

### Monitoring by Wireshark Basic tutorial

Ing. Pedro Escudero

Telf: 0994667184 Mail: <u>pedro.escudero@unach.edu.ec</u> Web: <u>https://www.researchgate.net/profile/Pedro-Escudero-3/research</u>

Abril - 2025



- What is a network trace?
- What is Wireshark?

## .Basic UI

• Some of the most useful parts of the UI.

.Packet Capture

- How do we capture packets?
- .Trace Analysis
- .Individual Packet Analysis
- •Filters
- .Exercises

## Introduction

.Network Traffic Trace

- A recording of the network packets both received by and transmitted from a network interface.
- .What is a pcap file?
  - pcap = Packet Capture
  - File format originally designed for tcpdump/libpcap.
  - Most widely used packet capture format.

- .What is Wireshark?
  - A graphical network packet analyser.
  - Found at <u>http://www.wireshark.org</u>
  - The complete manual is located <u>here</u>.
- .What some are it's uses?
  - Troubleshoot network problems.
  - Learn network protocol internals.
  - Debug protocol/program implementation. Examine network-related security issues.

🔼 te	st.cap					
Eile	<u>E</u> dit <u>V</u> iew <u>G</u> o	<u>Capture</u> <u>Analyze</u> <u>Stat</u>	stics Telephon <u>y T</u> ools <u>I</u> nterna	als <u>H</u> elp		Menu
		🖻 🖬 🗶 🎜 🖁	3   🔍 🗢 🗢 🕹 🕹 3			2 🖪 💥 🔯
Filter:				Expression	Clear Apply	
No.	Time	Source	Destination	Protocol Le	ngth Info	<u>~</u>
	1 0.000000	192.168.0.2	Broadcast	ARP	42 Gratuitous AR	P for 192.168.0.2 (F
	2 0.299139	192.168.0.1	192.168.0.2	NBNS	92 Name query NB	STAT *<00><00><00> <c< td=""></c<>
	3 0.299214	192.168.0.2	192.168.0.1	ICMP	70 Destination u	nreachable (Port unr
	4 1.025659	192.168.0.2	224.0.0.22	IGMP	54 V3 Membership	Report / Join group
	5 1.044366	192.168.0.2	192.168.0.1	DNS	110 Standard quer	y srv _ldaptcp.nbc
	6 1.048652	192.168.0.2	239.255.255.250	SSDP	175 M SEARCH A	Ket List
	7 1.050784	192.168.0.2	192.168.0.1	DNS	at and and ouer	y soa nb10061d.ww004
	8 1.055053	192.168.0.1	192.168.0.2	SSDP	3. HITP/1.1 200	ок
	9 1.082038	192.168.0.2	192.168.0.255	NBNS	110 Registration	NB NB10061D<00>
	10 1.111945	192.168.0.2	192.168.0.1	DNS	87 Standard quer	y A proxyconf.ww004.
	11 1.226156	192.168.0.2	192.168.0.1	TCP	62 ncu-2 > http	[SYN] Seq=0 Win=6424
	12 1.227282	192.168.0.1	192.168.0.2	TCP	60 http > ncu-2	[SYN, ACK] Seq=0 Ack
<			1111			>
I Er	ame 11 · 62 ł	nutes on wire $(4^\circ)$	16 hits) 62 bytes can	tured (496	hits)	
E ET	hernet II.	Src: 192.168.0.2	(00:0b:5d:20:cd:02).	Dst: Netgea	r 2d:75:9a (00:09:	5b:2d:75:9a)
E Tr	ternet Proto	ncol. src: 192.10	8.0.2 (192.168.0.2).	Dst: 192.16	8.0.1 (192.168.0.1	
	ansmission (	Control Protocol.	Src Port: ncu-2 (319	6). Dst Por	t: http (80), Seq:	0. Len: 0
	Source port	ncu-2 (3196)		-2,	at wash (any) and	
	Destination	nort: http (80)			-	
	[stream inde	PX: 5]				
	Sequence num	nher: 0 (relat	ive sequence number)			
	Header lengt	th: 28 hytes	The sequence numbers		Paci	(et Details-
	Elans: 0x02	(SVN)				
	Window size	value: 64240				<b>Sec</b>
-	W11100W 512C	1414C. 04240				
0000	00 09 56 2	d 75 9a 00 0b 5	d 20 cd 02 08 00 45 00	0[-u	]	
0010		8 40 00 80 06 6	1 2C CU 48 UU U2 CU 48 5 F8 00 00 00 00 70 07	5 .U.HG	a,	
0020	fa f0 27 e		5 b4 01 01 04 02	· · · · · · · · · · · · · · · · · · ·		
1000000000	1942 1942 1944		Statute in ord The Cold Street with Cold The		Pa	icket Bytes
Eil	e: "C:/test.cap" 14	KB 00:00:02	Packets: 120 Displayed: 120 Ma	arked: O Load time:	0:00.000 Profile: D	efault

