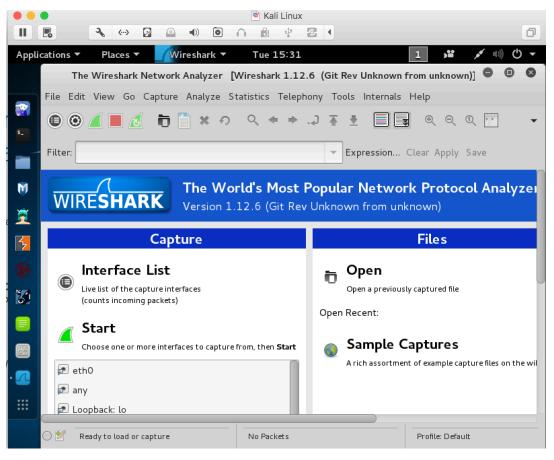


Figure 1: Wireshark in Kali Linux



Background

TCP/IP Network Stack

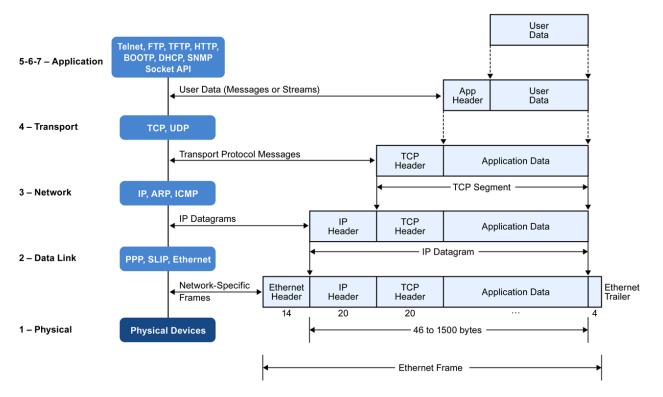


Figure 2: Encapsulation of Data in the TCP/IP Network Stack

In the CSC 4190 Introduction to Computer Networking (one of the perquisite courses), 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



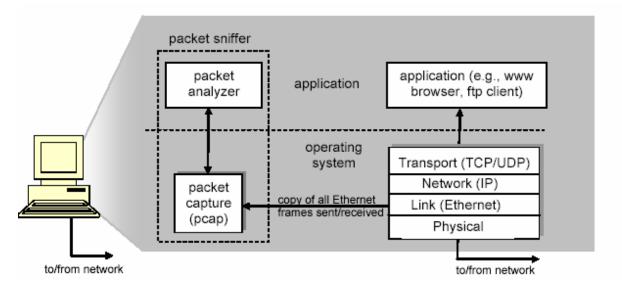


Figure 3: Packet Sniffer Structure

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 structure, so it can extract the HTTP message contained in 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

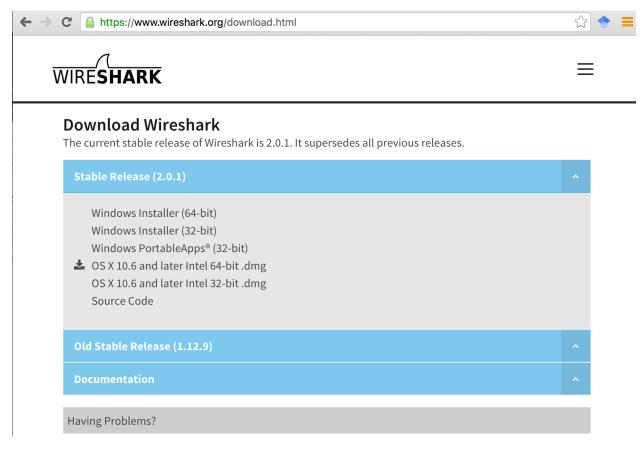


Figure 4: Download Page of Wireshark



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.

