

MODULO 5- REVERSING DE REDES Y PROTOCOLOS

!error Gdn	: Display error	eax (64 bits)
!address	: Display information about memory	eax (32 bits)
~	: List threads	ax (16 bits)
bl	: List breakpoints	ah (8 bits)
bc	: Cancel breakpoints	al (8 bits)
be	: Enable breakpoints	
bd	: Disable breakpoints	
bp [Addr]	: Set breakpoint at the address	
bm SymPattern	: Set breakpoint at the symbol	
ba [riwle] Addr	: Set breakpoint on Access	
k	: Display call stack	
z	: Dump all registers	
u	: Disassemble	
dN	: Display where N:	
	a: ascii chars u: Unicode char	
	b: byte + ascii w: word	
	M: word + ascii d: dword	
	c: dword + ascii q: qword	
	b: bin + byte d: bin + dword	
eN Addr Value	: Edit memory	
.writemem f A S	: Dump memory	
	f: file name	
	A: Address	
	S: Size (Lx)	

dec hex char	dec hex char	dec hex char	dec hex char
0 0x00 NUL	32 0x20 SPACE	64 0x40 @	96 0x60 a
1 0x01 SOH	33 0x21 !	65 0x41 A	97 0x61 A
2 0x02 STX	34 0x22 "	66 0x42 B	98 0x62 b
3 0x03 ETX	35 0x23 #	67 0x43 C	99 0x63 c
4 0x04 EOT	36 0x24 \$	68 0x44 D	100 0x64 d
5 0x05 ENQ	37 0x25 %	69 0x45 E	101 0x65 e
6 0x06 ACK	38 0x26 &	70 0x46 F	102 0x66 f
7 0x07 BEL	39 0x27 *	71 0x47 G	103 0x67 g
8 0x08 BS	40 0x28 (72 0x48 H	104 0x68 h
9 0x09 TAB	41 0x29)	73 0x49 I	105 0x69 i
10 0x0A LF	42 0x2A *	74 0x4A J	106 0x6A j
11 0x0B VT	43 0x2B +	75 0x4B K	107 0x6B k
12 0x0C FF	44 0x2C ,	76 0x4C L	108 0x6C l
13 0x0D CR	45 0x2D -	77 0x4D M	109 0x6D m
14 0x0E SO	46 0x2E .	78 0x4E N	110 0x6E n
15 0x0F SI	47 0x2F /	79 0x4F O	111 0x6F o
16 0x10 DLE	48 0x30 0	80 0x50 P	112 0x70 p
17 0x11 DC1	49 0x31 1	81 0x51 Q	113 0x71 q
18 0x12 DC2	50 0x32 2	82 0x52 R	114 0x72 r
19 0x13 DC3	51 0x33 3	83 0x53 S	115 0x73 s
20 0x14 DC4	52 0x34 4	84 0x54 T	116 0x74 t
21 0x15 NAK	53 0x35 5	85 0x55 U	117 0x75 u
22 0x16 SHX	54 0x36 6	86 0x56 V	118 0x76 v
23 0x17 STB	55 0x37 7	87 0x57 W	119 0x77 w
24 0x18 CAN	56 0x38 8	88 0x58 X	120 0x78 x
25 0x19 EM	57 0x39 9	89 0x59 Y	121 0x79 y
26 0x1A SUB	58 0x3A :	90 0x5A Z	122 0x7A z
27 0x1B ESC	59 0x3B ;	91 0x5B [123 0x7B {
28 0x1C FS	60 0x3C <	92 0x5C \	124 0x7C
29 0x1D GS	61 0x3D =	93 0x5D]	125 0x7D ~
30 0x1E RS	62 0x3E >	94 0x5E ^	126 0x7E _
31 0x1F US	63 0x3F ?	95 0x5F `	127 0x7F DEL

[IDA Pro shortcuts]			
Navigation:			
Enter	: Jump to operand	ESC	: Jump to previous position
G	: Go to address	Ctrl+L	: Jump by name
Ctrl+F	: Jump to function	X	: xref
Ctrl+E	: Jump to entry point		
Search			
Alt+C	: Next code	Ctrl+B	: Next data
Alt+I	: Immediate value	Ctrl+I	: Next immediate value
Alt+T	: Text	Ctrl+T	: Next text
Alt+B	: Sequence of bytes	Ctrl+B	: Next sequence of bytes
Graphing			
F12	: Flow chart	Ctrl+F12	: Function calls
Subviews			
Shift+F4	: Name	Shift+F3	: Functions
Shift+F12	: Strings	Shift+F7	: Segments
Debugger			
F9	: Start	Ctrl+F2	: Stop process
F7	: Step into	F8	: Step over
Ctrl+F7	: Run until return	Ctrl+Alt+B	: List breakpoints
Other			
C	: Code	D	: Data
U	: Undefined	N	: Rename
Shift+;	: Enter comment	/	: Enter repeatable comment
P	: Create function	Alt+F	: Edit function
E	: Set function end	Y	: Declare function type
M	: Member enumeration	Shift+F2	: Run script
[Immunity Debugger shortcuts]			
F2	: Set breakpoint	F9	: run
F7	: Step into	F8	: Step over
Ctrl+F9	: Execute till ret	F12	: Pause
Alt+B	: Open breakpoint w/	Alt+C	: Open CPU window
Alt+E	: Open module window	Alt+L	: Open log window
Alt+M	: Open memory window	Alt+O	: Open option window