- .File -> Open
  - Opens a packet capture file.
- .View -> Time Display Format
  - Change the format of the packet timestamps in the packet list pane.
  - Switch between absolute and relative timestamps.
  - Change level of precision.
- .View -> Name Resolution
  - Allow wireshark to resolve names from addresses at different protocol layers.

- .Capture -> Interfaces
  - Available network interfaces for capture.
  - Total packets per interface.
  - Packet rate per interface.

- .Capture -> Options
  - Set various capture parameters.
- . Promiscous mode
  - On record all packets reaching the interface.
  - Off record only those packets directed to the host.

	Wireshark: Captu	re Interfaces				- +
Device	Description	IP	Packets	Packets/s		Stop
🗩 em 1		192.168.0.11	55	0	😂 Start	Options
p37p1		unknown	0	0	Start Start	Options
🔊 any Pseudo-	device that captures on all interfaces	unknown	71	0	Start Start	Options
🛃 lo		127.0.0.1	16	0	Start	Options
Help						× Close

			Wir	eshark: Capture Optio	ns – +				
Capture									
Interface:	m1				~				
IP address: 1	192.168.0	.11, fe	80::2	218:f3ff:fe2e:4256					
Link-layer he	ader type	: Ethe	Wireless Settings						
🗹 Capture p	oackets in	promi	scuo	us mode	Buffer size: 1 💭 megabyte(s				
Capture p	backets in	monit	or m	ode					
Capture p	oackets in	pcap-	ng fo	ormat					
Limit each	n packet t	o 655	535	bytes					
👹 Capture	Filter:				✓ Compile BPF				
Capture File (	s)				Display Options				
File:				Browse	Update list of packets in real tim				
🗌 Use multi	ple files								
🖾 Next file (	every	1		🗘 megabyte(s) 👘	🕼 Automatic scrolling in live captur				
🗇 Next file (	every	1		C minute(s)	🖉 Llida captura infa dialog				
🗍 Ring buffe	er with	2	-	files	Mide capture into diatog				
Stop capi	ture after	1		file(s)	Name Resolution				
Stop Capture				<u>N</u>	Enable MAC name resolution				
			181.	andvat(c)					
		packet(s)		Jacker(s)	Enable network name resolution				
		megabyte		megabyte(s) +					
	🗆 after 🛛 1			minute(c)	Enable transport name resolution				

.Analyze -> Follow TCP Stream

• Applies a filter to follow a single tcp conversation within the trace.

- Displays the reassembiled data section of each packet in the conversation.
- Useful for debugging or analyzing any TCP based application layer protocol.
  - HTTP, FTP, SSH, LDAP, SMTP, etc.

### .Statistics -> Protocol Hierarchy

- Presents descriptive statistics per protocol.
- Useful for determining the types, amounts, and relative proportions of protocols within a trace.

