Wireshark – Packet & Traffic Analysis

1 OVERVIEW

The learning objective of this lab is for students to get familiar with the concepts of packet and traffic analysis. There are many tools for packet sniffing, network traffic analysis, and HTTP debugging. However, Wireshark is the most popular, complete tool for the job. Understanding of this tool is often considered a required skill for many security and network auditing jobs.

1.1 LAB SETUP

Wireshark has already been installed on lab VMs. Wireshark also works on Windows and macOS. Administrator or root privileges are required to run Wireshark at full functionality.

2 LAB TASKS

2.1 TASK 1: CAPTURING A TRACE

2.1.1 Start Wireshark

Wireshark is a network packet analyzer. A network packet analyzer will try to capture network packets and tries to display that packet data as detailed as possible. Wireshark is already installed on Lab VM, start Wireshark from Dash menu on the left. You should see following window.

🧐 🗇 💿 The Wireshark Network Analyzer		
📶 🔳 🖉 🛯 🔛 🖸 🔍 🔇 🔍 🗸 > 🕞 🖃 🖉		
Apply a display filter <ctrl-></ctrl->		Expression +
Welcome to Wireshark		
Open		
/home/seed/example.pcapng (3983 KB)		
Capture		
using this filter:		-
enn0s3		
any/		
Loopback: lo/		Ú
nfqueue		
usbmon1		
Learn		
User's Guide Wiki Questions and Answers Mailing Lists		
You are running Wireshark 2.2.6 (Git Rev Unknown from unknown).		
Ready to load or capture	No Packets	Profile: Default

2.1.2 Wireshark Live Capture

Wireshark can capture traffic from many different network media types - and despite its name - including wireless LAN as well. Which media types are supported, depends on many things like the operating system you are using. An overview of the supported media types can be found at https://wiki.wireshark.org/CaptureSetup/NetworkMedia.

You can start a live capture of your ethernet device by clicking on a device **enp0s3** in the Capture list. This captures all traffic going through your ethernet device. This capture can cause performance issues as it captures and tries to present all information passing through your network device. To narrow down your search you can apply filters to your capture. For example, filter *tcp port http* will capture only TCP traffic through port 80.

2.1.2.1 Saving Captured Traces

You can save captured packets simply by using the File > Save As... menu item. You can choose which packets to save and which file format to be used. Not all information will be saved in a capture file. For example, most file formats don't record the number of dropped packets.

2.1.3 Inspecting Captured Traces

Wireshark also allows you to inspect captured packet traces from files. Wireshark is able to handle files generated from different software such as tcpdump, nmap, Microsoft Network Monitor, Cisco Networks NetXray, etc.

2.2 TASK 2: TCP BASICS AND BEHAVIOR

Start a TCP trace for your ethernet device (enp0s3)