Máster en Análisis de Malware, Reversing y Bug Hunting



Ramon Gonzalez Gaztelupe

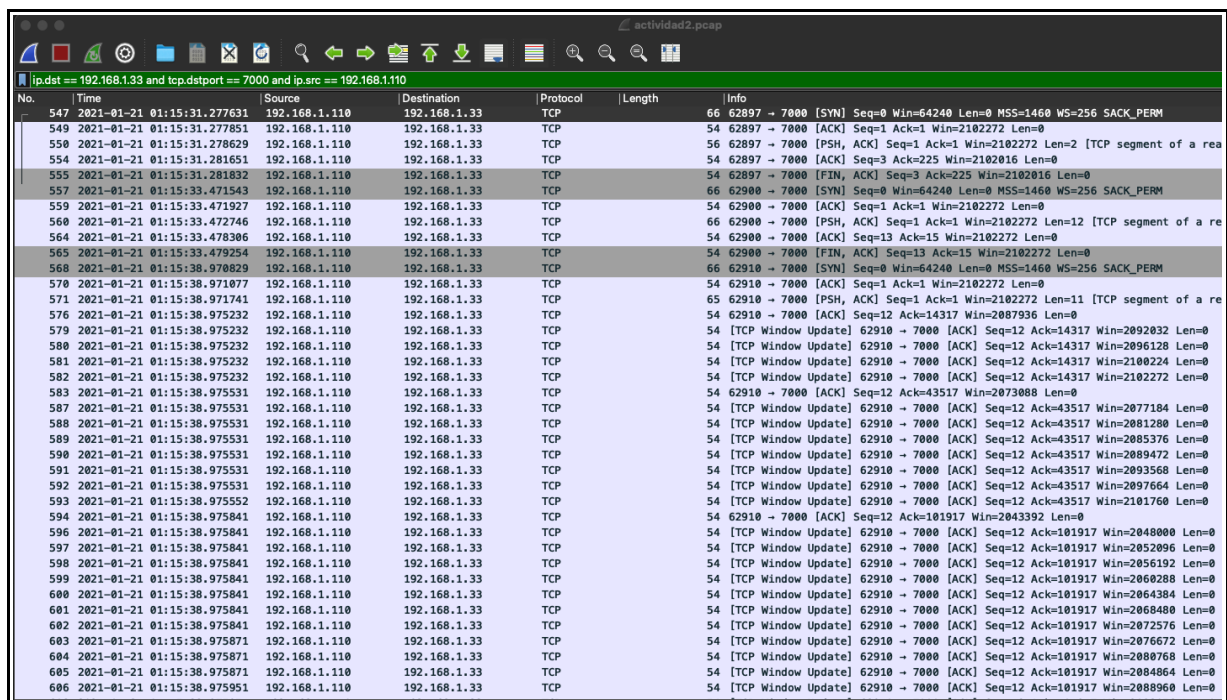
1.- Enunciado

El objetivo de esta actividad individual es analizar con Wireshark el tráfico de una aplicación desconocida a partir de una captura de tráfico de la misma. En particular, hay que analizar el tráfico incluido en el fichero actividad2.pcap, que acompaña a este enunciado. Esa captura de tráfico se ha realizado en la misma máquina (192.168.1.33) donde se ejecuta el servidor de la aplicación a analizar y que corre en el puerto TCP/7000. El fichero de captura también incluye otro tipo de tráfico, que no está relacionado con la aplicación, por lo que se deberían utilizar los filtros de visualización de Wireshark para encontrar el tráfico de la aplicación

El objetivo de la práctica es estudiar el protocolo empleado por la aplicación, identificar el tipo de mensajes intercambiados por el cliente y el servidor, así como intentar interpretar el contenido y significado de los mismos, e idealmente obtener la información intercambiada. Opcionalmente se propone intentar identificar el protocolo exacto que está siendo empleado por la aplicación. En la memoria de la actividad se deben incluir capturas de pantalla de Wireshark para ilustrar la especificación del protocolo de la aplicación, así como capturas de pantalla de la información intercambiada entre el cliente y el servidor.

2.- Wireshark

Filtraremos el tráfico según los parámetros que se nos ofrece para obtener una vista mas clara de la comunicación con el servidor:



The screenshot shows the Wireshark interface with a packet capture filter applied: `ip.dst == 192.168.1.33 and tcp.dstport == 7000 and ip.src == 192.168.1.110`. The packet list shows 24 packets, all of which are TCP segments from 192.168.1.110 to 192.168.1.33 on port 7000. The packets include SYN, ACK, PSH, FIN, and window update messages, indicating a complete TCP connection and data transfer.

