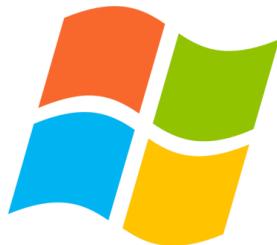


# Laboratorio Virtuale

## CSPT0324 Modulo 1



Windows 7

1

Yilei Wu

19 Luglio 2024

# Indice

Consegna dell'esercizio.....	3
Architettura del laboratorio virtuale e della rete interna.....	3
Requisiti.....	4
Configurazione di Kali Linux.....	4
Controlli preliminari su Virtual Box.....	4
Impostare l'indirizzo IP statico a 192.168.32.100 .....	5
Configurazione di Windows 7.....	7
Controlli preliminari su Virtual Box .....	7
Impostare l'indirizzo IP statico a 192.168.32.101 .....	7
Configurazione firewall di Windows 7 .....	9
Verifica e test sulla rete del laboratorio virtuale.....	11
Configurazione Client, HTTPS Server & DNS Server .....	12
Configurazione DNS di Windows 7 (Client).....	12
Configurazione Inetsim su Kali Linux (Server HTTPS & DNS).....	12
Test sulla configurazione di INetSim .....	14
Analisi e risoluzione del mancato funzionamento del server DNS.....	15
Test sulla risoluzione DNS del dominio epicode.internal .....	17
Ulteriori verifiche sulla risoluzione dei domini personalizzati.....	18
Wireshark .....	19
Analisi con protocollo HTTPS.....	19
MAC Address .....	20
Analisi contenuto richiesta HTTPS .....	21
Analisi con protocollo HTTP .....	22
Configurazione in HTTP .....	22
Analisi contenuto richiesta HTTP .....	23

## Consegna dell'esercizio

### Requisiti e servizi:

- Kali Linux: IP 192.168.32.100
- Windows 7: IP 192.168.32.101
- HTTPS server: attivo
- Servizio DNS per risoluzione nomi di dominio: attivo

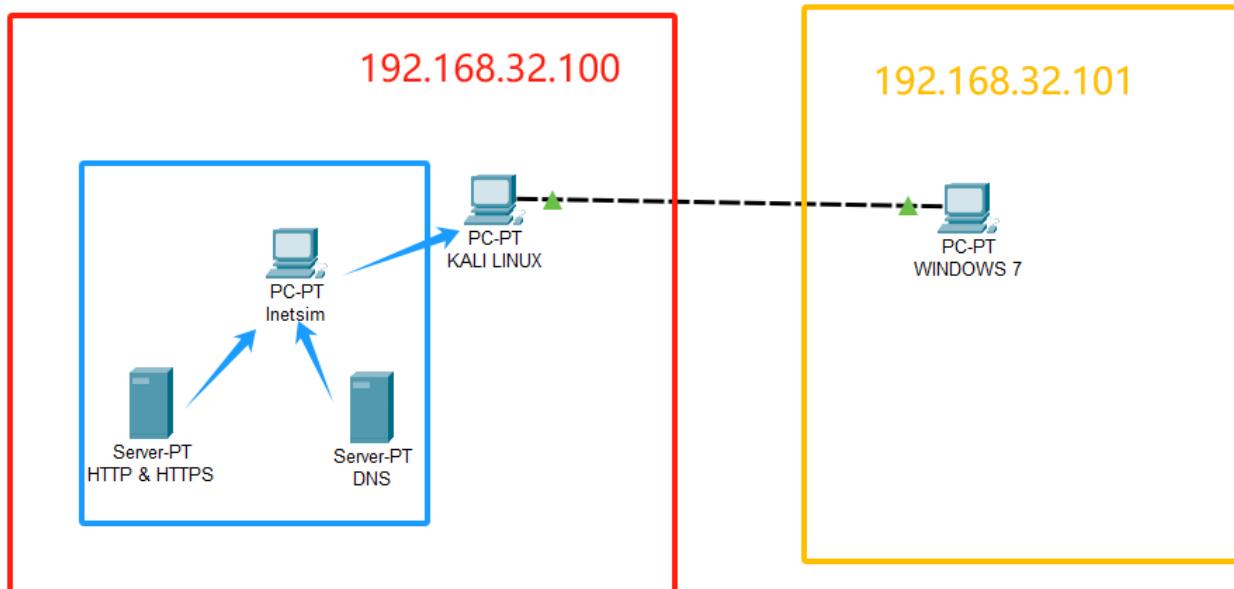
### Traccia:

Simulare, in ambiente di laboratorio virtuale, un'architettura client server in cui un client con indirizzo 192.168.32.101 (Windows 7) richieda tramite web browser una risorsa all'hostname **epicode.internal** che risponda all'indirizzo 192.168.32.100 (Kali).

Si intercetti poi la comunicazione con Wireshark, evidenziando i MAC address di sorgente e destinazione ed il contenuto della richiesta HTTS.