Open a web browser and go to www.uhcl.edu

See Capturing from enp0s3 (tcp)									
◢ ■ ⊴ ◎ 🚔 🖹 🗙 🖉 🔍 🖌 ≽ 🛏 🖳 🔜 Θ 😐 🛯 🎛									
	pply a display filter <ctrl-></ctrl->					Expression +			
No.	Time	Source	Destination	Protocol Length	Info	F			
_	1 2018-09-24 14:53:35.5489213	10.0.2.15	23.78.220.19	TCP	74 55816 → 80 [SYN]	Seg=1828941974 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=			
	2 2018-09-24 14:53:35.5695650	23.78.220.19	10.0.2.15	TCP	60 80 → 55816 [SYN,	ACK] Seg=9224 Ack=1828941975 Win=32768 Len=0 MSS=1460			
	3 2018-09-24 14:53:35.5696022	10.0.2.15	23.78.220.19	TCP	54 55816 → 80 [ACK]	Seg=1828941975 Ack=9225 Win=29200 Len=0			
	4 2018-09-24 14:53:35.5708106	10.0.2.15	23.78.220.19	HTTP	348 GET /success.txt	HTTP/1.1			
	5 2018-09-24 14:53:35.5916046	23.78.220.19	10.0.2.15	HTTP	438 HTTP/1.1 200 OK	(text/plain)			
	6 2018-09-24 14:53:35.5916288	10.0.2.15	23.78.220.19	TCP	54 55816 → 80 [ACK]	Seg=1828942269 Ack=9609 Win=30016 Len=0			
	7 2018-09-24 14:53:40.9469722	10.0.2.15	129.7.81.102	TCP	74 59326 → 80 [SYN]	Seq=761945500 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3			
	8 2018-09-24 14:53:40.9882289	129.7.81.102	10.0.2.15	TCP	60 80 → 59326 [SYN,	ACK] Seg=9485 Ack=761945501 Win=32768 Len=0 MSS=1460			
	9 2018-09-24 14:53:40.9882572	10.0.2.15	129.7.81.102	TCP	54 59326 → 80 [ACK]	Seg=761945501 Ack=9486 Win=29200 Len=0			
	10 2018-09-24 14:53:42.2477870	10.0.2.15	129.7.81.102	HTTP	447 GET / HTTP/1.1				
	11 2018-09-24 14:53:42.2857019	129.7.81.102	10.0.2.15	HTTP	542 HTTP/1.1 303 See	Other (text/html)			
	12 2018-09-24 14:53:42.2857310	10.0.2.15	129.7.81.102	TCP	54 59326 → 80 [ACK]	Seq=761945894 Ack=9974 Win=30016 Len=0			
	13 2018-09-24 14:53:42.3697811	10.0.2.15	129.7.81.102	TCP	74 58592 → 443 [SYN] Seq=2108898049 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval			
	14 2018-09-24 14:53:42.4071213	129.7.81.102	10.0.2.15	TCP	60 443 → 58592 [SYN	, ACK] Seq=9749 Ack=2108898050 Win=32768 Len=0 MSS=1460			
	15 2018-09-24 14:53:42.4071746	10.0.2.15	129.7.81.102	TCP	54 58592 → 443 [ACK] Seq=2108898050 Ack=9750 Win=29200 Len=0			
	16 2018-09-24 14:53:42.4157662	10.0.2.15	129.7.81.102	TLSv1.2	571 Client Hello				
<u>د (</u>									
► Fr	ame 1: 74 bytes on wire (592 bits	s), 74 bytes captured	(592 bits) on interfa	ace 0					
► Et	hernet II, Src: PcsCompu_5b:9f:dc	(08:00:27:5b:9f:dc)	, Dst: RealtekU_12:35	:00 (52:54:00:12:	35:00)				
► Ir	nternet Protocol Version 4, Src: 1	LO.O.2.15, Dst: 23.78	.220.19						
► Tr	ansmission Control Protocol, Src	Port: 55816, Dst Por	t: 80, Seq: 1828941974	1, Len: 0					
1									
0000	E2 E4 00 12 25 00 08 00 27 Eb	0f do 08 00 45 00							
0000	0 52 54 00 12 35 00 06 00 27 50 0 00 20 ph c4 40 00 40 06 9f 97	91 UC 08 00 45 00	< 0.0 LE.						
0010	dc 13 da 08 00 50 6d 03 70 96								
002	72 10 ff 9e 00 00 02 04 05 b4	04 02 08 0a 00 05	r						
0040	db 52 00 00 00 00 01 03 03 07	04 02 00 04 00 00	. R						
0 7	epp0s3: <live capture="" in="" progress=""></live>				Pa	ckets: 7151 · Displayed: 7151 (100.0%) Profile: Default			
-	anpeser are capedre in progress.				- 10				

You should see a trace like following in your Wireshark window.

The green packets are HTTP, TCP segments are colored as light purple.

- *i* Can you explain how encrypted connection is established?
- *i* Do you see public key infrastructure at work?

Identify the TCP segments that are used to initiate the TCP connection between the client computer and www.uhcl.edu.

- *i* How many segments are used?
- *i* What is in the TCP header that identifies each segment as a handshaking segment?

How are sequence and acknowledgment numbers are determined?

- *i* Can you identify series of at least six TCP segments using these numbers?
- *i* What is the length of each of these six TCP segments? The length of the TCP segment is only the number of data bytes carried inside the segment (excluding the headers).
- *i* What is specified by the value of the Acknowledgement field in any received ACK-segment? How does www.uhcl.edu; for example, determine this value?
- *i* How much data (number of bytes) does the receiver typically acknowledge in one ACK?