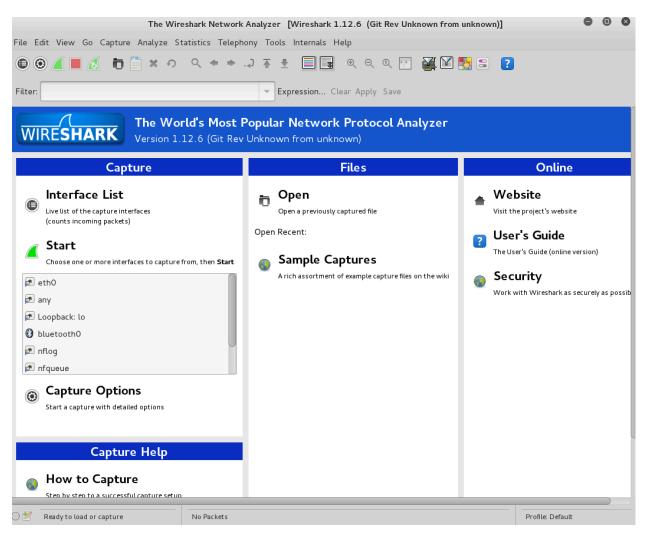


Figure 5: Initial Graphic User Interface of Wireshark

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 Interf	aces	(
		Device	Description	IP	Packets	Packets/s
	F	eth0		172.16.108.151	0	0
		any		none	0	0
	P	lo		127.0.0.1	0	0
	8	bluetooth0		none	44	0
	F	nflog		none	0	0
	F	nfqueue		none	0	0
		usbmon1		none	0	0
		usbmon2		none	0	0
He	lp		Start	Stop Opt	ions	Close

Figure 6: Capture Interfaces in Wireshark

				Capt	uring fr	om ethO	[Wi	reshark	1.12.6	(Git I	Rev U	nknown	from unl	(nown)]				•	0	⊗
File E	dit View Go	Capture Ana	lyze Sta	atistics	Teleph	nony Ta	ols Ir	nternals	Help											
•) 📕 📕 🙇	÷ 🗇 🗇	к О	٩, ١	+ +	.ə 🛧	<u>*</u>		•	Q	0, [•	. 🗹 🍢	8	?					
Filter:						-	Expre	ession	Clear A	Apply	Save									
	Time 1 0.00000000 2 0.00009300	0 172.16.10	08.151		172.10	5.108.2 5.108.2		DN DN	S	J	72 S 72 S	tandaro	query	0x630c	A www.ka AAAA www	.kali.o				
	3 0.00029100 4 0.00039000 5 0.00128300	0 172.16.10	08.151		172.10	5.108.2 5.108.2 5.108.2		DN DN DN	s		74 S	tandaro	query	0x2ea0	A tools. AAAA too A www.of	ls.kali	.org	com	1	
Doma	xin Name Syste	m (query)																		
0010 0020 0030	00 50 56 f0 1 00 3c 37 6f 4 6c 02 b4 f8 0 00 00 00 00 0 69 03 6f 72 6	40 00 40 11 00 35 00 28 00 00 05 74	d2 87 63 53 6f 6f	ac 10 2e a0	6c 97 01 00	ac 10 00 01	.<70 l)r)r 	l S ols.ka											
) 💅 🤇	eth0: <live capture<="" td=""><td>e in progress> Fi</td><td>le: /t</td><td>Packe</td><td>ts: 42 · D</td><td>isplayed: 4</td><td>12 (100</td><td>.0%)</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>Prof</td><td>ile: Default</td><td></td><td></td><td></td></live>	e in progress> Fi	le: /t	Packe	ts: 42 · D	isplayed: 4	12 (100	.0%)								Prof	ile: Default			





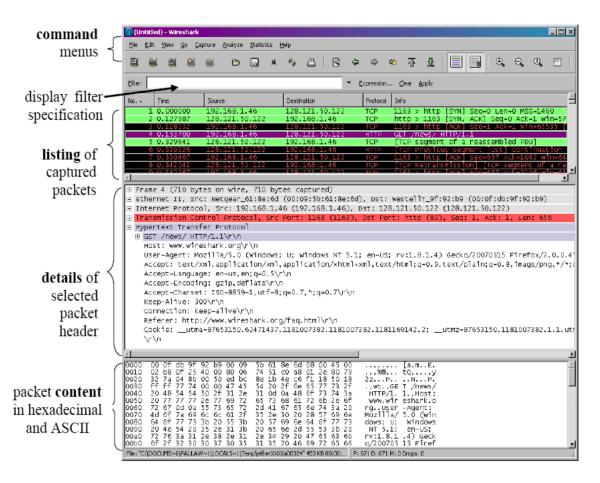


Figure 8: Wireshark Graphical User Interface on Microsoft Windows

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:

ile Edit View		Capturing from e	thO [Wireshark]	1.12.6 (Git Rev Unknown from	unknown)]	0
	Go Capture Analyze Statis	tics Telephony Tools Int	ternals Help			
0 0 🖌 🛯	📕 🧟 🛅 🗋 🗙 ໑ ເ	λ ★ ⇒ ₀ ∓ ±	.	۵. 🖭 🎽 🔛 🗉	2	
ilter:		- Expres	ssion Clear Apply	/ Save		
o. Time	Source	Destination	Protocol Ler	ngth Info		
	408250 172.16.108.152	192.122.185.45	TLSv1.2	95 Application Data		
	410640 192.122.185.45	172.16.108.152	TCP		202574 Ack=5826 Win=64240 Len=0	
	495260 192.122.185.45	172.16.108.152	TLSv1.2	95 Application Data		
	495330 172.16.108.152	192.122.185.45	TCP		826 Ack=202615 Win=65535 Len=0	
	2604700 172.16.108.152 2606360 50.31.164.175	50.31.164.175 172.16.108.152			87→443 [ACK] Seq=2186 Ack=3183 W 443→33587 [ACK] Seq=3183 Ack=21	
	393630 172.16.108.1	172.16.108.152		223 Dropbox LAN sync Disc		.87 Win=64240 Len=0
	2774900 172.16.108.152	50.31.164.175	TCP		37→443 [ACK] Seq=2186 Ack=3183 W	hin-42408 Len-0
	2776770 50.31.164.175	172.16.108.152	TCP		443-33587 [ACK] Seq=3183 Ack=21	
	8646130 50.31.164.175	172.16.108.152	TCP		Seg=3183 Ack=2187 Win=64240 Len	
	896550 172.16.108.1	172.16.108.255	DB-LSP-D	223 Dropbox LAN sync Disc		
Internet Pro	, Src: Vmware_f0:la:b5 (00 otocol Version 4, Src: 23. n Control Protocol, Src Po	61.75.27 (23.61.75.27)	, Dst: 172.16.10	8.152 (172.16.108.152)	0	
Transmission	i Control Protocol, Src Po	rt: 80 (80), Dst Port:	44266 (44266),	Seq: 3547, ACK: 883, Len:	U	
000 00 0c 29	9 6d 7a 35 00 50 56 f0 1a 0 c7 00 00 80 06 0f 08 17		z5.P VE.			
	0 50 ac ea e4 43 ca e9 a4					
010 00 28 b0						
010 00 28 b0 020 6c 98 00	9 03 00 00 00 00 00 00 00					
010 00 28 b0 020 6c 98 00	0 00 00 00 00 00 00 00 00					
010 00 28 b0 020 6c 98 00	9 03 00 00 00 00 00 00 00 00	000				
010 00 28 b0 020 6c 98 00	9 03 00 00 00 00 00 00 00 00	00				
010 00 28 b0 020 6c 98 00	9 03 00 00 00 00 00 00 00	000				
010 00 28 b0 020 6c 98 00	9 03 00 00 00 00 00 00 00 00	000				
010 00 28 b0 020 6c 98 00	9 03 00 00 00 00 00 00 00 00					
010 00 28 b0 020 6c 98 00	20 00 00 00 00 00 00 00 00					
010 00 28 b0 020 6c 98 00	9 03 00 00 00 00 00 00 00					
010 00 28 b0 020 6c 98 00	9 00 00 00 00 00 00 00 00	00				