Ripetere l'esercizio, sostituendo il server HTTPS, con un server HTTP. Si intercetti nuovamente il traffico, evidenziando le eventuali differenze tra il traffico appena catturato in HTTP ed il traffico precedente in HTTPS. Spiegare, motivandole, le principali differenze se presenti.

### Architettura del laboratorio virtuale e della rete interna



Per una migliore comprensione della struttura si considera che Kali Linux conterrà il programma Inetsim. Inetsim avrà diversi compiti chiavi nella rete: il server http e https (caricare le pagine web) e il servizio DNS (tradurre il dominio nell'indirizzo IP del server).

Windows 7 invece è il nostro cliente, dove partiranno tutte le interrogazioni al server.

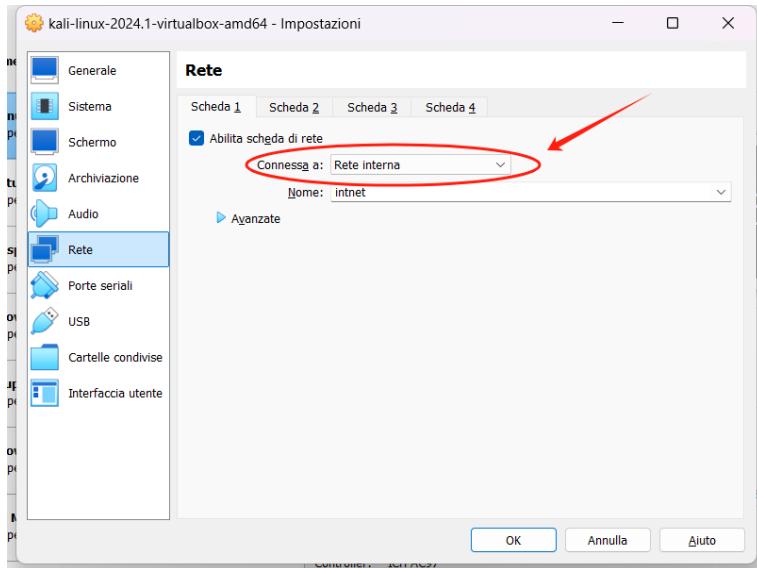
# Requisiti

## Configurazione di Kali Linux

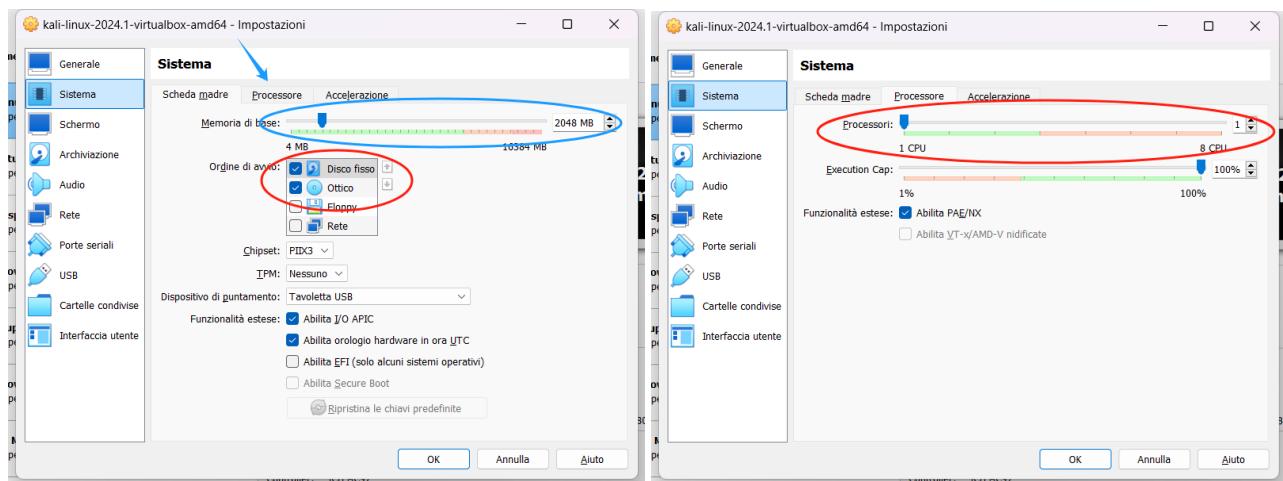
### Controlli preliminari su Virtual Box

Fondamentali sono i controlli prima di avviare la macchina virtuale. Come da consegna la rete su cui opereremo sarà una rete interna con indirizzo ip gateway 192.168.32.1.

Per cui controlliamo se sia impostata la rete interna come da figura.