2.3 TASK 3: HTTP BASICS AND BEHAVIOR

Go ahead and browse the university website to generate more HTTP traffic, you can also log in your webmail account (make sure you don't save any passwords on VM). If you leave capture open for long enough, you'll notice that it captures a lot of unnecessary TCP traffic as well.

For this exercise, you'll need to focus on HTTP traffic only. To make this simpler, apply an **http** filter to your capture. Once you are done browsing the web, you can stop the live capture.

))					🖾 🗔 🔹 Express
Time	Source	Destination	Protocol Len	th Info	
362 2018-09-24 15:00:04.6	860205. 10.0.2.15	72.21.91.29	OCSP	491 Request	
364 2018-09-24 15:00:04.7	115121 72.21.91.29	10.0.2.15	OCSP	842 Response	
49 2018-09-24 15:17:02.3	073641 10.0.2.15	184.26.62.17	OCSP	492 Request	
150 2018-09-24 15:17:02.3	298427 184.26.62.17	10.0.2.15	OCSP	923 Response	
12 2018-09-24 15:17:13.2	797311 10.0.2.15	72.21.91.29	OCSP	491 Request	
14 2018-09-24 15:17:13.3	079206_ 72.21.91.29	10.0.2.15	OCSP	842 Response	
257 2018-09-24 15:17:14.8	391321 10.0.2.15	72.21.91.29	OCSP	491 Request	
60 2018-09-24 15:17:14.8	657880 72.21.91.29	10.0.2.15	OCSP	842 Response	
62 2018-09-24 15:17:14.8	661468. 10.0.2.15	72.21.91.29	OCSP	491 Request	
68 2018-09-24 15:17:14.8	928678 72.21.91.29	10.0.2.15	OCSP	842 Response	
515 2018-09-24 15:17:15.4	8101/5_ 10.0.2.15	72.21.91.29	OCSP	491 Request	
27 2018-09-24 15:17:15.5	96/1/0 /2.21.91.29	10.0.2.15	OCSP	842 Response	
32 2018-09-24 15:17:15.5	10,0,2,15	72.21.91.29	OCSP	491 Request	
3/ 2010-03-24 10.1/.10.0	402903 10.0.2.10	23.33.111.21	UCSP	400 REQUEST	
20 2010-00-24 16:17:16 E	SEALES 72 24 04 20	10 0 2 15	0000	942 Decoopee	
39 2018-09-24 15:17:15.5 52 2018-09-24 15:17:15.6 8 9552: 559 bytes on wir	564466. 72.21.91.29 109913. 23.35.171.27 e (4472 bits). 559 bytes	10.0.2.15 10.0.2.15 captured (4472 bits)	OCSP OCSP	842 Response 559 Response	
539 2018-09-24 15:17:15.6 552 2018-09-24 15:17:15.6 me 9552: 559 bytes on wir ernet II, Src: Realteku] ernet Protocol Version 4, nsmission Control Protoco ource Port: 80 estination Port: 51976 Stream index: 125] TCP Segment Len: 505] equence number: 186202	564466. 72.21.91.29 189913. 23.35.171.27 2 (35:60 (52:54:06:12):35: 5 Src: 23.35.171.27, Dst: 1, Src Port: 80, Dst Por	10.0.2.15 10.0.2.15 captured (4472 bits) 00, 0st: PcsCompu_5b 10.0.2.15 1: 51976, Seq: 106262	OCSP OCSP on interface 6 :9f:dc (08:00:2) , Ack: 276878771	842 Response 559 Response :5b:9f:dc) , Len: 505	
539 2018-09-24 15:17:15.0 552 2018-09-24 15:17:15.0 552 2018-09-24 15:17:15.0 me 9552:559 bytes on wir ernet Protocol Version 4, nsmission Control Protoco ource Port: 89 Bestination Port: 51976 Stream index:1251 TCP Segment Len: 506] Sequence number: 10622 Next sequence number: 106	564466. 72.21.91.29 109913. 23.35.171.27 e (4472 bits), 559 bytes 2:35:00 (52:54:00:12:35: src: 23.35:71.27, Dst: l, src Port: 80, Dst Por	10.0.2.15 10.0.2.15 captured (4472 bits) 00), DSt: PcsCompu_5b 10.0.2.15 t: 51976, Seq: 106262	OCSP OCSP on interface 0 :9f:dc (08:00:27 , Ack: 276878771	842 Response 559 Response :Sb:9f:dc) , Len: 505	

Find an HTTP request from a client (source) to server (destination) and try to answer following questions.

i Inspect the HTTP header by expanding the down arrow beside "Hypertext Transfer Protocol" In the middle pane.

