

esercizio w3d4

Giacomo di Giacinto

L'esercizio di oggi mira a consolidare le conoscenze acquisite.

Vedremo due esercizi: I) la configurazione di una policy sul firewall windows; II) una packet capture con Wireshark.

Vedremo anche come simulare alcuni servizi di rete con un tool pre-installato su Kali Linux (InetSim)

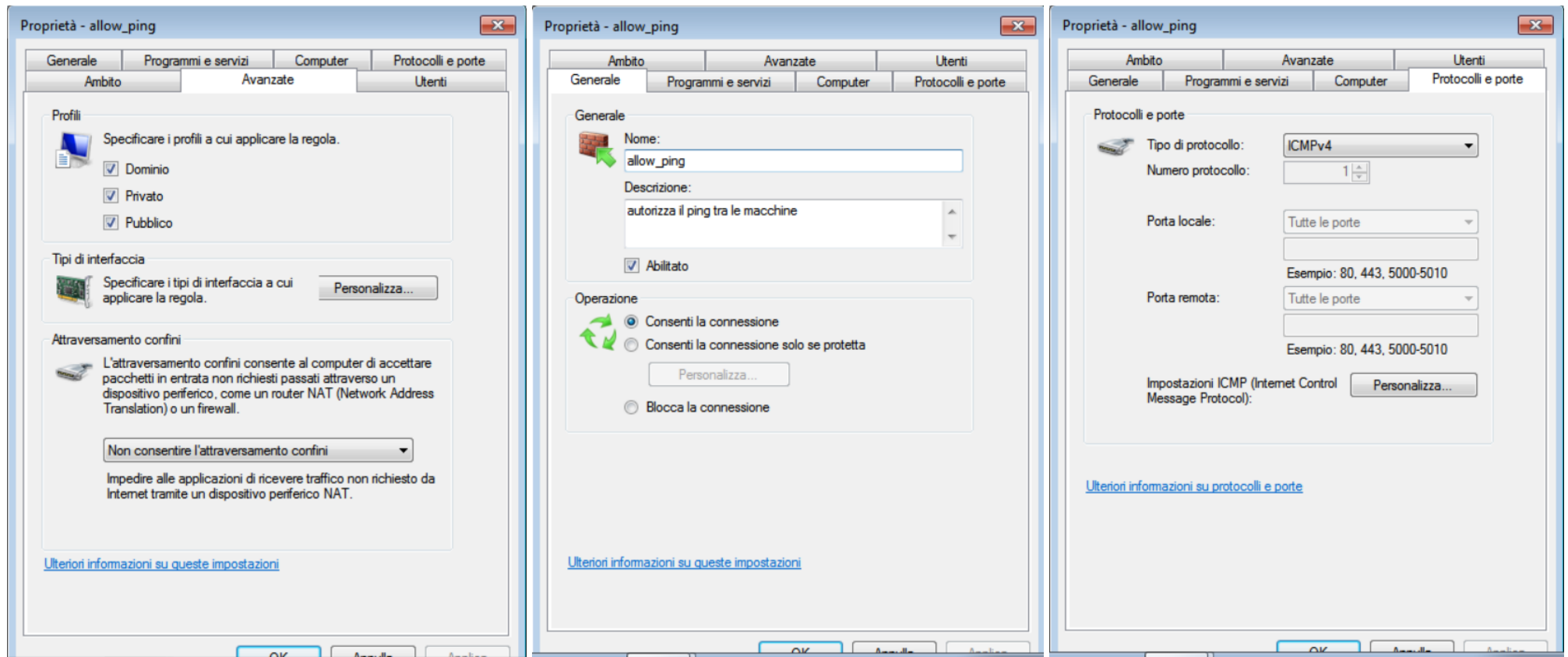
Esercizio:

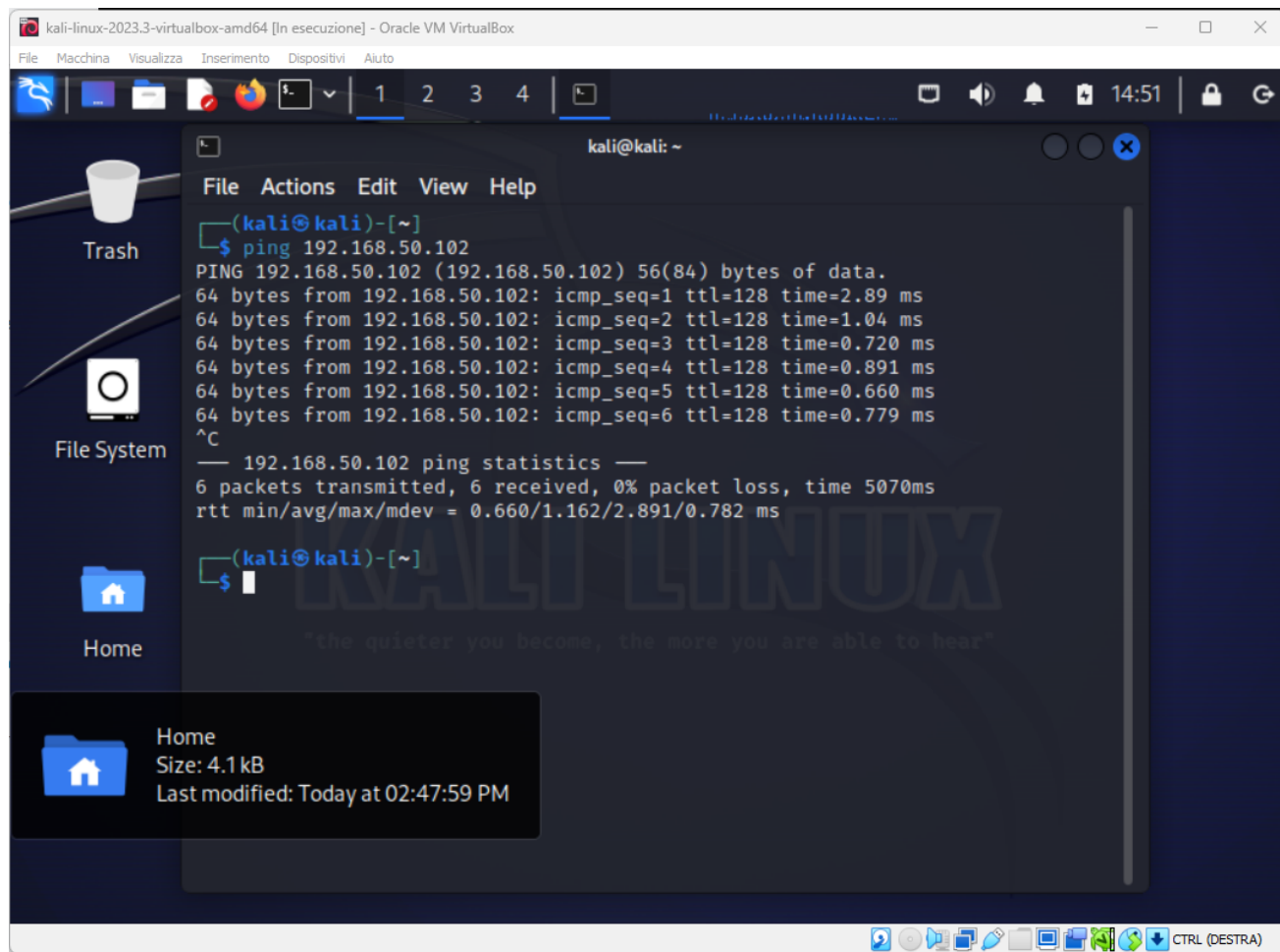
- ❑ Configurare policy per permettere il ping da macchine Linux a Macchina Windows 7 nel nostro laboratorio (Windows firewall)
- ❑ Utilizzo dell'utility InetSim per l'emulazione di servizi Internet
- ❑ Cattura di pacchetti con Wireshark

