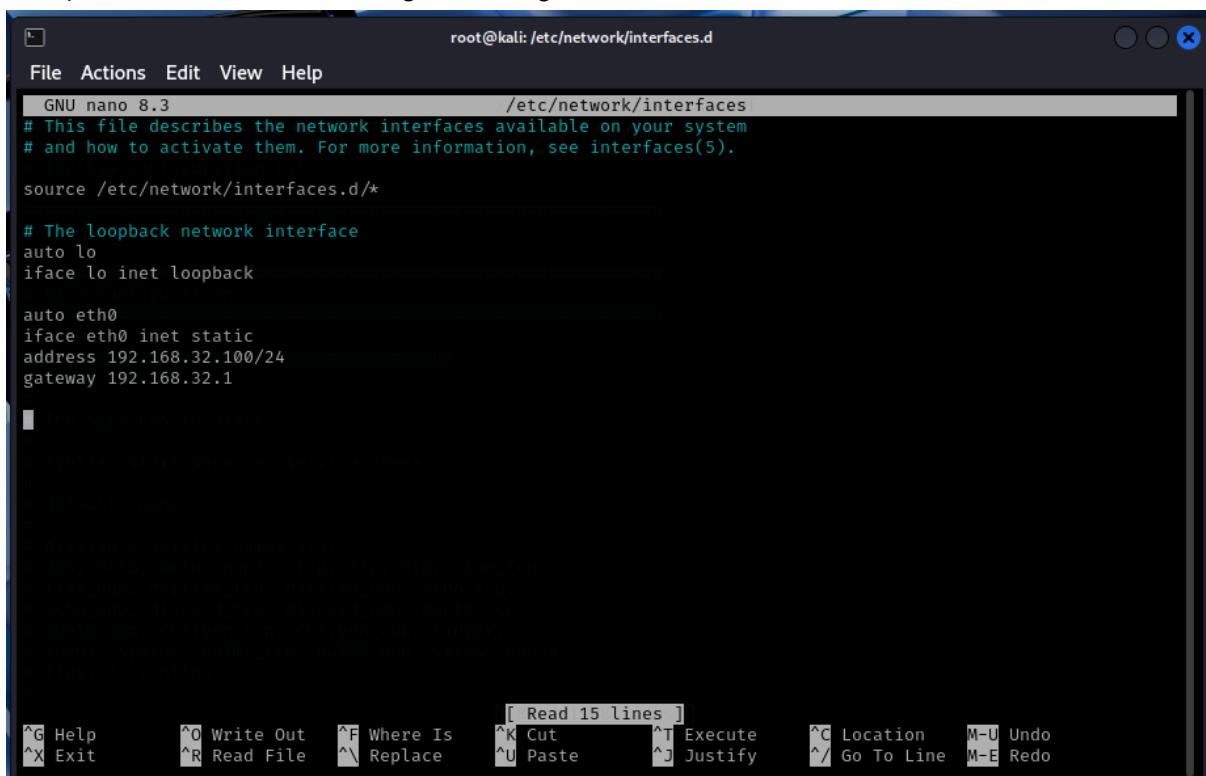


Traccia :

- 1) Simulare, in ambiente di laboratorio virtuale, un'architettura client server in cui un client con indirizzo 192.168.32.101 (Windows) richiede tramite web browser una risorsa all'hostname epicode.internal che risponde all'indirizzo 192.168.32.100 Kali.
- 2) Si intercetti poi la comunicazione con Wireshark, evidenziando i MAC address di sorgente e destinazione ed il contenuto della richiesta HTTPS.
- 3) Ripetere l'esercizio, sostituendo il server HTTPS, con un server HTTP.
- 4) Si intercetti nuovamente il traffico, evidenziando le eventuali differenze tra il traffico appena catturato in HTTP ed il traffico precedente in HTTPS.

Svolgimento :