No.	Time	Source	Destination	Protocol	Length	Info
547	2021-01-21 01:15:31.277631	192.168.1.110	192.168.1.33	TCP	66	62897 → 7000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
549	2021-01-21 01:15:31.277851	192.168.1.110	192.168.1.33	TCP	54	62897 → 7000 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
550	2021-01-21 01:15:31.278629	192.168.1.110	192.168.1.33	TCP	56	62897 → 7000 [PSH, ACK] Seq=1 Ack=1 Win=2102272 Len=2 [TCP segment of a re
554	2021-01-21 01:15:31.281651	192.168.1.110	192.168.1.33	TCP	54	62897 → 7000 [ACK] Seq=3 Ack=225 Win=2102016 Len=0
555	2021-01-21 01:15:31.281832	192.168.1.110	192.168.1.33	TCP	54	62897 → 7000 [FIN, ACK] Seq=3 Ack=225 Win=2102016 Len=0
557	2021-01-21 01:15:33.471543	192.168.1.110	192.168.1.33	TCP	66	62900 → 7000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
559	2021-01-21 01:15:33.471927	192.168.1.110	192.168.1.33	TCP	54	62900 → 7000 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
560	2021-01-21 01:15:33.472746	192.168.1.110	192.168.1.33	TCP	66	62900 → 7000 [PSH, ACK] Seq=1 Ack=1 Win=2102272 Len=12 [TCP segment of a re
564	2021-01-21 01:15:33.478306	192.168.1.110	192.168.1.33	TCP	54	62900 → 7000 [ACK] Seq=13 Ack=15 Win=2102272 Len=0
565	2021-01-21 01:15:33.479254	192.168.1.110	192.168.1.33	TCP	54	62900 → 7000 [FIN, ACK] Seq=13 Ack=15 Win=2102272 Len=0
568	2021-01-21 01:15:38.970829	192.168.1.110	192.168.1.33	TCP	66	62910 → 7000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
570	2021-01-21 01:15:38.971077	192.168.1.110	192.168.1.33	TCP	54	62910 → 7000 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
571	2021-01-21 01:15:38.971741	192.168.1.110	192.168.1.33	TCP	65	62910 → 7000 [PSH, ACK] Seq=1 Ack=1 Win=2102272 Len=11 [TCP segment of a re
576	2021-01-21 01:15:38.975232	192.168.1.110	192.168.1.33	TCP	54	62910 → 7000 [ACK] Seq=12 Ack=14317 Win=2087936 Len=0
579	2021-01-21 01:15:38.975232	192.168.1.110	192.168.1.33	TCP	54	[TCP Window Update] 62910 → 7000 [ACK] Seq=12 Ack=14317 Win=2092032 Len=0
580	2021-01-21 01:15:38.975232	192.168.1.110	192.168.1.33	TCP	54	[TCP Window Update] 62910 → 7000 [ACK] Seq=12 Ack=14317 Win=2096128 Len=0
581	2021-01-21 01:15:38.975232	192.168.1.110	192.168.1.33	TCP	54	[TCP Window Update] 62910 → 7000 [ACK] Seq=12 Ack=14317 Win=2100224 Len=0
582	2021-01-21 01:15:38.975232	192.168.1.110	192.168.1.33	TCP	54	[TCP Window Update] 62910 → 7000 [ACK] Seq=12 Ack=14317 Win=2102272 Len=0
583	2021-01-21 01:15:38.975531	192.168.1.110	192.168.1.33	TCP	54	62910 → 7000 [ACK] Seq=12 Ack=43517 Win=2073008 Len=0