Wireshark: Protocol Hierarchy Statistics				+ >	ĸ
Display filter: none					
Protocol	% Packets	Packets	% Bytes	В	8
▼ Frame	100.00 %	106117	100.00 %	7	l
▼ Ethernet	100.00 %	7	l		
Address Resolution Protocol	1.70 %	1805	0.14 %		l
▼ Logical-Link Control	0.60 %	632	0.06 %		l
Spanning Tree Protocol	0.57 %	609	0.05 %		l
▼ Cisco Discovery Protocol	0.02 %	20	0.01 %		l
Malformed Packet	0.02 %	20	0.01 %		l
Logical-Link Control Basic Format XID	0.00 %	3	0.00 %		l
▼ Internet Protocol Version 6	0.59 %	625	0.11 %		l
▼ User Datagram Protocol	0.42 %	450	0.09 %		4
DHCPv6	0.20 %	210	0.04 %		
Domain Name Service	0.19 %	204	0.04 %		
Hypertext Transfer Protocol	0.03 %	36	0.01 %		
Internet Control Message Protocol v6	0.16 %	175	0.02 %		
▼ Internet Protocol Version 4	97.07 %	103013	99.69 %	7	
▼ User Datagram Protocol	3.69 %	3912	1.00 %		
Domain Name Service	2.42 %	2573	0.60 %		
Dropbox LAN sync Discovery Protocol	0.46 %	486	0.11 %		
Bootstrap Protocol	0.46 %	484	0.22 %		
Common Unix Printing System (CUPS) Browsing Protocol	0.04 %	39	0.01 %		
▼ NetBIOS Datagram Service	0.05 %	55	0.02 %		
▼ SMB (Server Message Block Protocol)	0.05 %	55	0.02 %		2
				>	-
(2) Help			× Clo	se	

### .Statistics -> Conversations

 Generates descriptive statistics about each conversation for each protocol in the trace.

Fibre Chamler 11	11 14. 400	11 90.07	A JACK NC	110001	201	101.1525	onest rang ODI	14014 000 110
		Et	hernet Conve	sations				
Address A	Address B		Packets	Bytes		Packets A→B	Bytes A→B	Packets A←B
Brocade C_ef:8b:00	Broadcast		753	59 2	085	753	59 280	0
Spanning-tree-(for-bridges)_00	Cisco_ed:4e:5	9	609	38 9	976	0	0	609
Dell_77.19.25	Broadcast		486	29	343	486	29 343	0
Dell_77:19:25	IPv6mcast_00	01:00:02	16	2	352	16	2 352	0
Dell_4524.bb	Broadcast		2	120		2	120	0
Cadmus Co_e5:ac:58	Cisco-Li_c1:d1	.f9	101 328	77 500	345	42 867	5 114 212	58 461
Dell_9e:44:b0	Broadcast		165	26 (	054	165	26 054	0
QuantaCo_8f:42:cd	Broadcast		40	2 /	400	40	2 400	0
Dell_d5:c7:3b	Broadcast		67	4 (	525	67	4 625	0
IntelCor_3d:19:63	Broadcast		7		420	7	420	0
<								3
Vame resolution			🗆 Lir	nit to dis	play	filter		

### .Statistics -> Flow Graph

- Generates a sequence graph for the selected traffic.
- Useful for understanding seq. and ack. calculations.



## • Packet Capture

.Interface selection

- Capture -> Interfaces
  - Select the interface from which to capture packets.
    - any captures from all interfaces
    - lo captures from the loopback interface (i.e. from localhost)
  - Set the desired capture parameters under the options menu.

•Start Capture

- Click the start button next to the desired interface.
- Captured traffic will be displayed in the packet list pane.

## • Packet Capture

.Stop Capture

• Select Capture -> Stop

Saving Capture

- Once the capture has been stopped select File -> Save As.
- From the save dialog you can specify file type and which packets to save via the packet range menu.