Observe that the HTTP header follows the TCP and IP headers, as HTTP is an application protocol that is transported using TCP/IP. To view it, select the packet, find the HTTP block in the middle panel, and expand it (by using the "v" expander or icon).

Explore the headers that are sent along with the request. First, you will see the GET method at the start of the request, including details such as the path. Then you will see a series of headers

in the form of tagged parameters. There may be many headers, and the choice of headers and their values vary from browser to browser.

See if you have any of these common headers

- i Host. A mandatory header, it identifies the name (and port) of the server.
- *i* User-Agent. The kind of browser and its capabilities.
- *i* Accept, Accept-Encoding, Accept-Charset, Accept-Language. Descriptions of the formats that will be accepted in the response, e.g., text/html, including its encoding, e.g., gzip, and language.
- *i* Cookie. The name and value of cookies the browser holds for the website.
- *i* Cache-Control. Information about how the response can be cached.

Now select an HTTP packet sent from server (source) to the client (destination).

The Info for this packet should indicate "200 OK" in the case of a normal, successful transfer. You will see that the response is similar to the request, with a series of headers that follow the "200 OK" status code. However, different headers will be used, and the headers will be followed by the requested content.

Examine the common headers such as:

- *i* Server. The kind of server and its capabilities
- *i* Date, Last-Modified. The time of the response and the time the content last changed
- *i* Cache-Control, Expires, Etag. Information about how the response can be cached

2.3.1 Inspecting Unencrypted Transmission

For this exercise, we will need a web application that does not use an encrypted HTTP connection.

In your Lab VM, open the browser and go to http://www.csrflabelgg.com

Make sure that your Wireshark is running and actively capturing traffic from loopback and not ep0s3. (Since the web application is hosted locally on the VM)

On the CSRF Lab Site, log in as admin. With username "admin" and password "seedelgg"