587	2021-01-21 01:15:38.975531	192.168.1.110	192.168.1.33	TCP	54	[TCP Window Update] 62910 → 7000 [ACK] Seq=12 Ack=43517 Win=2077184 Len=0
588	2021-01-21 01:15:38.975531	192.168.1.110	192.168.1.33	TCP	54	[TCP Window Update] 62910 → 7000 [ACK] Seq=12 Ack=43517 Win=2081280 Len=0
589	2021-01-21 01:15:38.975531	192.168.1.110	192.168.1.33	TCP	54	[TCP Window Update] 62910 → 7000 [ACK] Seq=12 Ack=43517 Win=2085376 Len=0
590	2021-01-21 01:15:38.975531	192.168.1.110	192.168.1.33	TCP	54	[TCP Window Update] 62910 → 7000 [ACK] Seq=12 Ack=43517 Win=2089472 Len=0
591	2021-01-21 01:15:38.975531	192.168.1.110	192.168.1.33	TCP	54	[TCP Window Update] 62910 → 7000 [ACK] Seq=12 Ack=43517 Win=2093568 Len=0
592	2021-01-21 01:15:38.975531	192.168.1.110	192.168.1.33	TCP	54	[TCP Window Update] 62910 → 7000 [ACK] Seq=12 Ack=43517 Win=2097664 Len=0
593	2021-01-21 01:15:38.975552	192.168.1.110	192.168.1.33	TCP	54	[TCP Window Update] 62910 → 7000 [ACK] Seq=12 Ack=43517 Win=2101760 Len=0
594	2021-01-21 01:15:38.975841	192.168.1.110	192.168.1.33	TCP	54	62910 → 7000 [ACK] Seq=12 Ack=101917 Win=2043392 Len=0
596	2021-01-21 01:15:38.975841	192.168.1.110	192.168.1.33	TCP	54	[TCP Window Update] 62910 → 7000 [ACK] Seq=12 Ack=101917 Win=2048000 Len=0
597	2021-01-21 01:15:38.975841	192.168.1.110	192.168.1.33	TCP	54	[TCP Window Update] 62910 → 7000 [ACK] Seq=12 Ack=101917 Win=2052096 Len=0
598	2021-01-21 01:15:38.975841	192.168.1.110	192.168.1.33	TCP	54	[TCP Window Update] 62910 → 7000 [ACK] Seq=12 Ack=101917 Win=2056192 Len=0
599	2021-01-21 01:15:38.975841	192.168.1.110	192.168.1.33	TCP	54	[TCP Window Update] 62910 → 7000 [ACK] Seq=12 Ack=101917 Win=2060288 Len=0
600	2021-01-21 01:15:38.975841	192.168.1.110	192.168.1.33	TCP	54	[TCP Window Update] 62910 → 7000 [ACK] Seq=12 Ack=101917 Win=2064384 Len=0
601	2021-01-21 01:15:38.975841	192.168.1.110	192.168.1.33	TCP	54	[TCP Window Update] 62910 → 7000 [ACK] Seq=12 Ack=101917 Win=2068480 Len=0
602	2021-01-21 01:15:38.975841	192.168.1.110	192.168.1.33	TCP	54	[TCP Window Update] 62910 → 7000 [ACK] Seq=12 Ack=101917 Win=2072576 Len=0
603	2021-01-21 01:15:38.975871	192.168.1.110	192.168.1.33	TCP	54	[TCP Window Update] 62910 → 7000 [ACK] Seq=12 Ack=101917 Win=2076672 Len=0
604	2021-01-21 01:15:38.975871	192.168.1.110	192.168.1.33	TCP	54	[TCP Window Update] 62910 → 7000 [ACK] Seq=12 Ack=101917 Win=2080768 Len=0
605	2021-01-21 01:15:38.975871	192.168.1.110	192.168.1.33	TCP	54	[TCP Window Update] 62910 → 7000 [ACK] Seq=12 Ack=101917 Win=2084864 Len=0
606	2021-01-21 01:15:38.975951	192.168.1.110	192.168.1.33	TCP	54	[TCP Window Update] 62910 → 7000 [ACK] Seq=12 Ack=101917 Win=2088960 Len=0