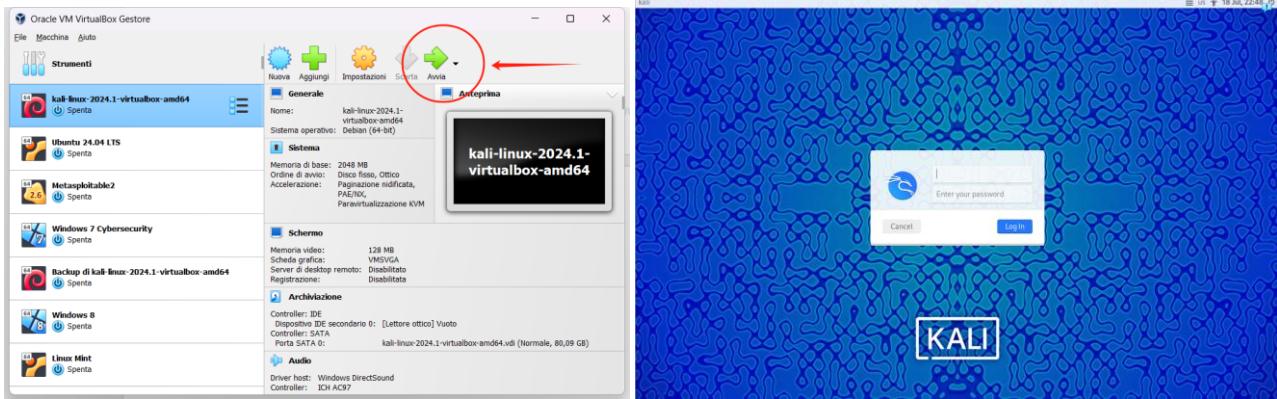
Inoltre per evitare possibili problemi dopo l'avvio della macchina virtuale per compatibilità si consiglia di disattivare "Floppy", una tecnologia nel 2024, ormai da ritenersi legacy.



Nella stessa schermata verifichiamo anche che ci sia abbastanza memoria RAM dedicata e che ci siano abbastanza CPU assegnate.

In questo caso trattasi di Kali Linux che non ha molti requisiti per il funzionamento, pertanto quanto assegnato in immagine qui sopra è più che sufficiente.

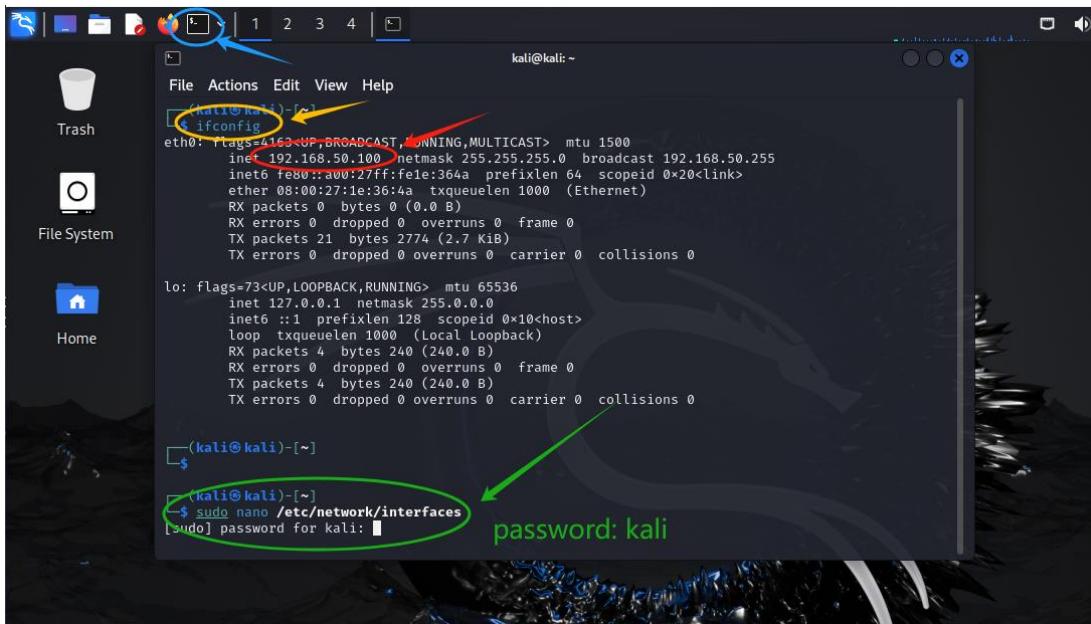
Terminati i controlli possiamo proseguire con l'avvio della macchina virtuale Kali Linux.



Dopo l'avvio ci chiede le credenziali di default che è "kali" sia per username, sia per password.