- 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 [Wiresha	ark 1.12.	6 (Git Rev Unknow	n from unknown)]	e	6
File Edit	t View Go C	apture Analyze St	atistics Telephony Too	ls Internals Help					
••		n 🗋 🛪 o	९. ♦ ⇒ .⊅ ∓	± .	ର୍ପ୍	🖭 💓 🗹 🍢	8 ?		
Filter:	ittp		•	Expression Clear A	pply Sav	/e			
	Time	Source	Destination	Protocol					
		172.16.108.152 172.16.108.152	141.217.1.160	HTTP			5/programs-min_1.png HTTP/1.1 5/apply-students-2015.jpg HTTP/1.1		
		141.217.1.160	172.16.108.15				(text/javascript)		
4173		172.16.108.152	141.217.1.160	HTTP			5/winter-registration2015-3section	1.ipg HTTP/1.1	
4187	25.67288800	141.217.1.160	172.16.108.15	2 HTTP		HTTP/1.1 200 OK			
4208	25.67323100	172.16.108.152	141.217.1.160	HTTP	512	GET /promos/1380	0/flu-shot-wayne-edu.jpg HTTP/1.1		
4223	25.68279200	141.217.1.160	172.16.108.15	2 HTTP	1284	HTTP/1.1 200 OK	(GIF89a)		
4233	25.68285500	141.217.1.160	172.16.108.15	2 HTTP	402	HTTP/1.1 200 OK	(PNG)		
		172.16.108.152	141.217.1.160	HTTP			s/van-jones-news.jpg HTTP/l.l		
		172.16.108.152	141.217.1.160	HTTP			/images/footer/give-to-wsu.gif HTTP	9/1.1	
		141.217.1.160	172.16.108.15			HTTP/1.1 200 OK	(PNG)		
			4 bits), 513 bytes c						
			(00:0c:29:6d:7a:35),						
			172.16.108.152 (172. Port: 52099 (52099)						
	text Transfer		Port: 52099 (52099)	, DSI POPI: 80 (80), seq:	I, ACK: I, Len:	459		
и пурет і	text fransfer	FIOLOCOL							
			7a 35 08 00 45 00	.PV)mz5E.					
			ac 10 6c 98 8d d9 14 6a 7a c6 50 18	2.@.@. ^l P&.jz.P.					
030 72	2 10 c3 8a OC	00 47 45 54 20	2f 70 72 6f 6d 6f	rGE T /promo					
			6c 79 2d 73 74 75	s/1376/a pply-stu					
	4 65 6e /4 /3 4 54 50 2f 31		2e 6a 70 67 20 48 6f 73 74 3a 20 77	dents-20 15.jpg H TTP/1.1Host: w					
	4 54 50 21 31 1 79 6e 65 2e		55 73 65 72 2d 41	avne.eduUser-A					
	7 65 6e 74 3a		6c 6c 61 2f 35 2e	gent: Mo zilla/5.					
	0 20 28 58 31		6e 75 78 20 69 36	0 (X11; Linux i6					
			30 29 20 47 65 63 31 20 46 69 72 65	86; rv:3 1.0) Gec ko/20100 101 Fire					
			31 20 46 69 72 65 63 65 77 65 61 73	fox/31.0 Iceweas					
			Oa 41 63 63 65 70	el/31.8. 0Accep					
00e0 74	4 3a 20 69 6d	61 67 65 2f 70	6e 67 2c 69 6d 61	t: image /png,ima					
00f0 67	7 65 2f 2a 3b	71 3d 30 2e 38	2c 2a 2f 2a 3b 71	ge/*;q=0 .8,*/*;q	1				
) 🕅 🔤	th0: clive capture	in progress> File: /t	Packets: 5085 · Displayed:	60 (1 2%)				Profile: Default	
e e		in progress rite. /t	rackets, 5005 Displayed.	00 (1.2.10)				Pronte, Deraud	

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==www.wayne.edu, 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:

*eth0 [Wireshark 1.12.6 (Git Rev Unknown from unknown)]	
ile Edit View Go Capture Analyze Statistics Telephony Tools Internals Help	
🗎 🖲 🥖 🛅 📩 🛪 🤈 오 🍝 🔸 🎝 ቾ 🛨 📃 📑 오 익 익 🍸 🎆 🗹 🔂 5 💈	
ilter: http:host == www.wayne.edu 🗾 👻 Expression Clear Apply Save	
. Time Source Destination Protocol Length Info	
4019 25.43579900 172.16.108.152 141.217.1.160 HTTP 467 GET / HTTP/1.1	
Frame 4019: 467 bytes on wire (3736 bits), 467 bytes captured (3736 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: 172.16.108.152 (172.16.108.152), Dst: 141.217.1.160 (141.217.1.160)	
Transmission Control Protocol, Src Port: 52089 (52089), Dst Port: 80 (80), Seq: 1, Ack: 1, Len: 413	
Hypertext Transfer Protocol	
200 00 50 56 f0 la b5 00 0c 29 6d 7a 35 08 00 45 00 .PV)mz5.E. 10 01 c5 37 81 40 00 40 06 59 90 ac 10 6c 98 8d d97.6.a, YL	
10 0 1 c5 37 81 40 00 40 06 59 90 ac 10 6c 98 80 09,e,e	
330 72 10 bc ee 00 00 47 45 54 20 2f 20 48 54 54 50 rGET / HTTP	
J40 2f 3l 2e 3l 0d 0a 48 6f 73 74 3a 20 77 77 77 2e /l.l.Ho st: www. 50 77 6l 79 6e 65 2e 65 64 75 0d 0a 55 73 65 72 2d wayne.ed uUser-	
00 41 67 65 66 74 3a 20 4d 6f 7a 69 6c 6c 61 2f 35 Agent: M ozilla/5	
070 2e 30 20 28 58 31 31 3b 20 4c 69 6e 75 78 20 69 .0 (X11; Linux i	
380 36 38 36 30 20 72 76 3a 33 31 2e 30 29 20 47 65 686; rv: 31.0) Ge	
)90 63 6b 6f 2f 32 30 31 30 30 31 30 46 69 72 cko/2010 010 Fir 30 65 66 6f 78 2f 33 12 e 30 20 49 63 65 77 65 61 efox/31.0 Icewea	
Jao 63 66 67 27 33 33 12 6 38 2e 30 0d 0a 41 63 63 65 sel/31.8 .0. Acce	
0c0 70 74 3a 20 74 65 78 74 2f 68 74 6d 6c 2c 6l 70 pt: text /html,ap	
0d0 70 6c 69 63 61 74 69 6f 6e 2f 78 68 74 6d 6c 2b plicatio n/xhtml+	
2e0 78 6d 6c 2c 61 70 70 6c 69 63 61 74 69 6f 6e 2f xml,application/ 5f0 78 6d 6c 3b 71 3d 30 2e 39 2c 2a 2f 2a 3b 71 3d xml;a=0.9.*/*:a=	
)f0 78 6d 6c 3b 71 3d 30 2e 39 2c 2a 2f 2a 3b 71 3d xml;q=0, 9,*/*;q=	
	Profile: Default

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

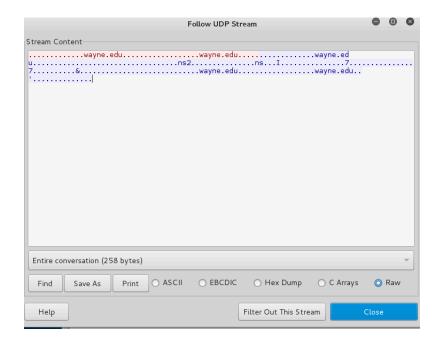
ile Edit View Go Capture Analyze S	Statistics Telephony Tools			
		internals Help		
ම් 🖲 📕 🧟 🛅 🗮 ව	् + + .⊒ ∓ ±	.	Q 🖭 🎬 🖺 📑 🕐	
ilter: dns	- Exp	ression Clear Apply	Save	
o. Time Source	Destination	Protocol Ler		
4015 25.43294300 172.16.108.2	172.16.108.152	DNS		6c5a CNAME whv2prod.cc.wayne.edu A 141.217.1.16
4128 25.64762900 172.16.108.152	172.16.108.2	DNS	77 Standard query 0x7e2e A f	
4129 25.64776200 172.16.108.152		DNS	77 Standard query Oxa3fa AAA	
4130 25.65439100 172.16.108.2	172.16.108.152	DNS		7e2e CNAME gstaticadssl.l.google.com A 192.122.
4203 25.67303700 172.16.108.2	172.16.108.152	DNS		a3fa CNAME gstaticadssl.l.google.com AAAA 2607:
4808 26.12153900 172.16.108.152		DNS	79 Standard query 0x2bdl A i	
4809 26.12165600 172.16.108.152		DNS	79 Standard query 0xd529 AAA	
4811 26.12442700 172.16.108.2	172.16.108.152	DNS		d529 CNAME insights-1411736383.eu-west-1.elb.am
4814 26.12644900 172.16.108.2	172.16.108.152	DNS		2bdl CNAME insights-1411736383.eu-west-1.elb.am
4822 26.13119000 172.16.108.152		DNS	79 Standard query Oxbbe2 A i	
4823 26.13129600 172.16.108.152	172.16.108.2	DNS	79 Standard query 0x7972 AAA	A insights.hotjar.com
User Datagram Protocol, Src Port: Domain Name System (response)	53 (53), Dst Port: 4043	10 (40430)		
000 00 0c 29 6d 7a 35 00 50 56 f	0 10 55 08 00 45 00)mz5.P VE.		
010 00 c4 bd 93 00 00 80 11 4b d		кl		
020 6c 98 00 35 9d ee 00 b0 12 5		.5Plz		
030 00 02 00 02 00 03 03 77 77 7 040 03 65 64 75 00 00 01 00 01 c		w ww.wayne du		
		wh v2prod.c		
060 63 c0 10 c0 2b 00 01 00 01 0	10 00 00 05 00 04 8d c.	+		
		2 nsq		
Da0 00 04 8d d9 9a a2 c0 55 00 0		U		
obo oo o4 8d d9 9a aa co 55 oo 1		U		
0c0 00 10 26 06 97 00 00 00 fo 0 0d0 00 02		δι		
40 00 02				