- 1) Impostiamo l'ip fisso dettato dall'esercizio sull'ambiente Linux
 - a) accendere la macchina Kali , aprire una shell da root e digitare
nano /etc/network/interfaces
- 2) Modificare il file di config come segue



```
root@kali: /etc/network/interfaces.d
File Actions Edit View Help
GNU nano 8.3 /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

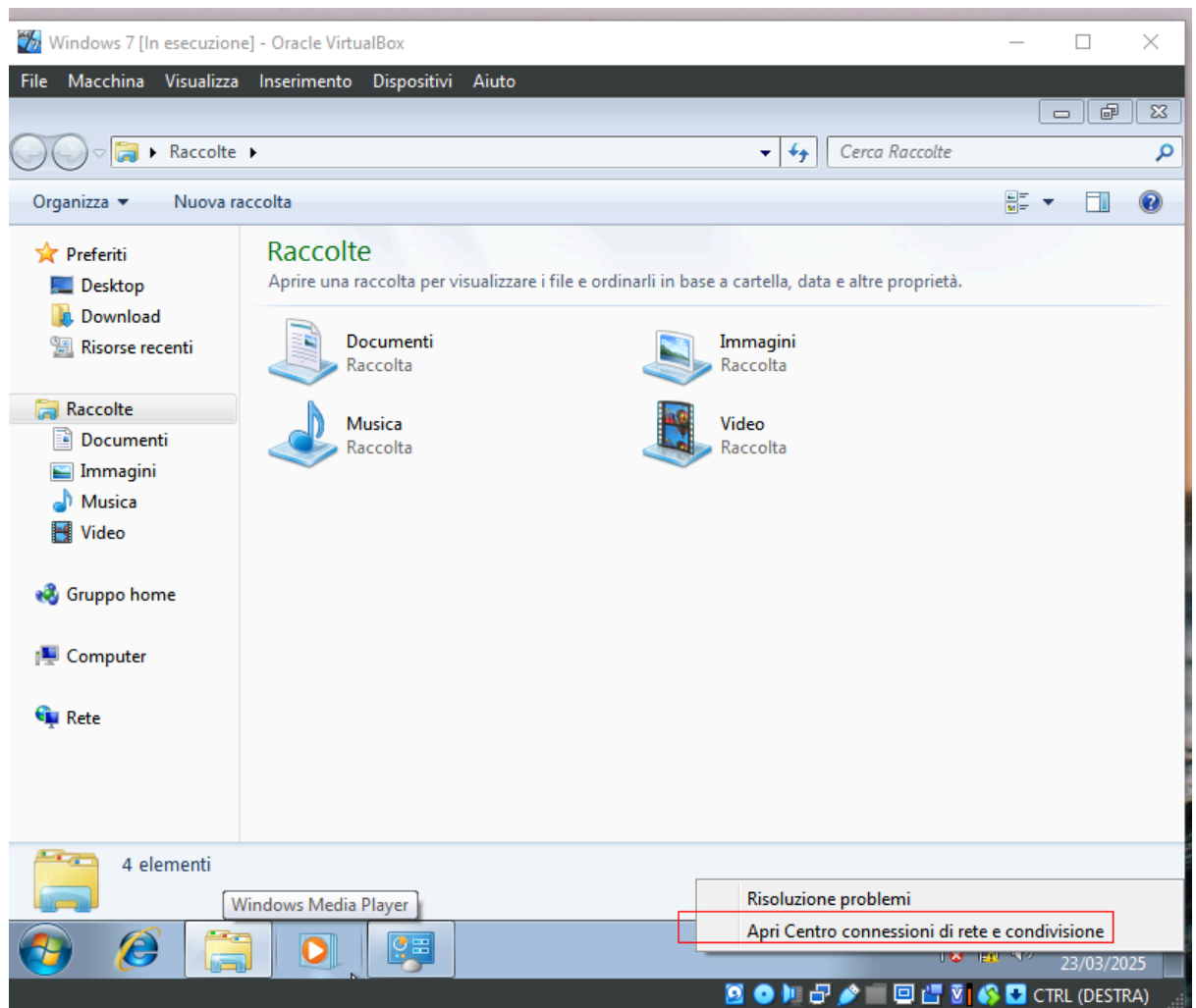