### Impostare l'indirizzo IP statico a 192.168.32.100

Controlliamo le informazioni relative all'indirizzo IP aprendo il terminale e inserendo il comando "ifconfig"



5

Dall'immagine vediamo che l'indirizzo IP attuale è 192.168.50.100 e per modificarlo lanciamo il comando "sudo nano /etc/network/interfaces", se ci chiede la password, come da immagine, questa è "kali" e si preme invio.

```

GNU nano 8.0                               /etc/network
# This file describes the network interfaces available
# and how to activate them. For more information, see
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

```

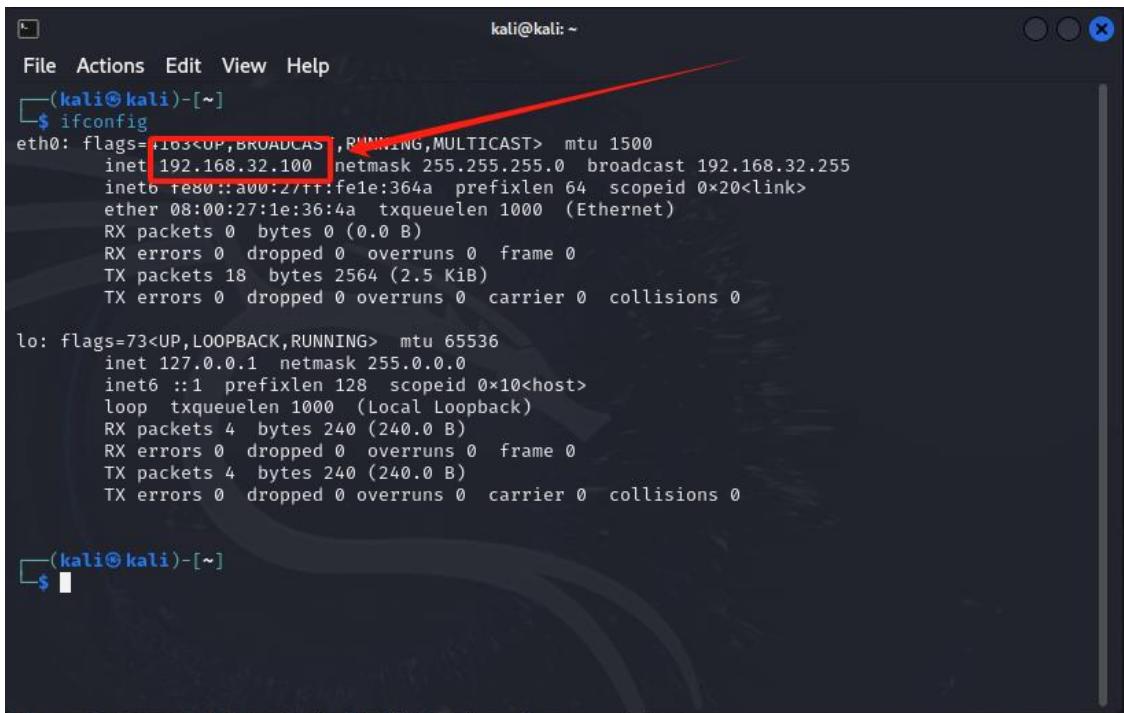
Come da immagine dobbiamo andare a modificare dopo “inet” se fosse stato in DHCP sostituirlo con “static” e impostare address 192.168.32.100. Lo slash /24 indica la subnet 255.255.255.0.

Digitato la nuova configurazione per salvare premiamo CTRL+O e Invio, poi per uscire CTRL+X. Lanciamo il comando “sudo reboot” per riavviare il sistema e applicare le modifiche.



```
(kali㉿kali)-[~]
$ sudo reboot
```

Una volta riavviato riapriamo il terminale e lanciamo il comando “ifconfig” per verificare che sia stata applicata il nuovo indirizzo IP.



```
kali@kali: ~
File Actions Edit View Help
(kali㉿kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.32.100 netmask 255.255.255.0 broadcast 192.168.32.255
        inet6 fe80::a00:2ff:fe1e:364a prefixlen 64 scopeid 0x20<link>
          ether 08:00:27:1e:36:4a txqueuelen 1000 (Ethernet)
            RX packets 0 bytes 0 (0.0 B)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 18 bytes 2564 (2.5 KiB)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
      inet 127.0.0.1 netmask 255.0.0.0
        inet6 ::1 prefixlen 128 scopeid 0x10<host>
          loop txqueuelen 1000 (Local Loopback)
            RX packets 4 bytes 240 (240.0 B)
            RX errors 0 dropped 0 overruns 0 frame 0
            TX packets 4 bytes 240 (240.0 B)
            TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

(kali㉿kali)-[~]
$
```

