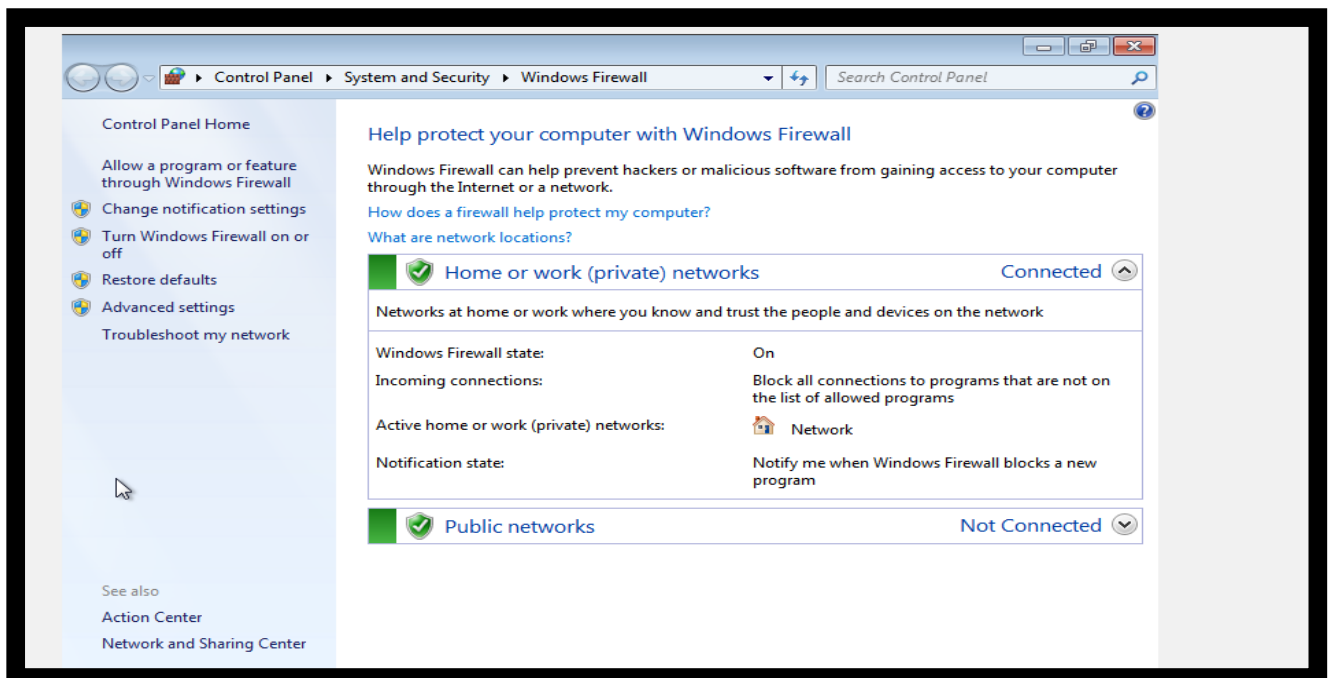


# REPORT CONFIGURAZIONE POLICY\PACKET