source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

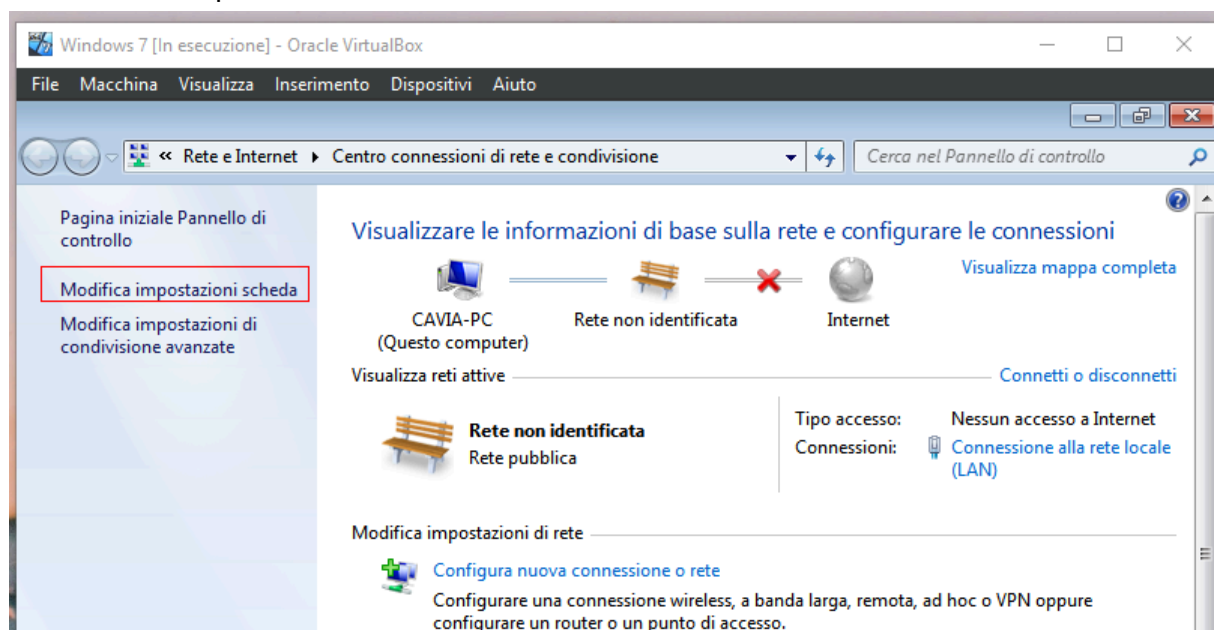
auto eth0
iface eth0 inet static
address 192.168.32.100/24
gateway 192.168.32.1

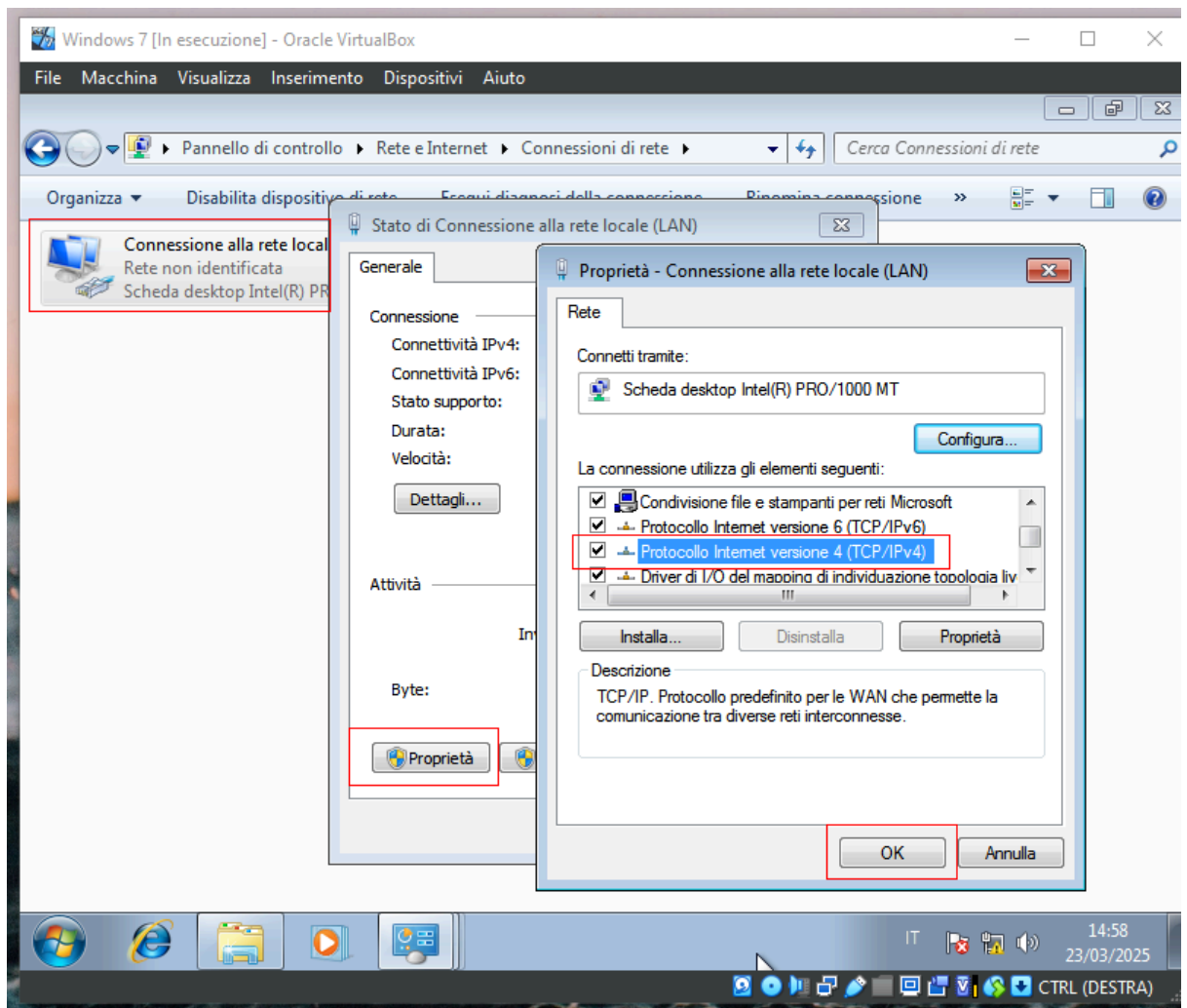
[ Read 15 lines ]
^G Help      ^O Write Out ^F Where Is  ^K Cut       ^T Execute   ^C Location  M-U Undo
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify   ^_ Go To Line M-E Redo
```