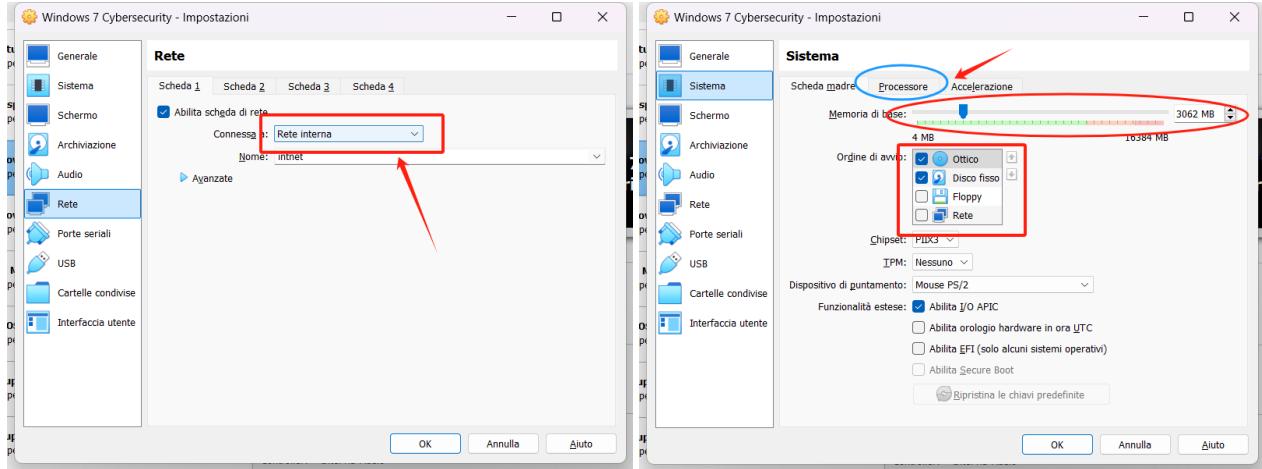
6

Perfetto, indirizzo IP statico di Kali Linux modificato con successo.

# Configurazione di Windows 7

## Controlli preliminari su Virtual Box

Anche per questa macchina virtuale è necessario fare delle verifiche preliminari prima di avviarla.



Ci assicuriamo che sia configurato con la rete interna, che sia deselezionato il "Floppy" per evitare problemi di compatibilità, che ci sia abbastanza Ram e CPU assegnati. Per Windows 7 è consigliabile almeno 2 CPU e 2 GB Ram o superiore.

## Impostare l'indirizzo IP statico a 192.168.32.101

Avviamo la macchina virtuale e ci dirigiamo subito in basso a sinistra sull'inconso Windows e cerchiamo "cmd" e per evitarcene ogni possibile limitazione, lo avviamo con i privilegi di amministratore.

Dal terminale cmd lanciamo il comando "ip config/all" per ottenere le informazioni più complete relativo all'indirizzo IP che in questo caso era stato configurato a 192.168.50.102.

```
Administrator: C:\Windows\System32\cmd.exe
Microsoft Windows [Versione 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Tutti i diritti riservati.

C:\Windows\system32>ipconfig /all

Configurazione IP di Windows

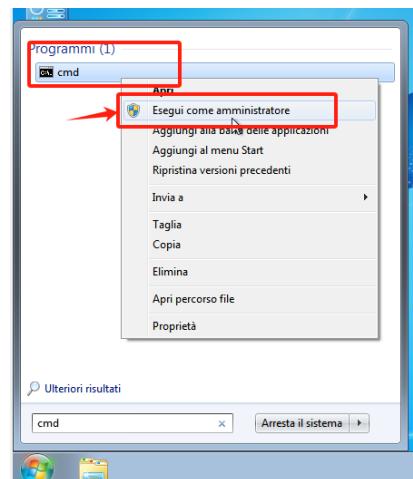
Nome host . . . . . : Corso-PC
Suffisso DNS primario . . . . . : 
Tipo nodo . . . . . : Ibrido
Routing IP abilitato. . . . . : No
Proxy WINS abilitato . . . . . : No

Scheda Ethernet Connessione alla rete locale <LAN>:
  Suffisso DNS specifico per connessione: 
  Descrizione . . . . . : Scheda desktop Intel(R) PRO/1000 MT
  Indirizzo fisico . . . . . : 08-00-27-37-B4-EB
  DHCP abilitato . . . . . : No
  Configurazione automatica abilitata . . . . . : Sì
  Indirizzo IPv6 locale rispetto al collegamento . . . fe80::385a:72eb:edd3:b128%11<referenziale>
  Indirizzo IPv4 . . . . . : 192.168.50.102<referenziale>
  Subnet mask . . . . . : 255.255.255.0
  Gateway predefinito . . . . . : 192.168.50.1
  ID GUID DHCPv6 . . . . . : 235405351
  DUID Client DHCPv6 . . . . . : 00-01-00-01-2E-0B 95-A6-08-00-27-37-B4-EB

  Server DNS . . . . . : 1.1.1.1
  1.0.0.1
  NetBIOS su TCP/IP . . . . . : Attivato

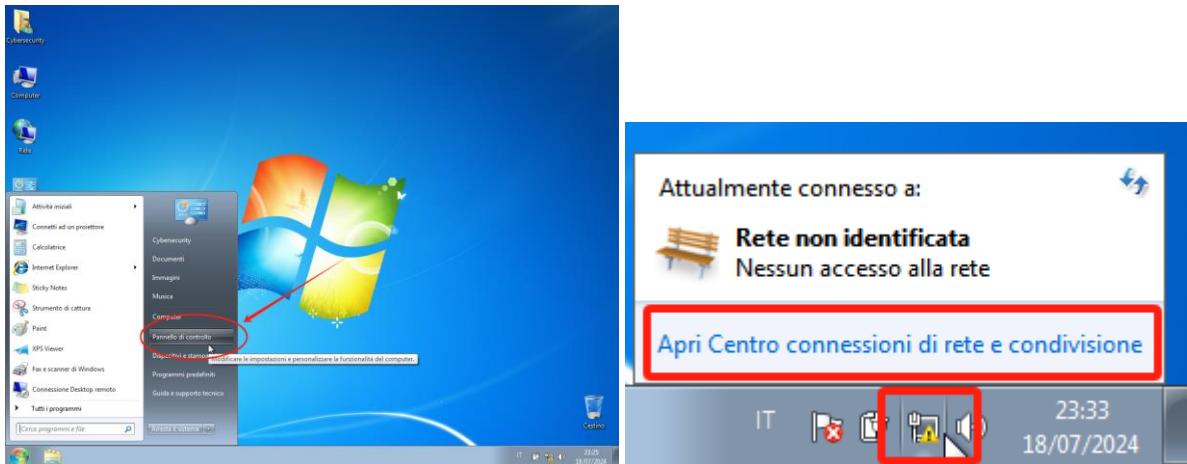
Scheda Tunnel isatap.<D14DF312-15C0-4370-9966-C43F9621BD3C>:
  Stato supporto . . . . . : Supporto disconnesso
  Suffisso DNS specifico per connessione: 
  Descrizione . . . . . : Microsoft ISATAP Adapter
  Indirizzo fisico . . . . . : 00-00-00-00-00-00-E0
  DHCP abilitato . . . . . : No
  Configurazione automatica abilitata . . . . . : Sì

C:\Windows\system32>
```