step 1

Configurazione delle policy sul firewall windows

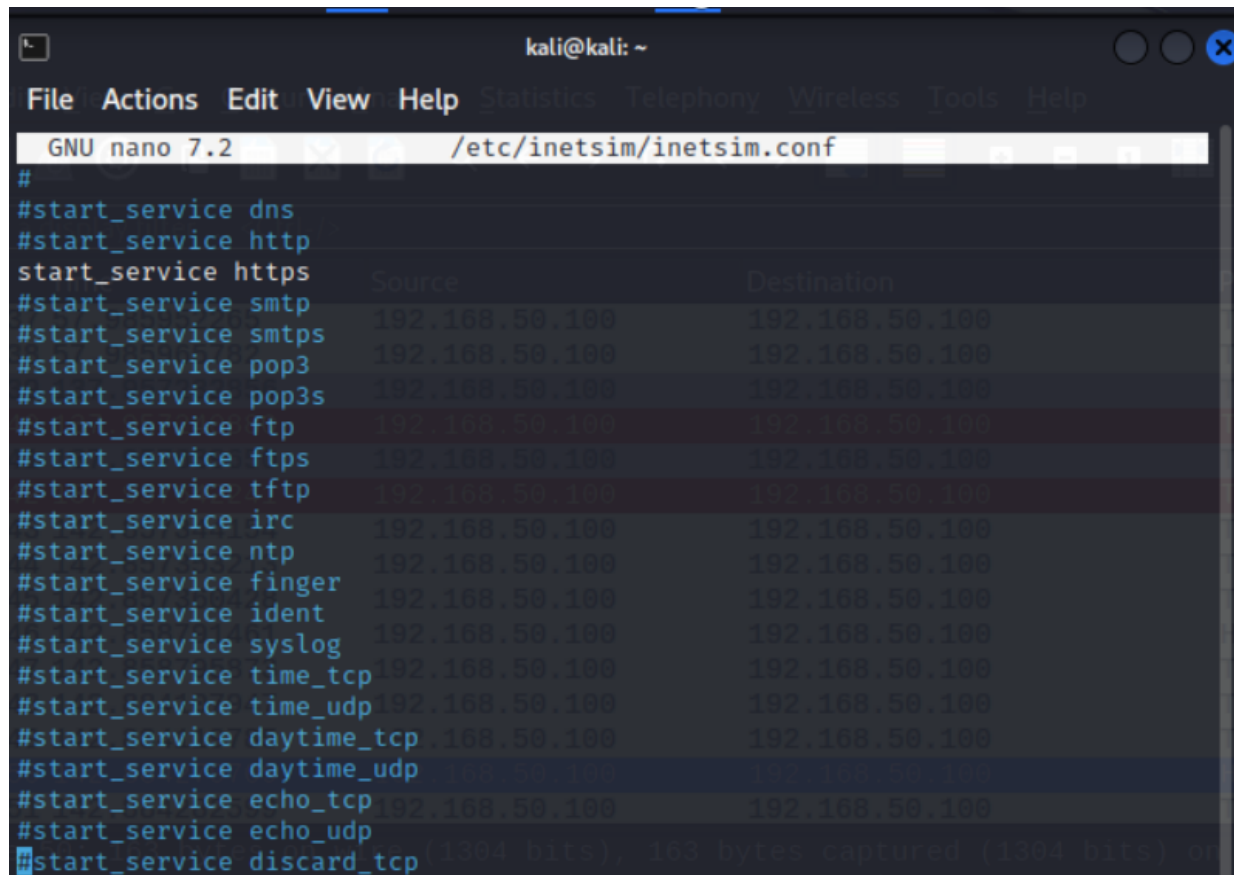
In questa fase creo la regola sul firewall come da esercizio e verifico che il ping tra la macchina kali linux e windows 7 va a buon fine.





step 2 configurazione inetsim

in questa fase configuro come da esercizio l'utility inetsim e, per i protocolli http e https, testo che il la pagina html ed html.txt funzioni correttamente



```
kali@kali: ~  
File Actions Edit View Help Statistics Telephony Wireless Tools Help  
GNU nano 7.2 /etc/inetsim/inetsim.conf  
#  
#start_service dns  
#start_service http  
start_service https  
#start_service smtp  
#start_service smtps  
#start_service pop3  
#start_service pop3s  
#start_service ftp  
#start_service ftps  
#start_service tftp  
#start_service irc  
#start_service ntp  
#start_service finger  
#start_service ident  
#start_service syslog  
#start_service time_tcp  
#start_service time_udp  
#start_service daytime_tcp  
#start_service daytime_udp  
#start_service echo_tcp  
#start_service echo_udp  
#start_service discard_tcp
```

```
kali@kali: ~
File Actions Edit View Help Statistics Telephony Wireless Tools Help
GNU nano 7.2 /etc/inetsim/inetsim.conf
#start_service discard_udp
#start_service quotd_tcp
#start_service quotd_udp
#start_service chargen_tcp
#start_service chargen_udp
#start_service dummy_tcp
#start_service dummy_udp
#####
# service_bind_address
#
# IP address to bind services to
#
# Syntax: service_bind_address <IP address>
#
# Default: 127.0.0.1
#
service_bind_address 192.168.50.100
#####
# service_run_as_user
#####
```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify

assembled TCP Segments (247 bytes): #48(150); #50(97)]



kali@kali: ~



File Actions Edit View Help

64 bytes from 192.168.50.102: icmp_seq=16 ttl=128 time=0.591 ms

^X64 bytes from 192.168.50.102: icmp_seq=17 ttl=128 time=0.373 ms

^C

— 192.168.50.102 ping statistics —

17 packets transmitted, 17 received, 0% packet loss, time 16043ms

rtt min/avg/max/mdev = 0.373/1.190/2.201/0.443 ms

(kali@kali)-[~]

\$ sudo inetsim

[sudo] password for kali:

INetSim 1.3.2 (2020-05-19) by Matthias Eckert & Thomas Hungenberg

Using log directory: /var/log/inetsim/

Using data directory: /var/lib/inetsim/

Using report directory: /var/log/inetsim/report/

Using configuration file: /etc/inetsim/inetsim.conf

Parsing configuration file.

Configuration file parsed successfully.

≡ INetSim main process started (PID 1317396) ≡

Session ID: 1317396

Listening on: 192.168.50.100

Real Date/Time: 2023-12-17 09:47:52

Fake Date/Time: 2023-12-17 09:47:52 (Delta: 0 seconds)

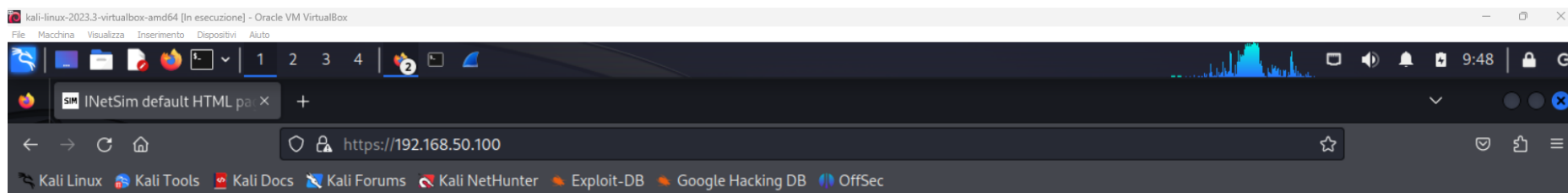
Forking services ...

* https_443_tcp - started (PID 1317398)

done.