- 3) Salvare e riavviare la macchina Linux per applicare le modifiche.
- 4) Aprire l'ambiente virtuale Windows 7 e aprire " impostazione rete e condivisione " facendo tasto destro sull'icona della network in basso a destra.

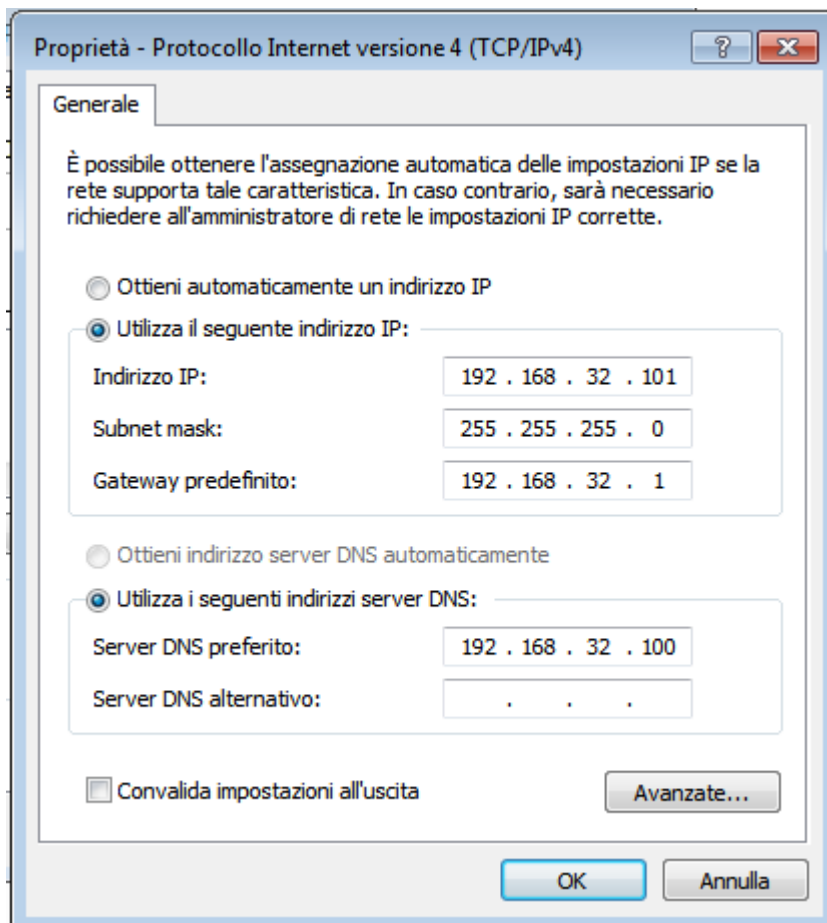


Poi “modifica opzioni scheda”





Impostare la scheda di rete coi seguenti parametri



Indirizzo Ip fisso da traccia + indirizzo IP DNS uguale a indirizzo IP di Linux per forwardare richieste DNS tramite simulazione Inetsim su Kali.

- 5) Tornare su Kali per impostare InetSim in modo che funga sia da web server HTTPS inizialmente, poi HTTP , e DNS.
- 6) Aprire un terminale da root e digitare **nano /etc/inetsim/inetsim.conf** (andiamo nel file di config. di Inetsim) e impostiamo tutti i parametri necessari come segue:

```
zsh: corrupt history file /home/kali/.zsh_history
root@kali: /home/kali
File Actions Edit View Help
GNU nano 8.3 /etc/inetsim/inetsim.conf
# The services to start
# Syntax: start_service <service name> /home/kali
#
# Default: none
#
# Available service names are:
# dns, http, smtp, pop3, tftp, ftp, ntp, time_tcp,
# time_udp, daytime_tcp, daytime_udp, echo_tcp,
# echo_udp, discard_tcp, discard_udp, quotd_tcp,
# quotd_udp, chargen_tcp, chargen_udp, finger,
# ident, syslog, dummy_tcp, dummy_udp, smtps, pop3s,
# ftps, irc, https
#
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
```

Abilitiamo i servizi DNS e HTTPS
cavando i cancelletti davanti ai
rispettivi nomi. In questo modo
non saranno più visti come
commenti dal programma

^G Help ^O Write Out ^F Where Is ^K Cut ^T Execute ^C Location M-U Undo
^X Exit ^R Read File ^V Replace ^U Paste ^J Justify ^_ Go To Line M-E Redo

DNS:

```
root@kali: /etc/network/interfaces.d
#####
#service_bind_address
#
# IP address to bind services to
#
# Syntax: service_bind_address <IP address>
#
# Default: 127.0.0.1
#
service_bind_address 0.0.0.0
#####
# service_run_as_user
#
```

Il parametro `service_bind_address` è utilizzato per specificare l'indirizzo IP locale su cui un servizio (come DNS, HTTP, ecc.) deve essere "bindato" o associato. Questo significa che il servizio ascolterà solo le connessioni in arrivo su quell'indirizzo IP specifico.

In questo caso ho impostato `0.0.0.0` pertanto in servizio è in ascolto su tutte le interfacce di rete del sistema.

```
#####
# dns_default_ip
#
# Default IP address to return with DNS replies
#
# Syntax: dns_default_ip <IP address>
#
# Default: 127.0.0.1
#
dns_default_ip 192.168.32.100
#####
```

Impostato l'IP statico del server DNS simulato . Questo IP andrà riportato sugli altri client della rete.

```
#####
# dns_static
#
# Static mappings for DNS
#
# Syntax: dns_static <fqdn hostname> <IP address>
#
# Default: none
#
dns_static epicode.internal 192.168.32.100
#dns_static ns1.foo.com 10.70.50.30
#dns_static ftp.bar.net 10.10.20.30
#####
# dns_version
```