SPOOFING

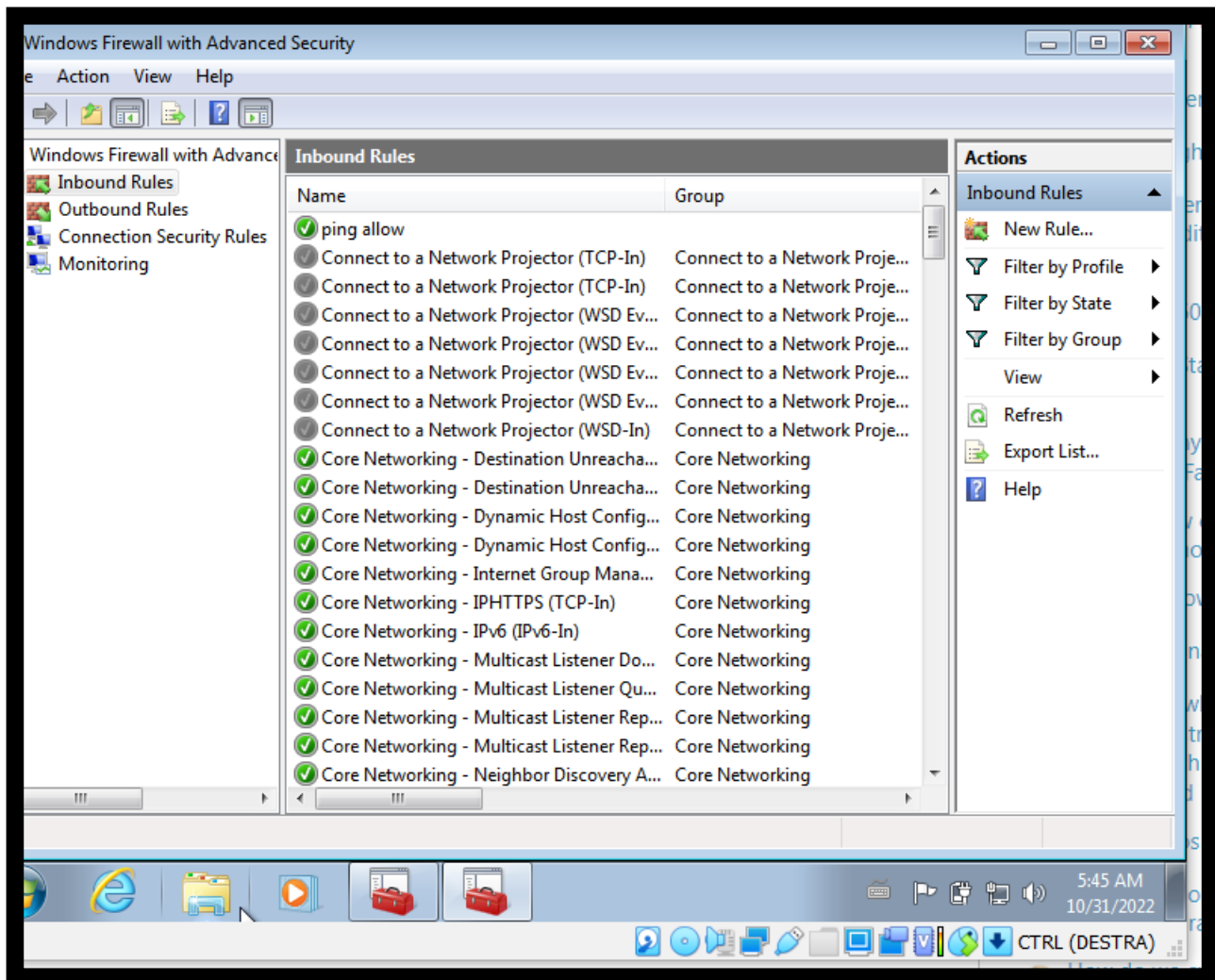
LE TASK RICHIEDONO:

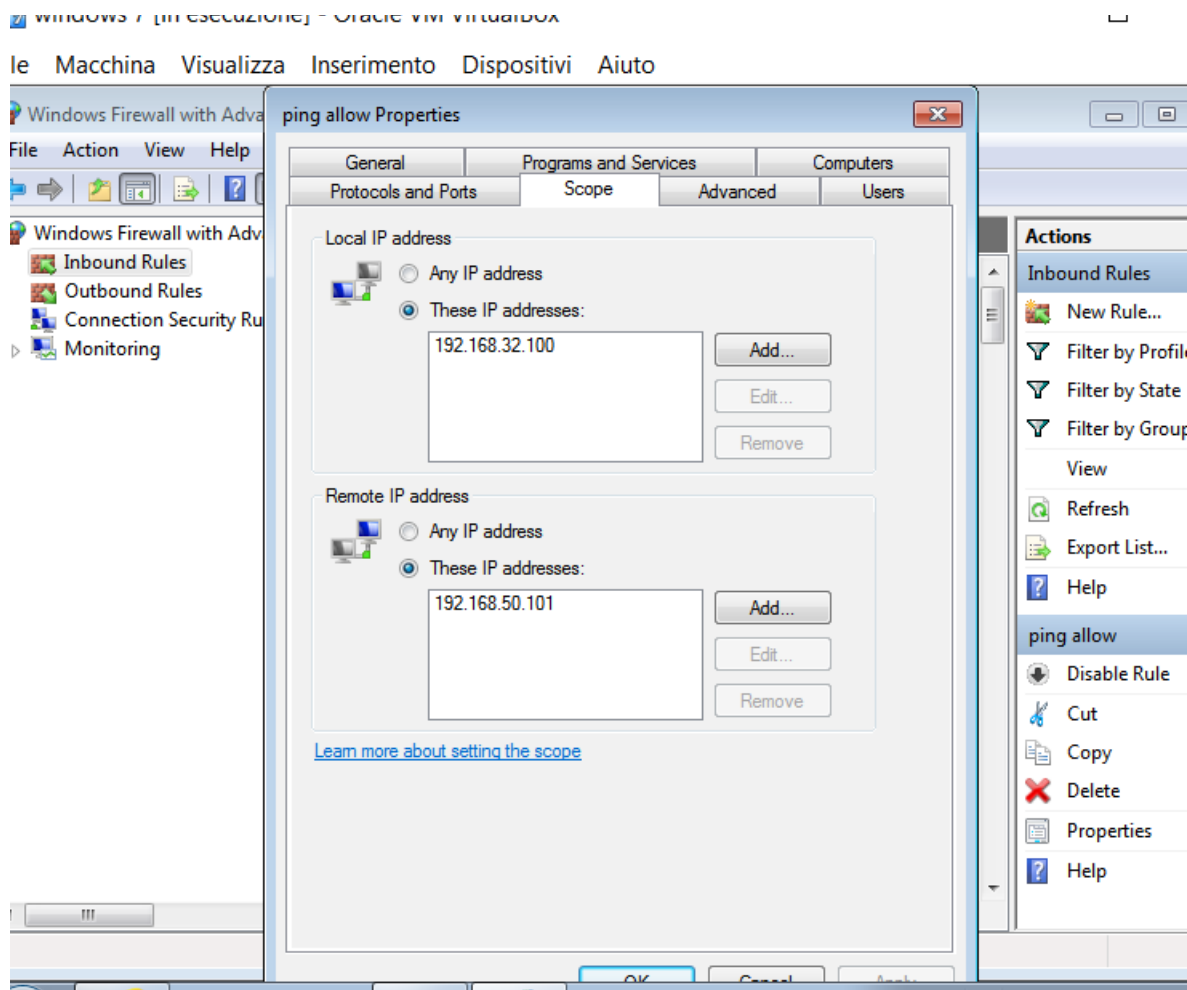
1. La configurazione di firewall policy per il pinginng delle macchine Linux e windows7 in virtual lab.
2. Utilizzo inetsim per la simulazione di servizi internet standard
3. Packet spoofing tramite tool wireshark

Primo step fondamentale è l'attivazione dei firewall windows, step senza la quale non avremmo alcun ping tra le macchine:

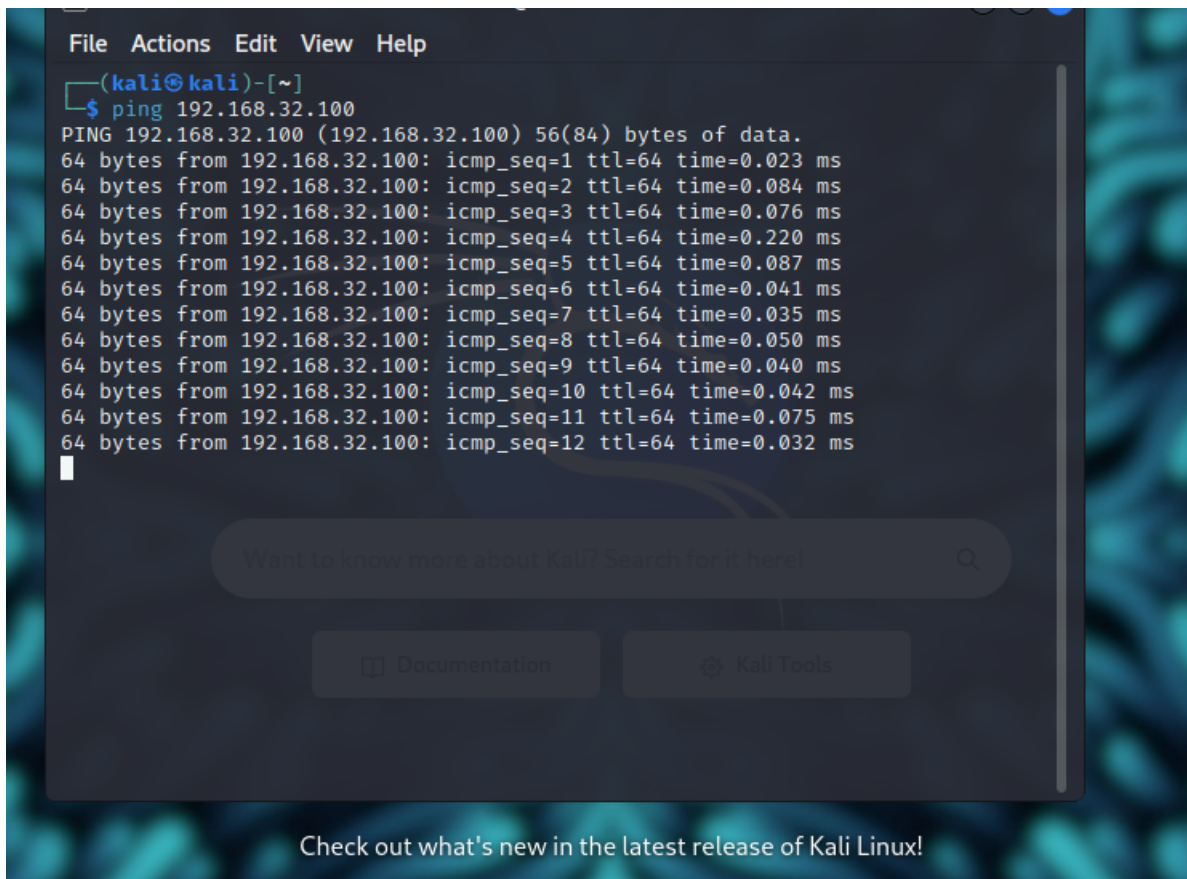


Impostiamo quindi una firewall policy dedicata, aggiungendo alla sezione "These IP address" gli indirizzi delle macchine di cui siamo interessati alla connessione.