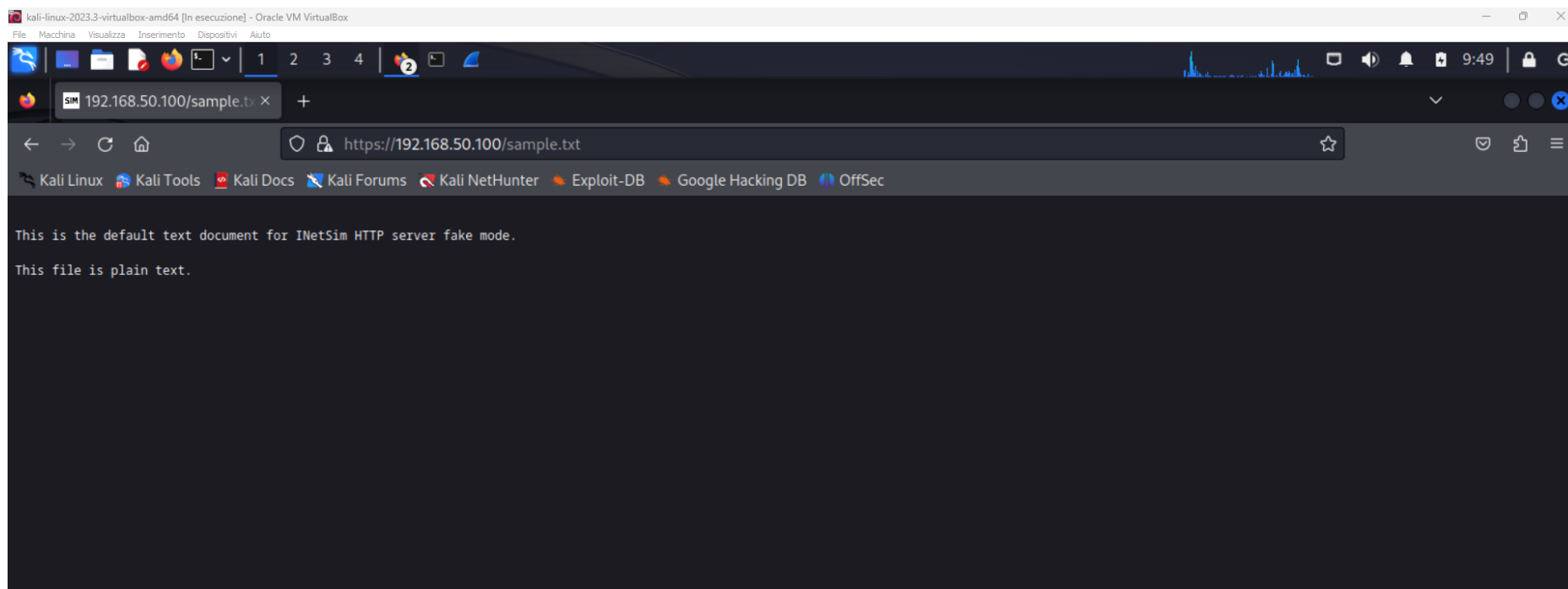
Simulation running.





This is the default HTML page for INetSim HTTP server fake mode.

This file is an HTML document.



lo stesso processo lo ripeto con il protocollo http.

step 3 cattura pacchetti con wireshark

in questa fase inizio la cattura dei pacchetti e verifico le differenze tra il protocollo http e https

The screenshot shows the Wireshark network protocol analyzer interface. The top toolbar contains icons for file operations, editing, and navigation. Below it is a filter bar with the text "Apply a display filter ... <Ctrl-/>" and a refresh button.

The main pane displays a list of captured packets:

No.	Time	Source	Destination	Protocol	Length Info
19	12.287103832	192.168.50.100	192.168.50.100	ICMP	138 Destination unreachable (Host unreachable)
20	12.287105501	192.168.50.100	192.168.50.100	ICMP	138 Destination unreachable (Host unreachable)
21	13.720900669	192.168.50.100	192.168.50.100	TCP	74 52050 → 443 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM TSval=386722...
22	13.720911290	192.168.50.100	192.168.50.100	TCP	74 443 → 52050 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM T...
23	13.720920193	192.168.50.100	192.168.50.100	TCP	66 52050 → 443 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=3867227084 TSecr=386...
24	13.722264672	192.168.50.100	192.168.50.100	TLSv1.3	687 Client Hello
25	13.722268700	192.168.50.100	192.168.50.100	TCP	66 443 → 52050 [ACK] Seq=1 Ack=622 Win=64896 Len=0 TSval=3867227086 TSecr=3...
26	13.754888704	192.168.50.100	192.168.50.100	TLSv1.3	1487 Server Hello, Change Cipher Spec, Application Data, Application Data, Ap...
27	13.754904611	192.168.50.100	192.168.50.100	TCP	66 52050 → 443 [ACK] Seq=622 Ack=1422 Win=64384 Len=0 TSval=3867227118 TSec...
28	13.761886920	192.168.50.100	192.168.50.100	TLSv1.3	146 Change Cipher Spec, Application Data
29	13.762048581	192.168.50.100	192.168.50.100	TLSv1.3	534 Application Data
30	13.762387783	192.168.50.100	192.168.50.100	TLSv1.3	321 Application Data
31	13.781147681	192.168.50.100	192.168.50.100	TLSv1.3	637 Application Data, Application Data, Application Data, Application Data
32	13.781567608	192.168.50.100	192.168.50.100	TCP	66 52050 → 443 [ACK] Seq=1170 Ack=2249 Win=65536 Len=0 TSval=3867227145 TSe...
33	13.781661454	192.168.50.100	192.168.50.100	TLSv1.3	90 Application Data