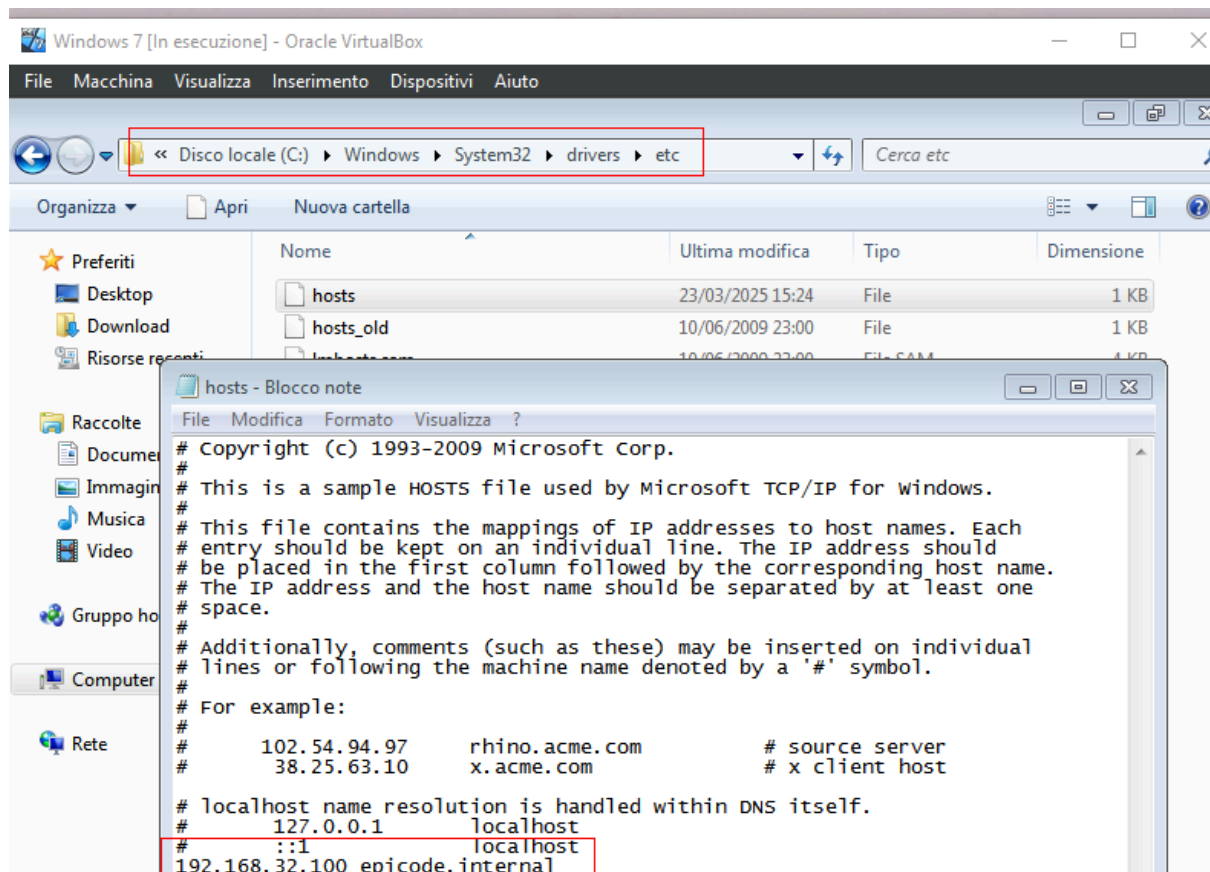
Impostato l'associazione statica nome → indirizzo come da traccia.

I parametri del servizio HTTPS rimangono quelli di Default.