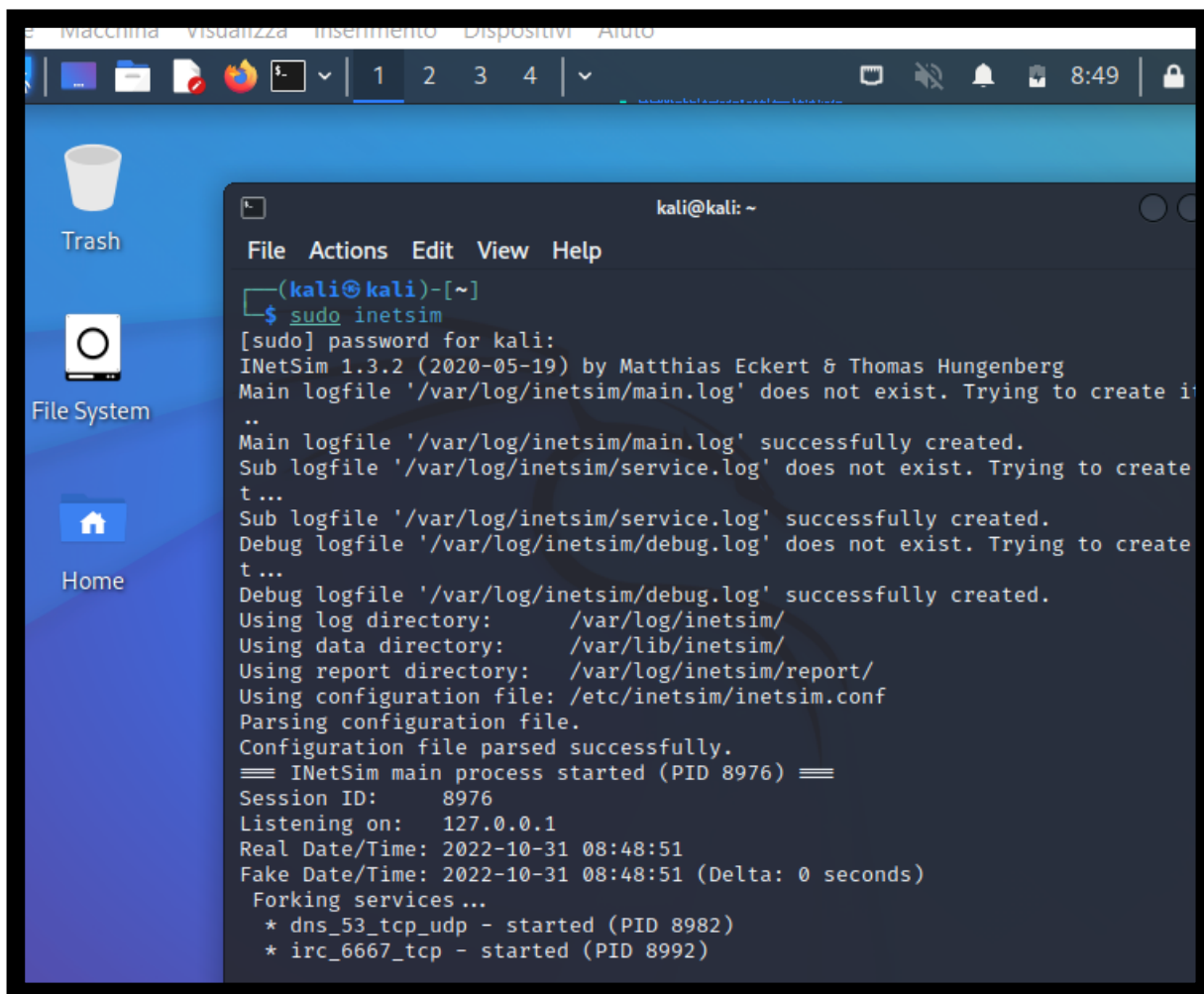




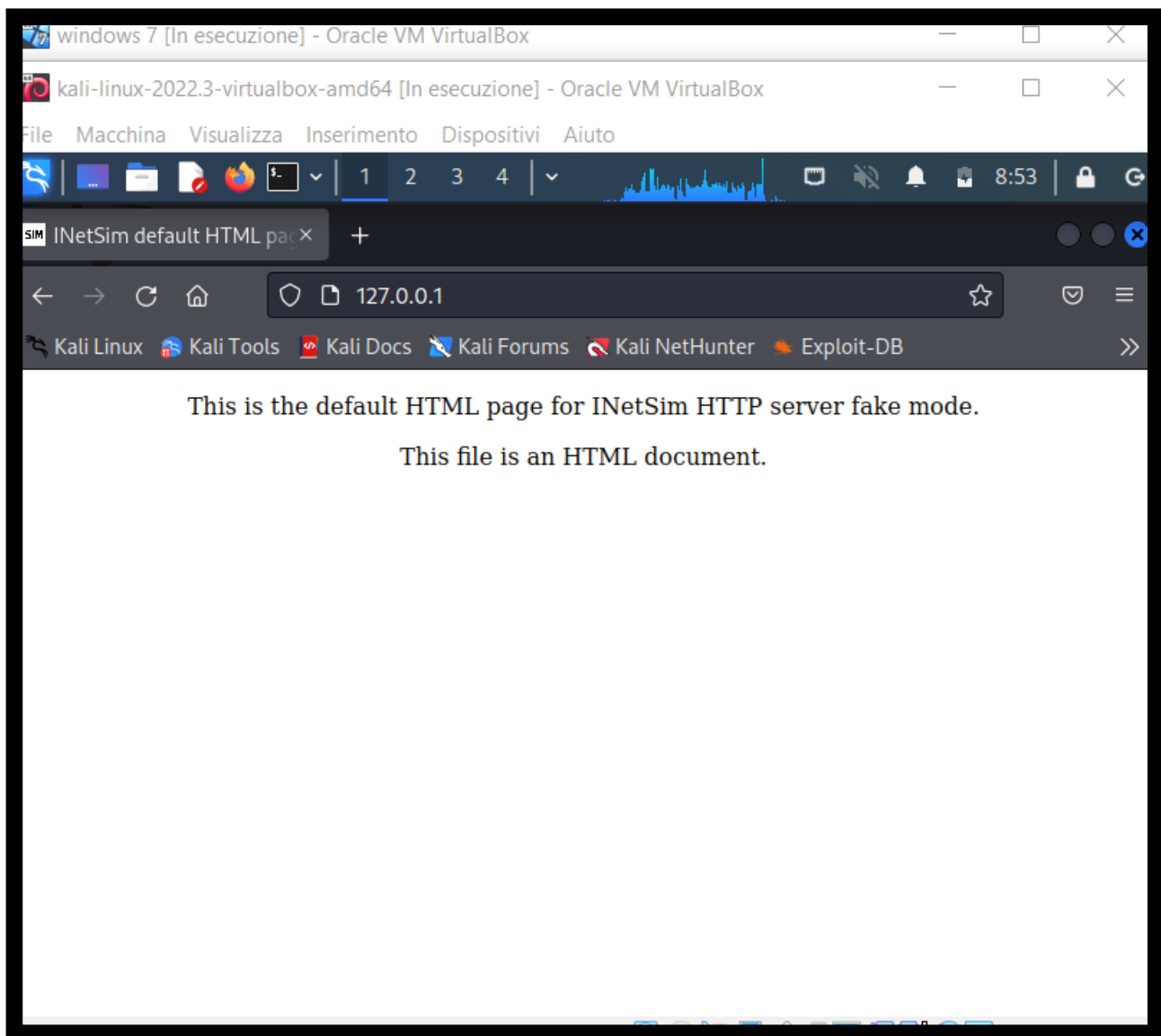
A questo punto, potremmo testare la comunicazione tra le due macchine



Una volta ottenuta la conferma sarà possibile attivare un server simulato con il comando `<<sudo inetsim>>` sul prompt comandi Kali.



Alla voce <<Listening on>> si troverà l'ip da inserire nella ricerca di Firefox in Kali



Adesso, aprendo wireshark ed effettuando una cattura del traffico intercorso otterremo questo risultato.