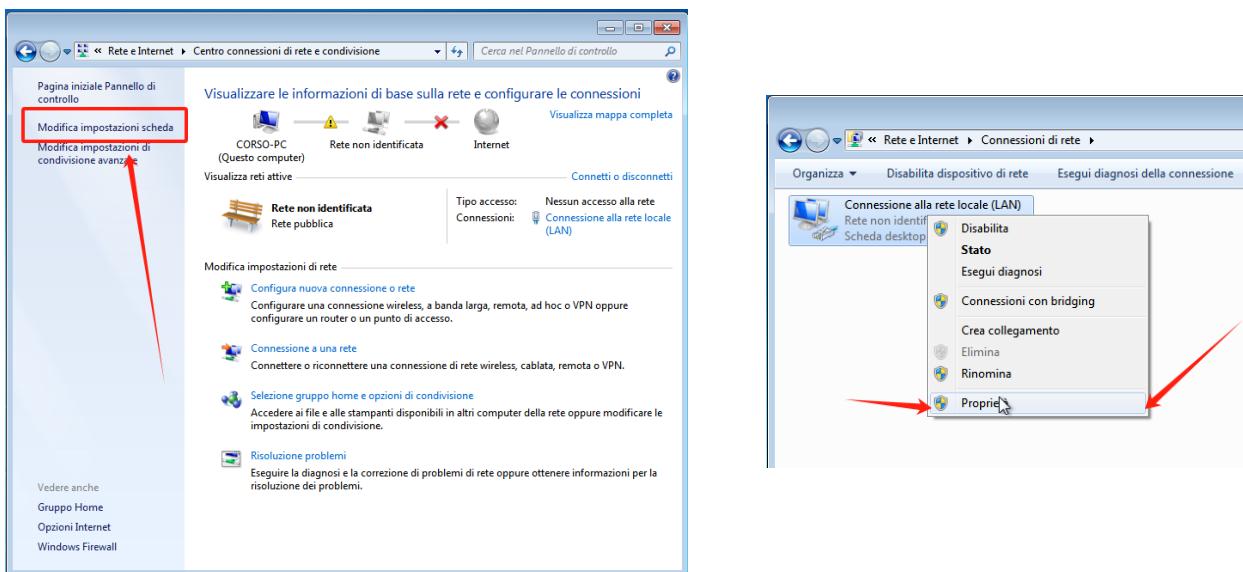


Noi abbiamo bisogno di configurarlo con IP statico a 192.168.32.101.