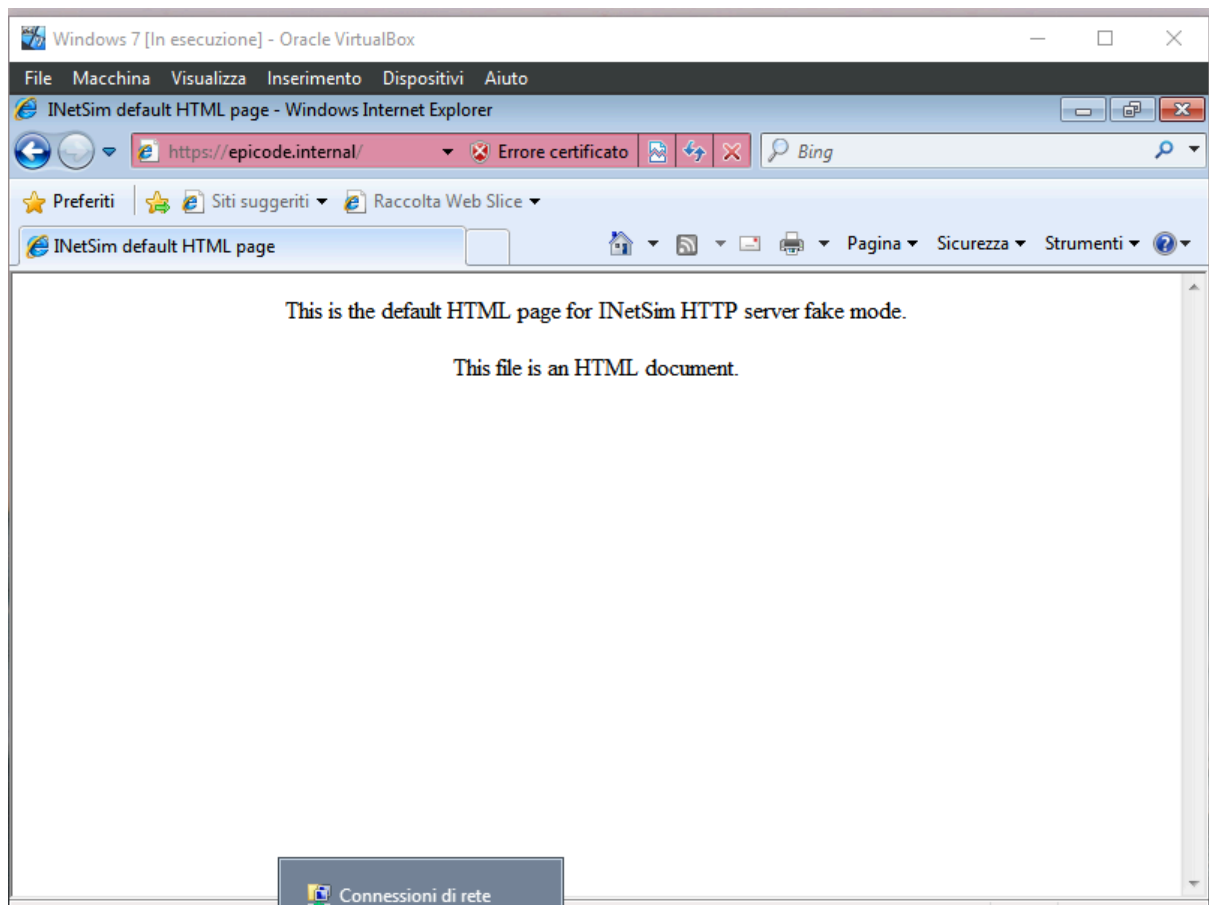
Avviamo il servizio InetSim , come segue

```
(root@kali)-[/etc/network/interfaces.d]
# /usr/bin/inetsim
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 17682) ==
Session ID: 17682
Listening on: 0.0.0.0
Real Date/Time: 2025-03-24 03:13:44
Fake Date/Time: 2025-03-24 03:13:44 (Delta: 0 seconds)
Forking services...
* dns_53_tcp_udp - started (PID 17692)
  deprecated method; prefer start_server() at /usr/share/perl5/INetSim/DNS.pm line 70.
  Attempt to start Net::DNS::Nameserver in a subprocess at /usr/share/perl5/INetSim/DNS.pm line 70.
* https_443_tcp - started (PID 17693)
  done.
Simulation running.
```

Dopo diverse prove mi son accorto che non funzionava allora ho modificato il file hosts sulla macchina windows e ho utilizzato lei come DNS.



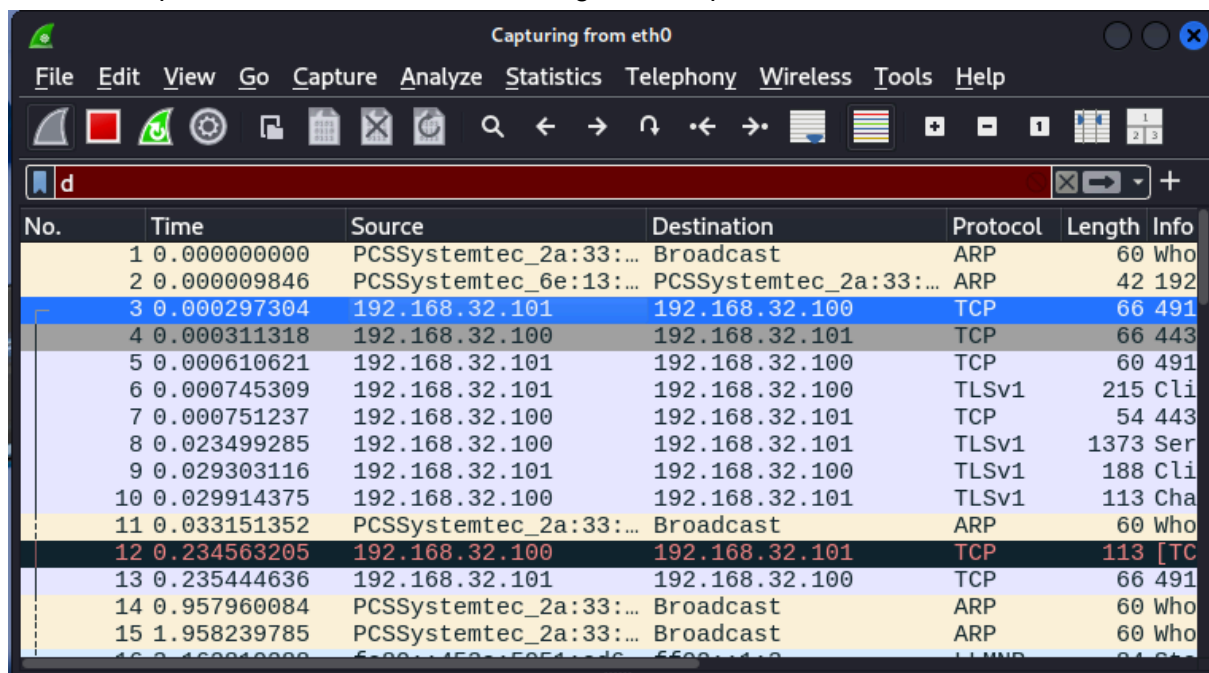
Aprimo il browser su Windows e cerchiamo nella barra di ricerca <https://epicode.internal/>