## • Trace Analysis

11 t	est.cap				
Eile	<u>E</u> dit <u>V</u> iew <u>G</u> o	<u>Capture Analyze Stal</u>	tistics Telephon <u>y T</u> ools <u>I</u> nterna	als <u>H</u> elp	Menu
		i 🖻 🖬 🗶 🖉 (	트   🤇 🗢 🇢 🕉 🛽		
Filte	r:		~	Expression Cle	ar Apply
No.	Time	Source	Destination	Protocol Leng	th Info 🧭
	1 0.000000	192.168.0.2	Broadcast	ARP	42 Gratuitous ARP for 192.168.0.2 (F
	2 0.299139	192.168.0.1	192.168.0.2	NBNS	92 Name query NBSTAT *<00><00><00> <c< td=""></c<>
	3 0.299214	192.168.0.2	192.168.0.1	ICMP	70 Destination unreachable (Port unr
	4 1.025659	192.168.0.2	224.0.0.22	IGMP	54 V3 Membership Report / Join group
	5 1.044366	192.168.0.2	192.168.0.1	DNS 1	10 Standard query SRV _ldaptcp.nbc
	6 1.048652	192.168.0.2	239.255.255.250	SSDP 1	
	7 1.050784	192.168.0.2	192.168.0.1	DNS	Standard query SOA nb10061d.ww004
	8 1.055053	192.168.0.1	192.168.0.2	SSDP 3	S HITP/1.1 200 OK
	9 1.082038	192.168.0.2	192.168.0.255	NBNS 1	10 Registration NB NB10061D<00>
	10 1.111945	192.168.0.2	192.168.0.1	DNS	87 Standard query A proxyconf.ww004.
	11 1.226156	192.168.0.2	192.168.0.1	TCP	62 ncu-2 > http [SYN] Seq=0 Win=6424
	12 1.227282	192.168.0.1	192.168.0.2	TCP	60 http > ncu-2 [SYN, ACK] Seq=0 Ack ∨
$\leq$			III		
ΞF	rame 11: 62	bytes on wire (4)	96 bits), 62 bytes cap	tured (496 bi	ts)
E E	thernet II,	src: 192.168.0.2	(00:0b:5d:20:cd:02), I	Dst: Netgear_	2d:75:9a (00:09:5b:2d:75:9a)
ÐI	nternet Prot	ocol, Src: 192.10	68.0.2 (192.168.0.2), (	Dst: 192.168.	0.1 (192.168.0.1)
ΘT	ransmission	Control Protocol	, Src Port: ncu-2 (319	6), Dst Port:	http (80), Seq: 0, Len: 0
	Source port	: ncu-2 (3196)			
	Destination	port: http (80)			
	[Stream ind	ex: 5]			
	Sequence nu	mber: 0 (relat	tive sequence number)		Decket Details
	Header leng	th: 28 bytes			Packet Details
Ē	Flags: 0x02	(SYN)			
- ~ ~	Window size	value: 64240			~
000	0 00 00 56	2d 75 02 00 06 5	d 30 cd 03 08 00 45 00	) <u>Fu</u>	
000	0 00 09 00 0	18 40 00 80 06 6	10 20 C0 02 08 00 41 00	,ч ко.на	
002	0 00 01 0c :	7c 00 50 3c 36 9	95 f8 00 00 00 00 70 02	2 .P<6	
003	0 fa f0 27 e	e0 00 00 02 04  0	)5 b4 01 01 04 02		D. L.L.D.
					Packet Bytes
F	File: "C:/test.cap" 14	KB 00:00:02	Packets: 120 Displayed: 120 Ma	rked: 0 Load time: 0:	00.000 Profile: Default

## • Trace Analysis

### Packet list

- Displays all of the packets in the trace in the order they were recorded.
- Columns
  - Time the timestamp at which the packet crossed the interface.
  - Source the originating host of the packet.
  - Destination the host to which the packet was sent.
  - Protocol the highest level protocol that Wireshark can detect.
  - Lenght the lenght in bytes of the packet on the wire.
  - Info an informational message pertaining to the protocol in the protocol column.

## • Trace Analysis

### Packet list

- Default Coloring
  - Gray TCP packets
  - Black with red letters TCP Packets with errors
  - Green HTTP Packets
  - Light Blue UDP Packets
  - Pale Blue ARP Packets
  - Lavender ICMP Packets
  - Black with green letters ICMP Packets with errors
- Colorings can be changed under View -> Coloring Rules