Find a HTTP POST request to page /action/login

S 🗇 🗇 Capturing from Loopback: lo											
	🛛 🖉 💿 📄 🖻 🕅	Q D	< 1	3 11-			0 0 0				
Apple	Apply a display filter <ctri-></ctri->										
blo	Time		Faurea		Deck	Instian	Brohoso	Length Jofe			1
NO.	19 2018-00-24 15:47:58	2854662	127 8 6	1 1	127		TCP	66 80 - 41793	[ETN ACK] Sen-32321	86017 Ack-1810330655	Win=45952 Len
	20 2018-09-24 15:47:58	3854722	127.0.0	.1	127	0.0.1	TCP	66 41792 - 86	[ACK] Seg=1819339655	Ack=3232186018 Win=1	75744 Len=0 T_
	21 2018-09-24 15:48:00.	4012607	127.0.6	.1	127	0.0.1	TCP	74 41794 → 86	[SYN] Seg=1753609592	Win=43690 Len=0 MSS=	65495 SACK PE_
	22 2018-09-24 15:48:00.	4012706	127.0.0	0.1	127	0.0.1	TCP	74 80 - 41794	[SYN, ACK] Seg=26829	81705 Ack=1753609593	Win=43690 Len_
	23 2018-09-24 15:48:00.	4012783	127.0.0	0.1	127.	0.0.1	TCP	66 41794 → 86	[ACK] Seq=1753609593	Ack=2682981786 Win=4	13776 Len=0 TS_
	24 2018-09-24 15:48:00.	4242992	127.0.6	0.1	127	0.0.1	HTTP	641 POST /acti	ion/login HTTP/1.1 (a	pplication/x-www-form	n-urlencoded)
	25 2018-09-24 15:48:00.	4243211	127.0.6	0.1	127	0.0.1	TCP	66 80 - 41794	[ACK] Seq=2682981706	Ack=1753610168 Win=4	14928 Len=0 TS_
-	26 2018-09-24 15:48:00.	6404703_	127.0.0	0.1	127.	0.0.1	HTTP	473 HTTP/1.1 3	802 Found		
	27 2018-09-24 15:48:00.	6404776_	127.0.0	0.1	127.	0.0.1	TCP	66 41794 → 86	[ACK] Seq=1753610168	Ack=2682982113 Win=4	14800 Len=0 TS_
+ 3	28 2018-09-24 15:48:00.	6496722_	127.0.0	0.1	127.	0.0.1	HTTP	470 GET / HTTP	/1.1		and the second
	29 2018-09-24 15:48:00.	6496827_	127.0.0	1.1	127.	0.0.1	TCP	66 80 - 41794	[ACK] Seq=2682982113	Ack=1753610572 Win=4	16080 Len=0 TS_
	30 2018-09-24 15:48:00.	6839209	127.0.0	1.1	127.	0.0.1	нттр	427 HTTP/1.1 3	SU2 Found		
	31 2018-09-24 15:48:00.	0912597	127.0.0	.1	127	0.0.1	HITP	478 GET /activ	(ICV) FORTBERORIATA	Askast75757640004 Minst	7222 10020 75
	22 2010 00 24 15:40:00.	7365434	127.0.0	. 1	127	0.0.1	HTTD	2200 HTTD/1 1 2	[ACK] SEQ-20029024/4	ACK-1753010904 WIII-4	17232 Leff=0 15
	34 2018-09-24 15:48:00	7366164	127 0 6	1 1	127	0.0.1	TCP	66 41794 - 80	[ACK] Seg=1752610984	Ack=2692995699 Win=1	76896 Len=0 T
n Ecom	o 24: 64t butos on utre	/E120 bi	to) 6	At huton	anturod	(E100 bi	tol on interface	0	front and recorded		
+ Ethe	rnet II Src: 80:88:88	00.00.00	(88-80	100162	AA) Det	00.00.	00 00'00'00 (00'	0 . 00 . 00 . 00 . 00]			
> Inte	rnet Protocol Version 4	Src: 12	7.0.0.	1. Dst: 1	7.0.0.1		00_00100100 (001	001001001001007			
+ Tran	smission Control Protoc	ol. Src F	Port: 4	1794. Dst	Port: 80	Sea: 1	753609593, Ack:	2682981706, Len: 575			
Hype	rtext Transfer Protocol	L					,				
▼ HTML	Form URL Encoded: appl	lication/>	-www-f	orm-urlen	oded						
F0	erm item: "elgg_token'	" = "s3dzl	DI7ZQhW	fLwGYde87	"A						
▶ F0	orm item: "elgg_ts" =	"1537818-	473"								
F0	erm item: "username" = '	"admin"									
▶ Fo	orm item: "password" = '	"seedelgg									
0000	00 00 00 00 00 00 00	00 00	0 00 00	8 00 45 0			E.				
	02 73 ba e1 40 00 40 00	5 7f a1 1	7f 00 0	0 01 7f 0	9 .s@						
	00 01 a3 42 00 50 68 85	5 f5 79 1	of eb 0	d 4a 80 1	BB.	≥hy	.J				
	01 56 00 68 00 00 01 01	1 08 0n (00 12 4	f ad 00 1	2 .V.h.		0				
	4t a7 50 4t 53 54 20 2t	61 63 1	4 69 6	t 6e 2f 6	0.P0S	/ acti	on/1				
	of or oy or 20 48 54 54	1 20 42 3	51 ZC 3	1 00 0a 4	s ogin i	111 P/1.	flab				
	55 6c 67 67 20 63 6f 6c	i Ad Aa	5 73 6	5 72 2d 4	elan	om lis	or A				
07	Loophack lay dive conture in	arearease.			exaga.	1103			chote E1 Disalaund E1 (100	0001	Drofile: Dofault

Inspect HTTP request headers and contents

- *i* Do you see the username and password in plain text?
- i What are the implications of sending unencrypted HTTP requests?

Can you replicate the results for the UHCL webmail or your favorite email service?

- *i* If not, why?
- *i* How can you encrypt your web traffic?