Si filtramos el puerto 7000 nos devuelve un único nodo de comunicación:

actividad2.pcap

tcp.port == 7000

No.	Time	Source	Destination	Protocol	Length	Info
547	2021-01-21 01:15:31.277631	192.168.1.110	192.168.1.33	TCP	66	62897 → 7000 [FIN, ACK] Seq=62897 Win=0 Len=0
548	2021-01-21 01:15:31.277655	192.168.1.33	192.168.1.110	TCP	66	7000 → 62897 [ACK] Seq=62897 Win=0 Len=0
549	2021-01-21 01:15:31.277851	192.168.1.110	192.168.1.33	TCP	54	62897 → 7000 [ACK] Seq=62897 Win=0 Len=0
550	2021-01-21 01:15:31.278629	192.168.1.110	192.168.1.33	TCP	56	62897 → 7000 [PSH, ACK] Seq=62897 Win=0 Len=2 [TCP segment of a reassembled PDU]
551	2021-01-21 01:15:31.278696	192.168.1.33	192.168.1.110	TCP	54	7000 → 62897 [ACK] Seq=62897 Win=0 Len=0
552	2021-01-21 01:15:31.281381	192.168.1.33	192.168.1.110	TCP	111	7000 → 62897 [PSH, ACK] Seq=62897 Win=0 Len=57 [TCP segment of a reassembled PDU]
553	2021-01-21 01:15:31.281439	192.168.1.33	192.168.1.110	TCP	220	7000 → 62897 [FIN, PSH, ACK] Seq=58 Ack=3 Win=64256 Len=166 [TCP segment of a reassembled PDU]
554	2021-01-21 01:15:31.281651	192.168.1.110	192.168.1.33	TCP	54	62897 → 7000 [ACK] Seq=62897 Win=0 Len=0
555	2021-01-21 01:15:31.281832	192.168.1.110	192.168.1.33	TCP	54	62897 → 7000 [ACK] Seq=62897 Win=0 Len=0
556	2021-01-21 01:15:31.281843	192.168.1.33	192.168.1.110	TCP	54	7000 → 62897 [ACK] Seq=62897 Win=0 Len=0
557	2021-01-21 01:15:31.471543	192.168.1.110	192.168.1.33	TCP	66	62900 → 7000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
558	2021-01-21 01:15:31.471574	192.168.1.33	192.168.1.110	TCP	66	7000 → 62910 [ACK] Seq=62910 Win=0 Len=0
559	2021-01-21 01:15:31.471927	192.168.1.110	192.168.1.33	TCP	54	62900 → 7000 [ACK] Seq=62900 Win=0 Len=0
560	2021-01-21 01:15:31.472746	192.168.1.110	192.168.1.33	TCP	66	62900 → 7000 [ACK] Seq=62900 Win=0 Len=0
561	2021-01-21 01:15:31.472757	192.168.1.33	192.168.1.110	TCP	54	7000 → 62910 [ACK] Seq=62910 Win=0 Len=0
562	2021-01-21 01:15:31.477960	192.168.1.33	192.168.1.110	TCP	67	7000 → 62910 [ACK] Seq=62910 Win=0 Len=0
563	2021-01-21 01:15:31.478032	192.168.1.33	192.168.1.110	TCP	54	7000 → 62910 [ACK] Seq=62910 Win=0 Len=0
564	2021-01-21 01:15:31.478306	192.168.1.110	192.168.1.33	TCP	54	62900 → 7000 [ACK] Seq=62900 Win=0 Len=0
565	2021-01-21 01:15:31.479254	192.168.1.110	192.168.1.33	TCP	54	62900 → 7000 [ACK] Seq=62900 Win=0 Len=0
566	2021-01-21 01:15:31.479271	192.168.1.33	192.168.1.110	TCP	54	7000 → 62910 [ACK] Seq=62910 Win=0 Len=0
568	2021-01-21 01:15:31.978029	192.168.1.110	192.168.1.33	TCP	66	62910 → 7000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
569	2021-01-21 01:15:31.978051	192.168.1.33	192.168.1.110	TCP	66	7000 → 62910 [ACK] Seq=62910 Win=0 Len=0
570	2021-01-21 01:15:31.971077	192.168.1.110	192.168.1.33	TCP	54	62910 → 7000 [ACK] Seq=62910 Win=0 Len=0
571	2021-01-21 01:15:31.971741	192.168.1.110	192.168.1.33	TCP	65	62910 → 7000 [PSH, ACK] Seq=62910 Win=0 Len=11 [TCP segment of a reassembled PDU]
572	2021-01-21 01:15:31.971747	192.168.1.33	192.168.1.110	TCP	54	7000 → 62910 [ACK] Seq=62910 Win=0 Len=0
573	2021-01-21 01:15:31.974082	192.168.1.33	192.168.1.110	TCP	4150	7000 → 62910 [PSH, ACK] Seq=62910 Win=0 Len=4096 [TCP segment of a reassembled PDU]
574	2021-01-21 01:15:31.974821	192.168.1.33	192.168.1.110	TCP	7354	7000 → 62910 [PSH, ACK] Seq=62910 Win=0 Len=7300 [TCP segment of a reassembled PDU]
575	2021-01-21 01:15:31.974834	192.168.1.33	192.168.1.110	Gryphon	2974	14654 - Invalid - Invalid -
576	2021-01-21 01:15:31.975232	192.168.1.110	192.168.1.33	TCP	54	62910 → 7000 [ACK] Seq=62910 Win=0 Len=0
577	2021-01-21 01:15:31.975241	192.168.1.33	192.168.1.110	Gryphon	14654	14654 - Invalid - Invalid -
578	2021-01-21 01:15:31.975243	192.168.1.33	192.168.1.110	Gryphon	14654	14654 - Invalid - Invalid -
579	2021-01-21 01:15:31.975232	192.168.1.110	192.168.1.33	TCP	54	[TCP Window Update] 62910 → 7000 [ACK] Seq=12 Ack=14317 Win=2092032 Len=0
580	2021-01-21 01:15:31.975232	192.168.1.33	192.168.1.110	TCP	54	[TCP Window Update] 62910 → 7000 [ACK] Seq=12 Ack=14317 Win=2096128 Len=0

Teniendo en cuenta que la transmisión de datos a través del protocolo TCP y para garantizar una mayor integridad de la comunicación se realiza a través de la Three-Way Handshake y siguiendo el flag **FIN == 1** como final de una conexión terminada limpiamente, obtenemos todas las diferentes comunicaciones:

actividad2.pcap

tcp.dst == 192.168.1.33 and tcp.dport == 7000 and tcp.flags.fin == 1

No.	Time	Source	Destination	Protocol	Length	Info
555	2021-01-21 01:15:31.281832	192.168.1.110	192.168.1.33	TCP	54	62897 → afs3-fileserver(7000) [FIN, ACK] Seq=3 Ack=225 Win=2102016 Len=0
556	2021-01-21 01:15:31.281843	192.168.1.33	192.168.1.110	TCP	54	62910 → afs3-fileserver(7000) [FIN, ACK] Seq=13 Ack=15 Win=2102272 Len=0
650	2021-01-21 01:15:41.702555	192.168.1.110	192.168.1.33	TCP	54	62910 → afs3-fileserver(7000) [FIN, ACK] Seq=12 Ack=227337 Win=2102016 Len=0
1082	2021-01-21 01:16:00.280494	192.168.1.110	192.168.1.33	TCP	54	62949 → afs3-fileserver(7000) [FIN, ACK] Seq=8 Ack=49 Win=2102272 Len=0