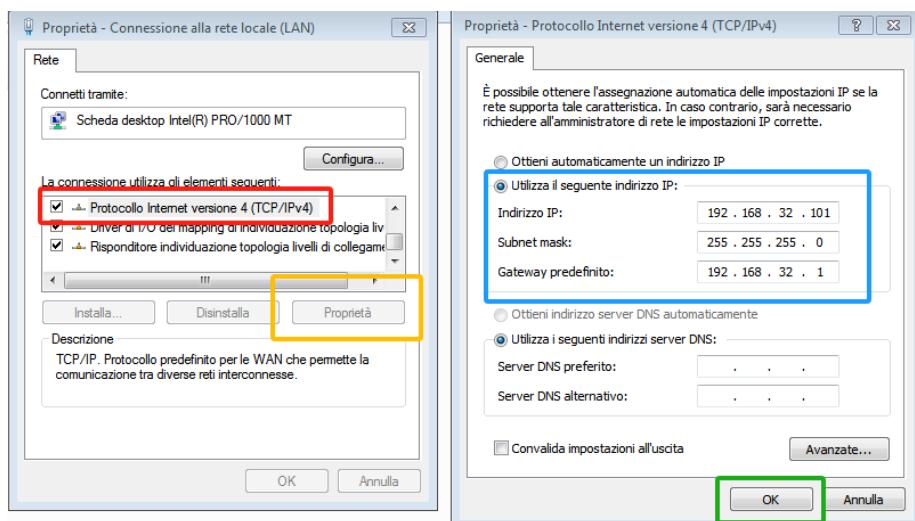
Per impostare l'indirizzo IP statico in Windows 7 apriamo l'icona Windows in basso a sinistra e rechiamoci su Pannello di Controllo > Rete e Internet > Centro connessioni di rete e condivisione oppure l'altro modo più veloce per raggiungere questa schermata è cliccare l'icona di rete in basso a destra col tasto destro del mouse e cliccare su "Apri Centro connessioni di rete e condivisione".



Andiamo su modifica impostazioni scheda e poi tasto destro del mouse sulla scheda di rete e poi proprietà.



Scorriamo per trovare IPv4 > Proprietà e configuriamo l'indirizzo IP come da immagine.



Inseriamo la configurazione 192.168.32.101, subnet mask 255.255.255.0 e Gateway predefinito 192.168.32.1 poi clicchiamo Ok per salvare.

Riavviamo il sistema e facciamo una verifica di aver applicato con successo il nuovo indirizzo ip statico.

Sempre con cmd e il comando “ipconfig /all”.

```
C:\Users\Cybersecurity>ipconfig /all
Configurazione IP di Windows
  Nome host . . . . . : Corso-PC
  Suffixo DNS primario . . . . . :
  Tipo nodo . . . . . : Ibrido
  Routing IP abilitato . . . . . : No
  Proxy WINS abilitato . . . . . : No

  Scheda Ethernet Connessione alla rete locale <LAN>:
    Suffixo DNS specifico per connessione:
    Descrizione . . . . . : Scheda desktop Intel(R) PRO/1000 MT
    Indirizzo fisico . . . . . : 08-00-27-37-B4-EB
    DHCP abilitato . . . . . : No
    Configurazione automatica abilitata . . . . . : Sì
    Indirizzo IPv6 locale rispetto al collegamento . . . : fe80::385a:72eb:edd3:b128%11<Preferenziale>
    Indirizzo IPv4 . . . . . : 192.168.32.101<Preferenziale>
    Subnet mask . . . . . : 255.255.255.0
    Gateway predefinito . . . . . : 192.168.32.1
    IAID DHCPv6 . . . . . : 235405351
    DUID Client DHCPv6 . . . . . : 00-01-00-01-2E-0B-95-A6-08-00-27-37-B4-EB
    Server DNS . . . . . : fec0:0:0:ffff::1x1
                           fec0:0:0:ffff::2x1
                           fec0:0:0:ffff::3x1
    NetBIOS su TCP/IP . . . . . : Attivato

  Scheda Tunnel isatap.<D14DF312-15C0-4370-9966-C43F9621BD3C>:
    Stato supporto . . . . . : Supporto disconnesso
    Suffixo DNS specifico per connessione:
    Descrizione . . . . . : Microsoft ISATAP Adapter
    Indirizzo fisico . . . . . : 00-00-00-00-00-00-E0
    DHCP abilitato . . . . . : No
    Configurazione automatica abilitata . . . . . : Sì

C:\Users\Cybersecurity>
```

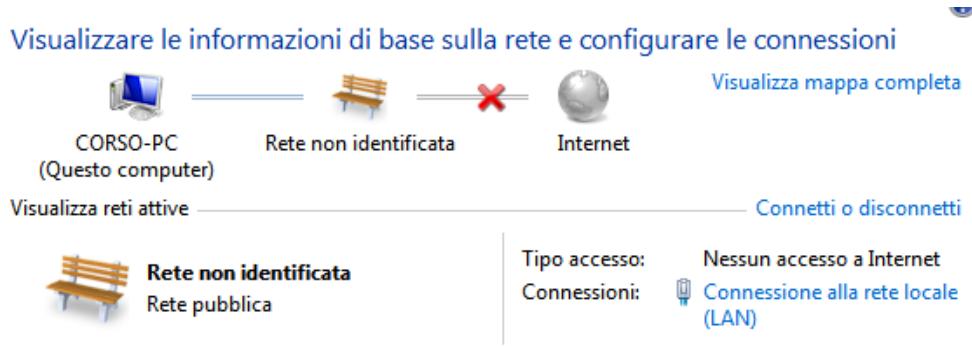
Perfetto! Abbiamo l'indirizzo IP statico impostato come da requisiti.