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 [Wireshare	1.12.6 (Git Rev Unknown from unknown)]	• •	•
e Edit View Go Capture Analyze Statistics	Telephony Tools Internals Help			
) 🖲 📕 🖉 🛅 📩 🛪 ० ९ 🗢	÷.,) ∓ ± 🗐 🚽 € 6	L Q 🛅 🎬 🖆 🕾 🔋		
ilter: dns	Expression Clear App	y Save		
Time Source D	estination Protocol Le	ngth Info		
	72.16.108.152 DNS	219 Standard query response 0xd529 CNAME insights-1411736383.eu-west-1.elb.amazo		
	72.16.108.152 DNS	386 Standard query response 0x2bd1 CNAME insights-1411736383.eu-west-1.elb.amazo	naws.com	A
	72.16 Mark Packet (toggle) 72.16	tandard query Oxbbe2 A insights.hotjar.com tandard query Ox7972 AAAA insights.hotjar.com		
	72.16 Ignore Packet (toggle)	tandard query exponse 0x7972 CNAME insights-1411736383.eu-west-1.elb.amazo	naws.com	
	72.16 Set Time Reference (toggle)	tandard guery response Oxbbe2 CNAME insights 1411736383.eu-west-1.elb.amazo		
4832 26.22835200 172.16.108.152	72.10	tandard query Ox2bfd A login.wayne.edu		
4833 26.22846200 172.16.108.152	72.16 Time Shift	tandard query 0x45c6 AAAA login.wayne.edu		
	72.16 Edit Packet	itandard query 0xd660 A parents.wayne.edu		
	72.16	itandard query 0x242b AAAA parents.wayne.edu		
	72.16 Packet Comment	tandard query 0x287c A alumni.wayne.edu		
Frame 4814: 386 bytes on wire (3088 bits), Ethernet II, Src: Vmware_f0:1a:b5 (00:50:5	6:fo: Manually Resolve Address	arface 0 Dc:29:6d:7a:35)		
Internet Protocol Version 4, Src: 172.16. User Datagram Protocol, Src Port: 53 (53),		> ³² (172.16.108.152)		
Domain Name System (response)	Prepare a Filter	>		
	Conversation Filter	>		
	Colorize Conversation	>		
	SCTP			
	Follow TCP Stream			
	Follow UDP Stream			
000 00 0c 29 6d 7a 35 00 50 56 f0 1a b5 0 010 01 74 bf 5f 00 00 80 11 49 5e ac 10 6				
220 6c 98 00 35 c8 46 01 60 4e d2 2b d1 8 330 00 04 00 04 00 04 08 69 6e 73 69 67 6	1 80	>		
040 68 6f 74 6a 61 72 03 63 6f 6d 00 00 0 050 0c 00 05 00 01 00 00 00 05 00 2e 13 6	9 6e Protocol Preferences	>		
060 67 68 74 73 2d 31 34 31 31 37 33 36 3 070 65 75 2d 77 65 73 74 2d 31 03 65 6c 6	2 09 Decode As			
080 61 7a 6f 6e 61 77 73 c0 1c c0 31 00 0 090 00 00 05 00 04 34 1f 8f 37 c0 31 00 0				
0a0 00 00 05 00 04 34 1e 80 ff c0 31 00 0	1 00			
0b0 00 00 05 00 04 36 ab ca 9a c0 45 00 0 0c0 00 00 05 00 13 06 6e 73 2d 33 34 31 0				
0d0 64 6e 73 2d 34 32 c0 1c c0 45 00 02 0	0 01 00 00 dns-42E			
0e0 00 05 00 16 06 6e 73 2d 37 33 39 09 6 0f0 6e 73 2d 32 38 03 6e 65 74 00 c0 45 0				
10 08 73 20 32 38 03 08 03 74 00 C0 45 0	0 02 00 01 HS-20.He L.E			
File: "/tmp/wireshark_pcapng_eth0_2 Packets:	5487 · Displayed: 260 (4.7%) · Dropped: 0 (0.	0%) Profile: Defaul	t	

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.