*Loopback: lo						
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help						
Apply a display filter ... <Ctrl-F>						
No.	Time	Source	Destination	Protocol	Length	Info
9	0.000041016	192.168.50.100	192.168.50.100	ICMP	113	Destination unreachable (Host unreachable)
10	0.592314025	127.0.0.1	127.0.0.1	TCP	74	38628 → 80 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM=1 TSval=1968578666 TSecr=0 WS=128
11	0.592334079	127.0.0.1	127.0.0.1	TCP	74	80 → 38628 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM=1 TSval=1968578666 TSecr=0
12	0.592349451	127.0.0.1	127.0.0.1	TCP	66	38628 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=1968578666 TSecr=1968578666
13	0.592628025	127.0.0.1	127.0.0.1	HTTP	506	GET / HTTP/1.1
14	0.594148460	127.0.0.1	127.0.0.1	TCP	66	80 → 38628 [ACK] Seq=1 Ack=441 Win=65152 Len=0 TSval=1968578666 TSecr=1968578666
15	0.659109876	127.0.0.1	127.0.0.1	TCP	216	80 → 38628 [PSH, ACK] Seq=1 Ack=441 Win=65536 Len=150 TSval=1968578733 TSecr=1968578666 [TCP s
16	0.659233509	127.0.0.1	127.0.0.1	TCP	66	38628 → 80 [ACK] Seq=441 Ack=151 Win=65408 Len=0 TSval=1968578733 TSecr=1968578733
17	0.659259856	127.0.0.1	127.0.0.1	HTTP	324	HTTP/1.1 200 OK (text/html)
18	0.659287801	127.0.0.1	127.0.0.1	TCP	66	38628 → 80 [ACK] Seq=441 Ack=409 Win=65152 Len=0 TSval=1968578733 TSecr=1968578733
19	0.659571759	127.0.0.1	127.0.0.1	TCP	66	38628 → 80 [FIN, ACK] Seq=441 Ack=409 Win=65536 Len=0 TSval=1968578733 TSecr=1968578733
20	0.674162903	127.0.0.1	127.0.0.1	TCP	66	80 → 38628 [FIN, ACK] Seq=409 Ack=442 Win=65536 Len=0 TSval=1968578748 TSecr=1968578733
21	0.674199868	127.0.0.1	127.0.0.1	TCP	66	38628 → 80 [ACK] Seq=442 Ack=419 Win=65536 Len=0 TSval=1968578748 TSecr=1968578748
22	3.067090319	192.168.50.100	192.168.50.100	ICMP	125	Destination unreachable (Host unreachable)
23	3.067097973	192.168.50.100	192.168.50.100	ICMP	123	Destination unreachable (Host unreachable)
24	3.067105885	192.168.50.100	192.168.50.100	ICMP	123	Destination unreachable (Host unreachable)
25	3.067110228	192.168.50.100	192.168.50.100	ICMP	125	Destination unreachable (Host unreachable)
26	3.067113930	192.168.50.100	192.168.50.100	ICMP	115	Destination unreachable (Host unreachable)
27	3.067118495	192.168.50.100	192.168.50.100	ICMP	115	Destination unreachable (Host unreachable)
▶ Frame 12: 506 bytes on wire (4048 bits), 506 bytes captured (4048 bits) on interface lo, id 0 ▶ Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00) ▶ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1 ▶ Transmission Control Protocol, Src Port: 38628, Dst Port: 80, Seq: 1, Ack: 1, Len: 440 ▶ Hypertext Transfer Protocol						
0000	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	00 00 00 00 00 00 00 00	45 00	.....E	
0010	01 ec dc 9b 40 00 00 06	5e 6e 7f 00 00 01 7f 00	.....@.@.An.....			
0020	00 01 96 e4 00 50 c5 a7	4a 01 6f bb 5c 1e 80 18	.....P..J.o.\....			
0030	02 00 ff e0 00 00 01 01	08 0a 75 56 20 6a 75 56	.......uV juV			
0040	20 6a 47 45 54 20 2f 20	48 54 54 50 2f 31 2e 31	jGET / HTTP/1.1			
0050	0d 0a 48 6f 73 74 3a 20	31 32 37 2e 30 2e 30 2e	.Host: 127.0.0.			
0060	31 0d 0a 55 73 65 72 2d	41 67 65 6e 74 3a 20 4d	1 .User-Agent: M			
0070	6f 7a 69 6e 6c 61 2f 35	2e 30 20 20 5b 31 31 3b	ozilla/5.0 (X11;			
0080	20 4c 69 6e 75 78 20 78	38 36 5f 38 3d 3b 20 72	Linux x 86_64; r			
0090	76 3a 39 31 2e 30 29 20	47 65 63 6b 6f 2f 32 30	v:91.0) Gecko/20			
00a0	31 30 30 31 30 31 20 46	69 72 65 66 6f 78 2f 39	100101 Firefox/9			
00b0	31 2e 30 0d 0a 41 63 63	65 70 74 3a 20 74 65 78	1.0 .Accept: tex			
00c0	74 2f 68 74 6d 6c 2c 61	70 70 6c 69 63 61 74 69	t/html,application			
00d0	6f 6e 2f 78 68 74 6d 6c	2b 78 6d 6c 2c 61 70 78	on/xhtml+xml,app			

Dove sarà possibile osservare come tramite filtro protocollo TCP, i messaggi ICMP vengono trasmessi tramite connessione in three-way handshake (SYN|SYN-ACK|ACK) utilizzando un canale http, quindi con flusso di dati in chiaro.