7) Simulazione con Wireshark

Sniffing del traffico su eth0

Prendiamo il pacchetto selezionato nell'immagine sotto per analisi



Analizzando il pacchetto sul layer 2 si vedono source e destination MAC delle schede di rete delle due macchine virtuali come indicato dalla traccia

```
[Protocols in Frame: eth.etherType.ip.tcp]
[Coloring Rule Name: TCP SYN/FIN]
[Coloring Rule String: tcp.flags & 0x02 || tcp.flags.fin == 1]
➤ Ethernet II, Src: PCSSystemtec_2a:33:31 (08:00:27:2a:33:31), Dst: PCSSyste
  ➤ Destination: PCSSystemtec_6e:13:6e (08:00:27:6e:13:6e)
  ➤ Source: PCSSystemtec_2a:33:31 (08:00:27:2a:33:31)
  Type: IPv4 (0x0800)
  [Stream index: 1]
➤ Internet Protocol Version 4, Src: 192.168.32.101, Dst: 192.168.32.100
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not Set)
```

Qui si vede il contenuto

3 0.00029384	192.168.32.101	192.168.32.100	TCP	66 49182 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=4 SACK_PERM
4 0.000311318	192.168.32.100	192.168.32.101	TCP	66 443 → 49182 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=12
5 0.000610621	192.168.32.101	192.168.32.100	TCP	66 49182 → 443 [ACK] Seq=1 Ack=1 Win=65700 Len=0
6 0.000745309	192.168.32.101	192.168.32.100	TLSv1	215 Client Hello (SNI=epicode.internal)
7 0.000751237	192.168.32.100	192.168.32.101	TCP	54 443 → 49182 [ACK] Seq=1 Ack=162 Win=64128 Len=0
8 0.023499285	192.168.32.100	192.168.32.101	TLSv1	1373 Server Hello, Certificate, Server Key Exchange, Server Hello Done
9 0.029303116	192.168.32.101	192.168.32.100	TLSv1	188 Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
10 0.029914375	192.168.32.100	192.168.32.101	TLSv1	113 Change Cipher Spec, Encrypted Handshake Message
11 0.033151352	PCSSystemtec_2a:33:31	Broadcast	ARP	60 Who has 192.168.32.1? Tell 192.168.32.101
12 0.234563205	192.168.32.100	192.168.32.101	TCP	113 [TCP Retransmission] 443 → 49182 [PSH, ACK] Seq=1320 Ack=296 Win=64128 Len=
13 0.235444636	192.168.32.101	192.168.32.100	TCP	66 49182 → 443 [ACK] Seq=296 Ack=1379 Win=64320 Len=0 SLE=1320 SRE=1379

Start servizio HTTP da Inetsim e disable di quello HTTPS

```
File Actions Edit View Help
GNU nano 8.3 /etc/inetsim/inetsim.conf *
# Default: none
#
# Available service names are:
# dns, http, smtp, pop3, tftp, ftp, ntp, time_tcp,
# time_udp, daytime_tcp, daytime_udp, echo_tcp,
# echo_udp, discard_tcp, discard_udp, quotd_tcp,
# quotd_udp, chargen_tcp, chargen_udp, finger,
# ident, syslog, dummy_tcp, dummy_udp, smtps, pop3s,
# ftps, irc, https
#
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
```