## Individual Packet Analysis

11 t	est.c	ap															_	
Eile	Edit	⊻iew	Go	<u>C</u> apture	Analyze	<u>S</u> tatis	stics Tel	ephon <u>y</u>	<u>T</u> ools	Internals	Help	-	/		Me	nu		
		94 94			<b>7</b> ×	2	3   9	🗢 🔿	-	<b>T L</b>			~~ •	1 1994		H		
Filte	r:									~	Expression	Clear	Apply					
No.		Time		Source			De	stination			Protocol	Length	Info					~
	1	0.000	0000	192.	168.0.	2	В	roadca	st		ARP	42	Gratuito	us AR	P for	192.	168.0.3	2 (F
	2	0.299	)139	192.	168.0.	.1	1	92.168	.0.2		NBNS	92	Name que	ry NB	STAT '	*<00>	<00><0	0><0
	3	0.299	9214	192.	168.0.	2	1	92.168	.0.1		ICMP	70	Destinat	ion u	Inreact	nable	(Port	unr
	4	1.025	659	192.	168.0.	2	2	24.0.0	.22		IGMP	54	V3 Membe	rship	Repor	rt /	Join g	roup
	5	1.044	366	192.	168.0.	.2	1	92.168	.0.1		DNS	110	Standard	quer	y SRV	_lda	ptcp	. nbç
	6	1.048	3652	192.	168.0.	.2	2	39.255	.255.	250	SSDP	175	MEEARCH	Pac	<b>ike</b>	C L	IST	
	7	1.050	)784	192.	168.0.	. 2	1	92.168	.0.1		DNS		atandard	quer	y SOA	nb10	061d.w	w004
	8	1.055	053	192.	168.0.	1	1	92.168	.0.2		SSDP	3-	HIP/1.1	200	ок			
	9	1.082	2038	192.	168.0.	.2	1	92.168	.0.25	5	NBNS	110	Registra	tion	NB NB1	L0061	D<00>	
	10	1.111	.945	192.	168.0.	.2	1	92.168	.0.1		DNS	87	Standard	quer	у Арг	тохус	onf.ww	004.
	11	1.226	5156	192.	168.0.	2	1	92.168	.0.1		TCP	62	ncu-2 >	http	[SYN]	Seq=	0 Win=	6424
	12	1.227	282	192.	168.0.	1	1	92.168	.0.2		TCP	60	http > n	cu-2	[SYN,	ACK]	Seq=0	Ack 🕶
<							1111					1						>
ΞF	rame	e 11:	62 k	ovtes d	on wir	e (49	6 bits	), 62	bytes	captu	red (490	bits	)					~
E E	ther	net I	Ι, 3	Src: 19	92.168	.0.2	(00:0b	:5d:20	:cd:(	)2), Ds	t: Netge	ar_2d	:75:9a (0	0:09:	5b:2d:	75:9	a)	
ÐI	nter	net P	roto	ocol, s	5rc: 1	92.16	8.0.2	(192.1	68.0.	2), Ds	t: 192.1	.68.0.1	L (192.16	8.0.1	)			
ΘT	rans	missi	on c	Contro	Prot	ocol,	Src P	ort: n	cu-2	(3196)	, Dst Po	ort: ht	ttp (80),	Seq:	0, Le	en: O		
	SOL	ince p	ort	ncu-2	2 (319	6)												
	Des	tinat	ion	port:	http	(80)							-					
	[St	ream	inde	ex: 5]										-				
	Sec	quence	nun	nber: (	0 (	relat	ive se	quence	numb	per)				-		D		
	Неа	ider 1	engt	:h: 28	bytes			al no strong octobe						aci	κετ	De	etai	IS
i ä	Fla	ngs: 0	x02	(SYN)														
	Wir	ndow s	ize	value:	: 6424	0												~
000	~ ~	A AA	<b>5</b>	1 75 0						45 00	F 32	2.7	-					00000
000	0 0	0 09	50 Z 18 A	a 75 9 8 40 0		00 00 06 61		1 02 00	5 00	45 00	[-u.	•• •						
002	ŏŏ	0 01	0 < 7	C 00 5	0 3 c 3	36 95	5 f8 00		00 0	70 02		<6						
003	0 Ē	a fo	27 e	0 00 0	0 02 0	04 05	5 b4 01	. 01 04	1 02	10-10-00-T-T-				-	-		-	
														Pā	ack	et	Byt	es
F	File: "C	:/test.ca	p" 14	KB 00:00:0	02		Packet	s: 120 Dis	played:	120 Marke	d: O Load tim	ne: 0:00.0	100 F	Profile: [	Default			4

## Individual Packet Analysis

### .Packet Details

- Detailed information about the currently selected packet is displayed in the packet details pane.
- All packet layers are displayed in the tree menu.
- Any portion of any layer can be exported via a right click and selecting Export Selected Packet Bytes

.Packet Bytes

- Displays the raw packet bytes.
- The selected packet layer is highlighted.