Siguiendo la secuencia 6 de la comunicación TCP observamos el tracing completo de la comunicación:

actividad2.pcap

tcp.stream eq 6

No.	Time	Source	Destination	Protocol	Length	Info
547	2021-01-21 01:15:31.277631	192.168.1.110	192.168.1.33	TCP	66	62897 → 7000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
548	2021-01-21 01:15:31.277655	192.168.1.33	192.168.1.110	TCP	66	7000 → 62897 [ACK] Seq=62897 Win=0 Len=0
549	2021-01-21 01:15:31.277851	192.168.1.110	192.168.1.33	TCP	54	62897 → 7000 [ACK] Seq=62897 Win=0 Len=0
550	2021-01-21 01:15:31.278629	192.168.1.110	192.168.1.33	TCP	56	62897 → 7000 [PSH, ACK] Seq=62897 Win=0 Len=2 [TCP segment of a reassembled PDU]
551	2021-01-21 01:15:31.278696	192.168.1.33	192.168.1.110	TCP	54	7000 → 62897 [ACK] Seq=62897 Win=0 Len=0
552	2021-01-21 01:15:31.281381	192.168.1.33	192.168.1.110	TCP	111	7000 → 62897 [PSH, ACK] Seq=62897 Win=0 Len=57 [TCP segment of a reassembled PDU]
553	2021-01-21 01:15:31.281439	192.168.1.33	192.168.1.110	TCP	220	7000 → 62897 [FIN, PSH, ACK] Seq=58 Ack=3 Win=64256 Len=166 [TCP segment of a reassembled PDU]
554	2021-01-21 01:15:31.281651	192.168.1.110	192.168.1.33	TCP	54	62897 → 7000 [ACK] Seq=62897 Win=0 Len=0
555	2021-01-21 01:15:31.281832	192.168.1.110	192.168.1.33	TCP	54	62897 → 7000 [ACK] Seq=62897 Win=0 Len=0
556	2021-01-21 01:15:31.281843	192.168.1.33	192.168.1.110	TCP	54	7000 → 62897 [ACK] Seq=62897 Win=0 Len=0

Wireshark - Seguir secuencia TCP (tcp.stream eq 6) - actividad2.pcap

Welcome to the 192.168.1.33:7000 server! fake (NULL) 0

fake (NULL) 0

Hello /hello.txt 192.168.1.33 7000 +

1RfCs /dir/ 192.168.1.33 7000 +

gLena /lena.gif 192.168.1.33 7000 +

lQuux Mega Server / 142.4.200.132 7000

Conversación completa (225 bytes) Mostrar datos como ASCII Secuencia 6

Buscar: Buscar siguiente

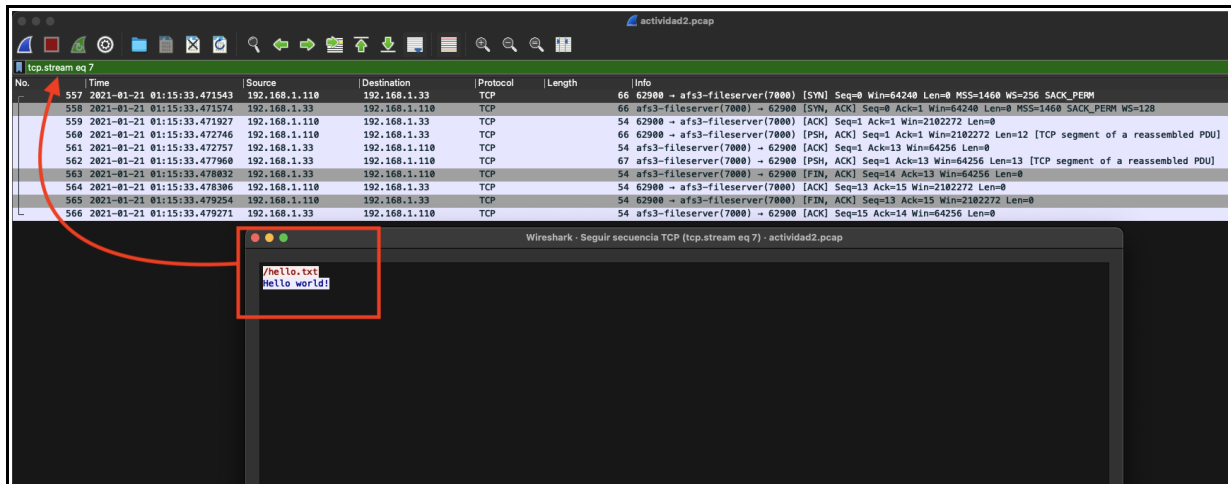
Help Filtrar secuencia Imprimir Guardar como... Atrás Close