	*ethO [Wireshark 1.12.6 (Git Re	/ Unknown from unknown)]	00
le Edit View Go Capture Analyze Statistics Telephony Tools			
🗎 🖲 🔏 📕 🗴 ウ 🔍 🕈 🖢 🚺 🗎	📃 🛃 ९ ९ ९ 🖭 🌌	1 🔁 🖻 🔹 👔	
ilter: http.host==www.wayne.edu 🗾 Ex	pression Clear Apply Save		
o. Time Source Destination	Protocol Length Info		
4019 25.43579900 172.16.108.152 141.217.1.160	Mark Packet (toggle)	.1	
	Ignore Packet (toggle)		
	Set Time Reference (toggle)		
	Time Shift		
	Edit Packet		
	Packet Comment		
	Manually Resolve Address		
Frame 4019: 467 bytes on wire (3736 bits), 467 bytes cap Ethernet II, Src: Vmware 6d:7a:35 (00:0c:29:6d:7a:35), D	τι [']	>:b5)	
Internet Protocol Version 4, Src: 172.16.108.152 (172.16	.]	217.1.160)	
Transmission Control Protocol, Src Port: 52089 (52089), Hypertext Transfer Protocol	Conversation Filter	.en: 413	
	Colorize Conversation	Ś	
	SCTP	\$	
	Follow TCP Stream		
	Follow UDP Stream		
	Follow SSL Stream		
00 00 50 56 f0 1a b5 00 0c 29 6d 7a 35 08 00 45 00 .	N.		
10 01 c5 37 81 40 00 40 06 59 90 ac 10 6c 98 8d d9 . 20 01 a0 cb 79 00 50 d9 ba bf c9 7e 11 bf b9 50 18 .		·	
30 72 10 bc ee 00 00 47 45 54 20 2f 20 48 54 54 50 r 40 2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 77 77 77 2e //		>	
50 77 61 79 6e 65 2e 65 64 75 0d 0a 55 73 65 72 2d w. 60 41 67 65 6e 74 3a 20 4d 6f 7a 69 6c 6c 61 2f 35 A	je		
70 2e 30 20 28 58 31 31 3b 20 4c 69 6e 75 78 20 69 . 30 36 38 36 3b 20 72 76 3a 33 31 2e 30 29 20 47 65 6i			
90 63 6b 6f 2f 32 30 31 30 30 31 30 31 20 46 69 72 cl a0 65 66 6f 78 2f 33 31 2e 30 20 49 63 65 77 65 61 e	C Show Packet in New Window		
b0 73 65 6c 2f 33 31 2e 38 2e 30 0d 0a 41 63 63 65 s	l/31.8 .OAcce :: text /html.ap		
d0 70 6c 69 63 61 74 69 6f 6e 2f 78 68 74 6d 6c 2b p	icatio n/xhtml+ nl,appl ication/		
	l;q=0. 9,*/*;q=		
File: "/tmp/wireshark_pcapng_eth0_2 Packets: 5487 · Displayed: 1	0.0%) · Dropped: 0 (0.0%)	Pro	file: Default

. . .

Follow TCP Stream (tcp.stream eq 22)	•	9	0
Stream Content			
<pre>GET / HTTP/1.1 Host: www.wayne.edu User-Agent: Mozilla/S.0 (X11; Linux 1686; rv:31.0) Gecko/20100101 Firefox/31.0 Iceweasel/31.8.0 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8 Accept: Encoding: gzip, deflate Cookie: PHPSESSID=4887q:ar5niganphlh3psmebm7; _ga=GA1.2.1522201160.1452108309; _gat_Wayne=1; _gat_allWayneState=1 Connection: keep-alive HTTP/1.1 307 Temporary Redirect Server: nginx/0.8.55 Date: Wed, 06 Jan 2016 19:34:49 GMT Content-Tol: max-age=0 Expires: Wed, 06 Jan 2016 19:34:49 GMT Vary: Accept-Encoding Content-Encoding: gzip Content-Encoding: gzip Content-Length: 245</pre>			
Entire conversation (984 bytes)			-
Find Save As Print O ASCII O EBCDIC O Hex Dump O C Arrays	0	Raw	
Help Filter Out This Stream	lose		



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 [TBA in the Class]. 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!