- Packets captures usually contain many packets irrelevant to the specific analysis task.
- To remove these packets from display or from the capture Wireshark provides the ability to create filters.
- Filters are evaluted against each individual packet.
- . Boolean expresions dealing with packet properties.
- Supports regular expressions.
- Can either be manually constructed, composed via the Expressions menu or composed based on a selected packet's properties.

#### .Expressions Menu

- Field name selects the packet property.
- Relation selects the boolean test.
- Predefined values common values against which the selected packet property is tested.
- Value Arbitrary Textual or Numeric value against which the selected packet property is tested.



### .Compound Filters

- Filters can be composed of multiple tests joined with boolean connectives.
  - && logical conjuction (i.e. AND)
  - || logical disjunction (i.e OR)
  - ! logical negation (i.e. NOT)
- Supports the order of operations.
- .Regular Expressions
  - Fields can be evaluated against a regular expression using the "matches" test.
  - Uses <u>Perl regex syntax</u>.

.Filter Text Box

- Green valid filter
- Red invalid filter
- Yellow may produce unexpected results
- .Packet based filters
  - Filters can be constructed on the basis of individual packets by right clicking on a packet and selecting either:
    - Prepare as filter creates a filter.
    - Apply as filter creates a filter and applies it to the trace.
    - Follow TCP Stream creates a filter from a TCP packet's stream number and applies it to the trace.

.Filter examples

- http.request Display all HTTP requests.
- http.request || http.response Display all HTTP request and responses.
- ip.addr = = 127.0.0.1 Display all IP packets whose source or destination is localhost.
- tcp.len < 100 Display all TCP packets whose data length is less than 100 bytes.
- http.request.uri matches "(gif)\$" Display all HTTP requests in which the uri ends with "gif".
- dns.query.name = "www.google.com" Display all DNS queries for "www.google.com".

# Exercises

### **Exercise 1: Basic Packet Capture and Analysis**

**Objective:** Learn to capture network traffic and understand packet structures.

- **1. Step 1:** Open Wireshark and start a packet capture on the network interface that has internet connectivity.
- **2. Step 2:** Browse to a website (e.g., <u>http://example.com</u>) to generate some traffic.
- **3. Step 3:** Stop the capture after the page fully loads.

Tasks:

- Filter HTTP packets and identify the GET request for the page.
- Analyze the TCP 3-way handshake (SYN, SYN-ACK, ACK) for the connection.
- Find the server's IP address and the port used for the HTTP service.
- Take a screenshot of the packet structure and identify fields such as the source and destination IP addresses, source and destination ports, and sequence numbers.

### **Exercise 2: Analyzing DNS Requests**

**Objective:** Understand how DNS requests are handled in network communication.

- **1. Step 1:** Clear your DNS cache to ensure all DNS requests are fresh.
- 2. Step 2: Start Wireshark and begin capturing on your active network interface.
- **3. Step 3:** In a browser, visit several websites (e.g., open Google, Yahoo, etc.) and then stop the capture.

#### Tasks:

- Use a filter to isolate DNS traffic (dns in the filter).
- Identify at least one DNS query for each website and find the corresponding response.
- Note the response time and TTL (time-to-live) for each DNS query.
- Examine the DNS response and identify if any DNS queries returned multiple IP addresses.

## **Exercise 3: Monitoring HTTP and HTTPS Traffic**

**Objective:** Differentiate between HTTP and HTTPS traffic and understand how encryption affects packet analysis.

- 1. Step 1: Start capturing packets in Wireshark on your network interface.
- Step 2: In a browser, visit a website that uses HTTP (e.g., <u>http://neverssl.com</u>) and another that uses HTTPS (e.g., <u>https://example.com</u>).
- **3. Step 3:** Stop the capture.

#### Tasks:

- Use a filter to view only HTTP and HTTPS packets.
- Compare the packets from the HTTP and HTTPS connections. Note which fields are visible and inspectable in HTTP vs. HTTPS.
- Identify any unencrypted data in the HTTP packets, such as the contents of the GET requests.
- Try to locate the TLS handshake in the HTTPS packets. Identify fields such as the ServerHello and Certificate messages.