Sniffing del traffico e si vedono tutte le chiamate in chiaro , senza encryption

1	0.000000000	192.168.32.101	192.168.32.100	TCP	66 49185 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM
2	0.000024532	192.168.32.100	192.168.32.101	TCP	66 80 → 49185 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
3	0.000482348	192.168.32.101	192.168.32.100	TCP	60 49185 → 80 [ACK] Seq=1 Ack=1 Win=65700 Len=0
4	0.000597625	192.168.32.101	192.168.32.100	TCP	54 80 → 49185 [ACK] Seq=1 Ack=308 Win=64128 Len=0
5	0.000566280	192.168.32.100	192.168.32.101	TCP	54 80 → 49185 [ACK] Seq=1 Ack=308 Win=64128 Len=0
6	0.027553952	192.168.32.100	192.168.32.101	TCP	204 80 → 49185 [PSH, ACK] Seq=1 Ack=308 Win=64128 Len=150 [TCP PDU reassembled in 7]
7	0.030139501	192.168.32.100	192.168.32.101	HTTP	312 HTTP/1.1 200 OK (text/html)
8	0.030438426	192.168.32.101	192.168.32.100	TCP	60 49185 → 80 [ACK] Seq=308 Ack=410 Win=65292 Len=0
9	0.030438591	192.168.32.101	192.168.32.100	TCP	60 49185 → 80 [FIN, ACK] Seq=308 Ack=410 Win=65292 Len=0
10	0.030461417	192.168.32.100	192.168.32.101	TCP	54 80 → 49185 [ACK] Seq=410 Ack=309 Win=64128 Len=0
11	0.041286178	192.168.32.101	192.168.32.100	DNS	77 Standard query 0x7abc A urs.microsoft.com
12	0.041305240	192.168.32.100	192.168.32.101	ICMP	105 Destination unreachable (Port unreachable)
13	0.041833886	192.168.32.101	192.168.32.100	DNS	77 Standard query 0xddfe A urs.microsoft.com
+ Frame 4: 361 bytes on wire (2888 bits), 361 bytes captured (2888 bits) on interface eth0, id 0					0000 08 00 27 6e 13 6e 08 00 27 2a 33 31 08 00 45 00 ...n n...''31 E
+ Ethernet II, Src: PCSSystemtec_2a:33:31 (08:00:27:2a:33:31), Dst: PCSSystemtec_6e:13:6e (08:00:27:6e:13:6e)					0010 01 5b 04 af 49 00 00 06 32 d4 c0 a0 20 65 c0 a8 ...[@... 2 ...e
+ Internet Protocol Version 4, Src: 192.168.32.101, Dst: 192.168.32.100					0020 20 64 c0 21 00 50 03 95 c9 24 31 9f 4e 6e 50 18 ...d i p...31 NIP
+ Transmission Control Protocol, Src Port: 49185, Dst Port: 80, Seq: 1, Ack: 1, Len: 307					0030 40 29 cb 06 00 00 47 45 54 20 2f 20 4b 54 54 50 ...@)...GE T / HTTP
+ Hypertext Transfer Protocol					0040 2f 31 2e 31 0d 0a 41 63 63 65 70 74 3a 20 2a 2f .../1..Ac cept: */
					0050 2a 0d 0a 41 63 63 65 70 74 2d 4c 61 6e 67 75 61 ...* Accep t-Langua
					0060 67 05 3a 20 69 74 2d 49 54 0d 0a 55 73 65 72 2d ...ge: it-1 T User-
					0070 41 67 65 6e 74 3a 20 4d 6f 7a 69 6c 6c 61 2f 34 ...Agent: M ozilla/4
					0080 2e 30 20 28 63 6f 6d 70 61 74 69 62 6c 65 3b 20 ..._0 (comp atible;
					0090 4d 53 49 45 20 38 2e 30 3b 20 57 69 6e 64 6f 77 ...MSIE 8.0 ; Window
					00a0 73 20 4e 54 20 36 2e 31 3b 20 57 4f 57 36 34 3b ...s NT 6.1 ; WOW64;
					00b0 20 54 72 69 64 65 6e 74 2f 34 2e 30 3b 20 53 4c ...Trident /4.0; SL
					00c0 43 43 32 3b 20 2e 4e 45 54 20 43 4c 52 20 32 2e ...CG2;_NE T CLR 2.
					00d0 30 2e 35 30 37 32 37 3b 20 2e 4e 45 54 20 43 4c ...0.58727;_NET CL
					00e0 52 20 33 2e 35 2e 33 30 37 32 39 3b 20 2e 4e 45 ...R 3.5.30 729;_NE
					00f0 54 20 43 4c 52 20 33 2e 30 2e 33 30 37 32 39 3b ...T CLR 3. 0.30729;
					0100 20 4d 65 64 69 61 20 43 65 6e 74 65 72 20 50 43 ...Media C enter pc
					0110 20 36 2e 30 20 0d 0a 41 63 63 65 70 74 2d 45 6e ...6.0) Ac cept:En
					0120 63 6f 64 69 6e 67 3a 20 67 7a 69 70 2c 20 64 65 ...coding: gzip, de
					0130 66 6c 61 74 65 0d 0a 48 6f 73 74 3a 20 65 70 69 ...flate_H ost: epi
					0140 63 6f 64 65 2c 69 6e 74 65 72 6e 61 6c 0d 0a 83 ...code.int ernal;_
					0150 8f 6e 0e 03 43 74 69 6f 6a 3a 20 4b 08 05 70 2d ...connectio n: Keep-
					0160 41 6c 69 76 65 0d 0a 0d 0a ...Alive.

Ecco le principali differenze tra HTTP e HTTPS:

- **Sicurezza:** HTTPS utilizza protocolli di sicurezza come SSL/TLS per crittografare i dati .

Questo impedisce a terzi non autorizzati di intercettare e leggere informazioni sensibili. HTTP, invece, trasmette i dati in chiaro, rendendoli vulnerabili all'intercettazione.

- **Crittografia:** HTTPS utilizza la crittografia per proteggere i dati trasmessi tra il client e il server. I protocolli SSL/TLS utilizzano chiavi crittografiche per cifrare e convalidare i dati.
- **Porta:** HTTP utilizza la porta 80, mentre HTTPS utilizza la porta 443.