Podemos observar como se realizan 3 accesos a directorios

1. /hello.txt
2. /dir/
3. /lena.gif

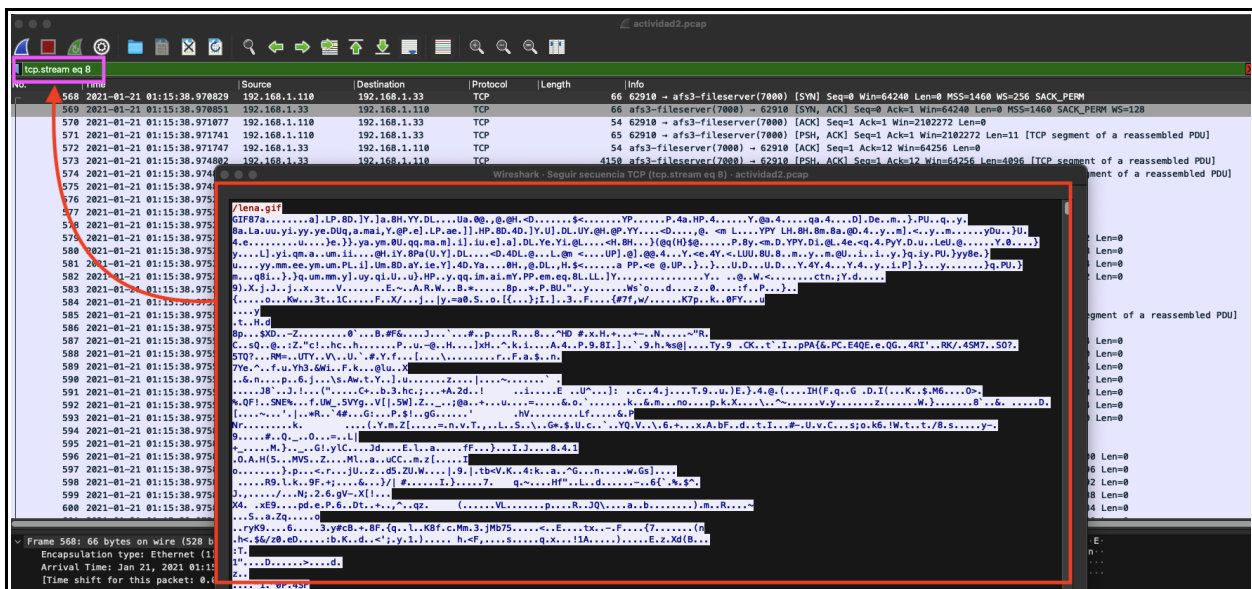
Filtremos la secuencia de comunicación numero 7:

Podemos observar como se accede al archivo hello.txt y el destino nos devuelve Hello world!