Below the packet list, the details pane shows information about the selected frame (Frame 1). It indicates that 138 bytes were on wire (1104 bits) and 138 bytes were captured (1104 bits) on the interface. The Ethernet II section shows Source MAC address as 00:00:00_00:00:00 and Destination MAC address as 00:00:00_00:00:00. The Internet Protocol Version 4 section shows Source IP as 192.168.50.100 and Destination IP as 192.168.50.100. The Internet Control Message Protocol section is expanded, showing the type as Echo (ping request). The Domain Name System section shows a query for mozilla.com.hometel.ecomital.it.

The packet bytes pane at the bottom displays the raw data of the first packet in hexadecimal and ASCII format. The ASCII column highlights the domain name "mozilla.com.hometel.ecomital.it".

Capturing from Loopback: lo

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
6	0.000010748	192.168.50.100	192.168.50.100	ICMP	113	Destination unreachable (Host unreachable)
7	1.486783415	192.168.50.100	192.168.50.100	TCP	74	56224 → 80 [SYN] Seq=0 Win=65495 Len=0 MSS=65495 SACK_PERM TSval=3867374598 ...
8	1.486800854	192.168.50.100	192.168.50.100	TCP	74	80 → 56224 [SYN, ACK] Seq=0 Ack=1 Win=65483 Len=0 MSS=65495 SACK_PERM TSval=...
9	1.486808697	192.168.50.100	192.168.50.100	TCP	66	56224 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0 TSval=3867374598 TSecr=38673745...
10	1.486900629	192.168.50.100	192.168.50.100	HTTP	414	GET /sample.txt HTTP/1.1
11	1.486903076	192.168.50.100	192.168.50.100	TCP	66	80 → 56224 [ACK] Seq=1 Ack=349 Win=65152 Len=0 TSval=3867374598 TSecr=386737...
12	1.509827417	192.168.50.100	192.168.50.100	TCP	216	80 → 56224 [PSH, ACK] Seq=1 Ack=349 Win=65536 Len=150 TSval=3867374621 TSecr=...
13	1.509840770	192.168.50.100	192.168.50.100	TCP	66	56224 → 80 [ACK] Seq=349 Ack=151 Win=65408 Len=0 TSval=3867374621 TSecr=3867...
14	1.509946594	192.168.50.100	192.168.50.100	HTTP	163	HTTP/1.1 200 OK (text/plain)
15	1.509948665	192.168.50.100	192.168.50.100	TCP	66	56224 → 80 [ACK] Seq=349 Ack=248 Win=65408 Len=0 TSval=3867374621 TSecr=3867...
16	1.510041832	192.168.50.100	192.168.50.100	TCP	66	56224 → 80 [FIN, ACK] Seq=349 Ack=248 Win=65536 Len=0 TSval=3867374621 TSecr=...
17	1.513717824	192.168.50.100	192.168.50.100	TCP	66	80 → 56224 [FIN, ACK] Seq=248 Ack=350 Win=65536 Len=0 TSval=3867374625 TSecr=...
18	1.513735004	192.168.50.100	192.168.50.100	TCP	66	56224 → 80 [ACK] Seq=350 Ack=249 Win=65536 Len=0 TSval=3867374625 TSecr=3867...

Frame 10: 414 bytes on wire (3312 bits), 414 bytes captured (3312 bits) on interface

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: 192.168.50.100, Dst: 192.168.50.100

Transmission Control Protocol, Src Port: 56224, Dst Port: 80, Seq: 1, Ack: 1, Len: 414

Hypertext Transfer Protocol

GET /sample.txt HTTP/1.1\r\n

Host: 192.168.50.100\r\n

User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0\r\n

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8\r\n

Accept-Language: en-US,en;q=0.5\r\n

Accept-Encoding: gzip, deflate\r\n

Connection: keep-alive\r\n

Upgrade-Insecure-Requests: 1\r\n

\r\n

[Full request URI: http://192.168.50.100/sample.txt]

[HTTP request 1/1]

[Response in frame: 14]

Frame (frame), 414 bytes

Packets: 70 · Displayed: 70 (100.0%)

Profile: Default

la cattura http restituisce come risultato tutte le informazioni dettagliate collegate al contenuto della pagina mentre in https questi dettagli non sono riportati.