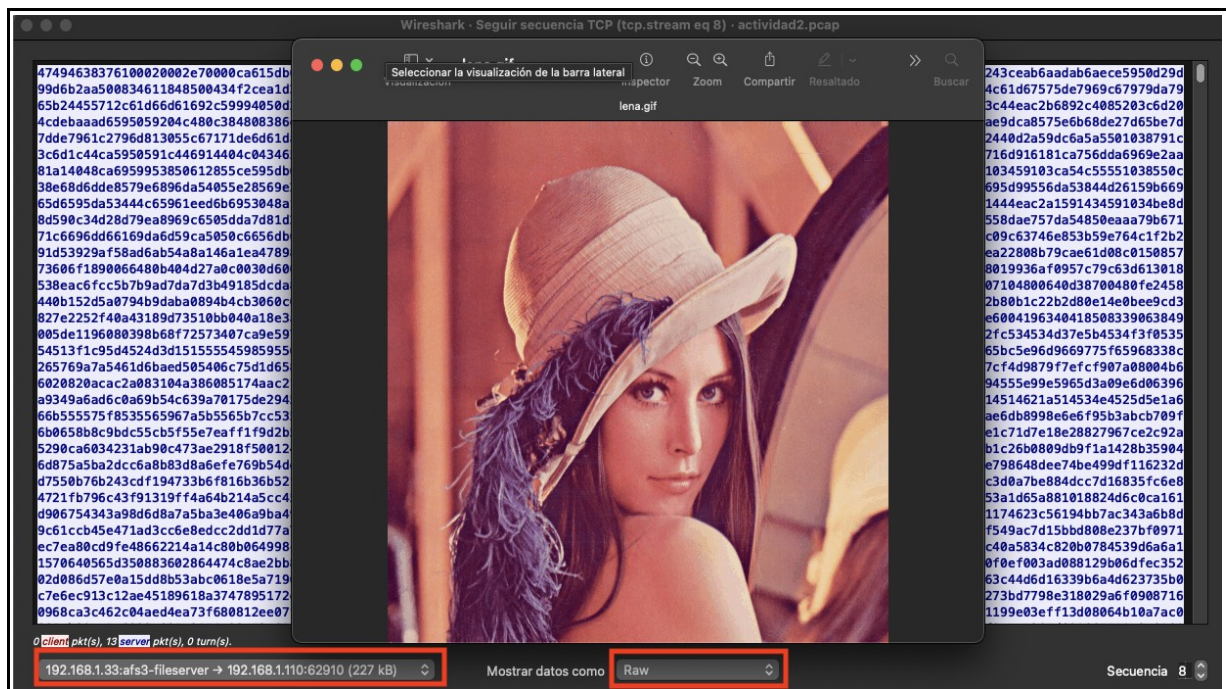


Filtramos la secuencia de comunicación numero 8 y el seguimiento de la secuencia nos vuelca una transacción de 227kB:

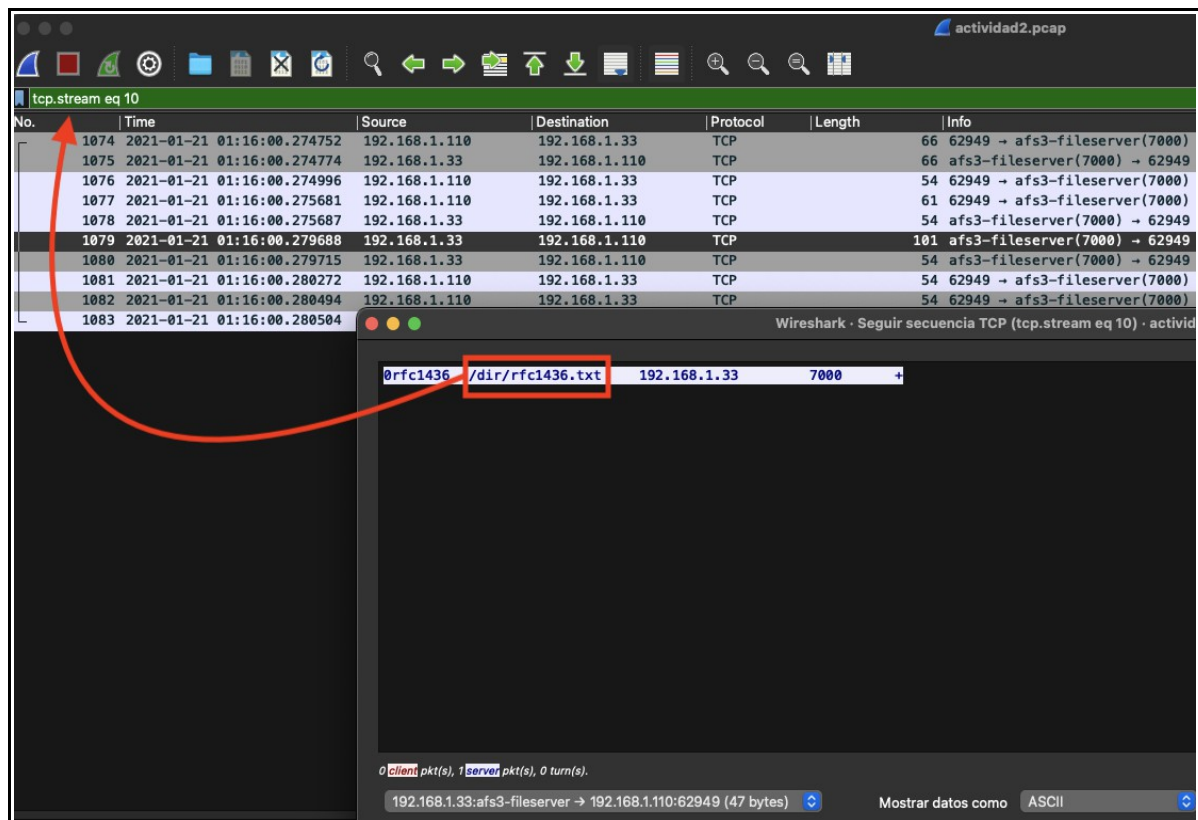
- 192.168.1.33:afs3-fileserver → 192.168.1.110:62910



Guardamos la conversación completa extrayendo los primeros 11 bytes correspondientes `/lena.gif` y almacenamos un raw de la comunicación como `lena.gif`.



Filtrando la ultima secuencia podemos observar como se intenta obtener el archivo `rfc1436.txt` pero la conversación termina antes de obtener el archivo.



RFC1436 nos esta haciendo referencia al protocolo Gopher:

Gopher es un servicio de Internet consistente en el acceso a la información a través de menús. La información se organiza en forma de árbol: solo los *nodos* contienen menús de acceso a otros menús o a *hojas*, mientras que las hojas contienen simplemente información textual. En cierto modo es considerado un predecesor de la Web, aunque solo se permiten enlaces desde nodos-menús hasta otros nodos-menús o a hojas, y las hojas no tienen ningún tipo de hiperenlaces.