Infine catturo con wireshark anche la comunicazione attraverso Ping tra la macchina kali linux e windows 7 e questo è il risultato

The image shows a Wireshark network traffic capture window. The top bar indicates the capture is from the 'eth0' interface. The main display area shows a list of captured packets, with packet 25 selected. The packet list table is as follows:

No.	Time	Source	Destination	Protocol	Length	Info
22	8.020790934	192.168.50.102	192.168.50.100	ICMP	98	Echo (ping) reply id=0x7997, seq=9/2304, ttl=128 (request in 21)
23	9.022298435	192.168.50.100	192.168.50.102	ICMP	98	Echo (ping) request id=0x7997, seq=10/2560, ttl=64 (reply in 24)
24	9.023254677	192.168.50.102	192.168.50.100	ICMP	98	Echo (ping) reply id=0x7997, seq=10/2560, ttl=128 (request in 23)
25	10.025603338	192.168.50.100	192.168.50.102	ICMP	98	Echo (ping) request id=0x7997, seq=11/2816, ttl=64 (reply in 26)
26	10.026113975	192.168.50.102	192.168.50.100	ICMP	98	Echo (ping) reply id=0x7997, seq=11/2816, ttl=128 (request in 25)
27	11.032079034	192.168.50.100	192.168.50.102	ICMP	98	Echo (ping) request id=0x7997, seq=12/3072, ttl=64 (reply in 28)
28	11.033335230	192.168.50.102	192.168.50.100	ICMP	98	Echo (ping) reply id=0x7997, seq=12/3072, ttl=128 (request in 27)
29	12.033377665	192.168.50.100	192.168.50.102	ICMP	98	Echo (ping) request id=0x7997, seq=13/3328, ttl=64 (reply in 30)
30	12.034155706	192.168.50.102	192.168.50.100	ICMP	98	Echo (ping) reply id=0x7997, seq=13/3328, ttl=128 (request in 29)
31	13.035850100	192.168.50.100	192.168.50.102	ICMP	98	Echo (ping) request id=0x7997, seq=14/3584, ttl=64 (reply in 32)
32	13.036848686	192.168.50.102	192.168.50.100	ICMP	98	Echo (ping) reply id=0x7997, seq=14/3584, ttl=128 (request in 31)
33	14.036991329	192.168.50.100	192.168.50.102	ICMP	98	Echo (ping) request id=0x7997, seq=15/3840, ttl=64 (reply in 34)
34	14.037870051	192.168.50.102	192.168.50.100	ICMP	98	Echo (ping) reply id=0x7997, seq=15/3840, ttl=128 (request in 33)

The packet details pane for packet 25 shows the following structure:

- Frame 25: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface
- Ethernet II, Src: PcsCompu_cb:7e:f5 (08:00:27:cb:7e:f5), Dst: PcsCompu_0e:c4:ea (08:00:27:0e:c4:ea)
 - Destination: PcsCompu_0e:c4:ea (08:00:27:0e:c4:ea)
 - Source: PcsCompu_cb:7e:f5 (08:00:27:cb:7e:f5)
 - Type: IPv4 (0x0800)
- Internet Protocol Version 4, Src: 192.168.50.100, Dst: 192.168.50.102
 - 0100 = Version: 4
 - ... 0101 = Header Length: 20 bytes (5)
 - Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
 - Total Length: 84
 - Identification: 0xae32 (44594)
 - 010. = Flags: 0x2, Don't fragment
 - ...0 0000 0000 0000 = Fragment Offset: 0
 - Time to Live: 64
 - Protocol: ICMP (1)
 - Header Checksum: 0xa65b [validation disabled]
 - [Header checksum status: Unverified]

The packet bytes pane shows the raw data of the selected packet, with a hex dump and ASCII representation.

eth0: <live capture in progress> Packets: 34 · Displayed: 34 (100.0%) Profile: Default