9

Abbiamo utilizzato il comando “ipconfig /all” per avere più informazioni, ma per questo passaggio si poteva usare anche il comando semplice “ipconfig”.

## Configurazione firewall di Windows 7

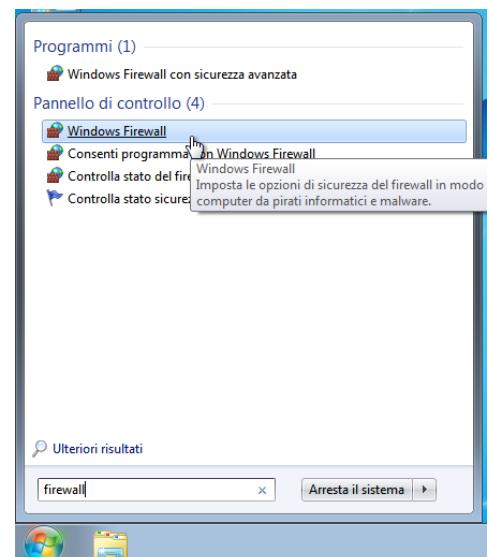
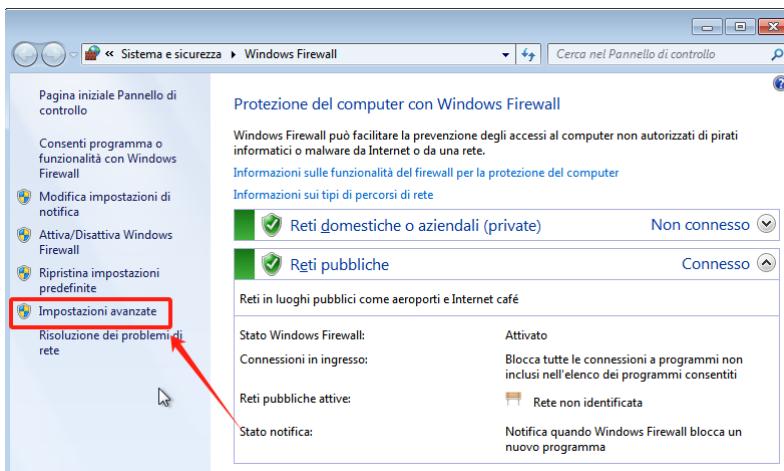
Di default in Windows 7 la nostra rete interna è considerata dal sistema operativo come una rete pubblica e nel nostro caso non possiamo modificarlo, questo implica delle restrizioni, ovvero bloccherà tutti i tentativi di connessione in entrata, compreso Kali Linux.



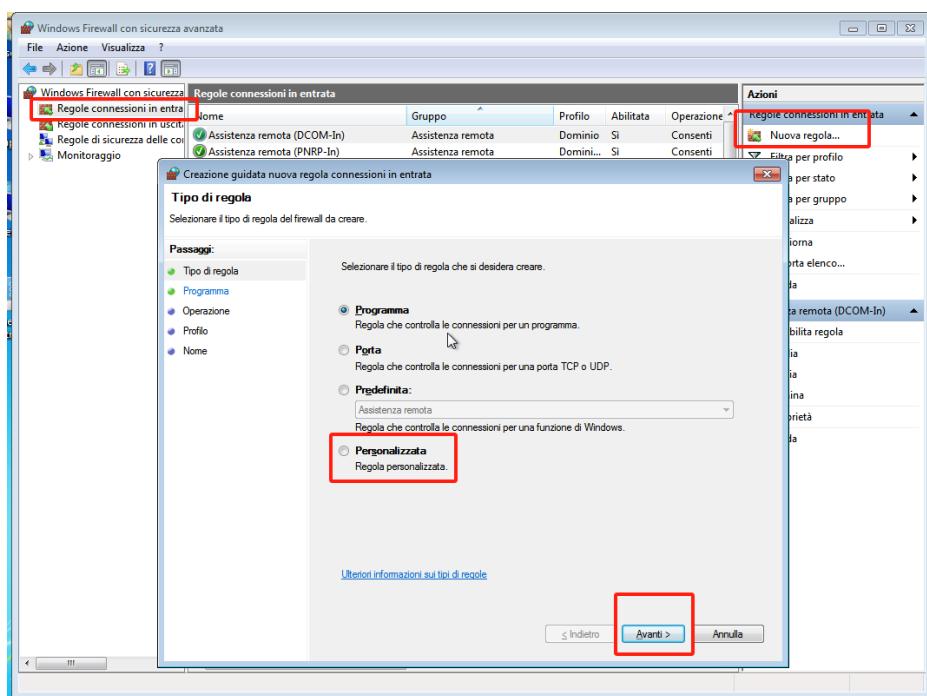
Per superare questa restrizione abbiamo bisogno di configurare, anzi creare una regola che permetta il traffico in entrata da Kali Linux.

Apriamo Windows Firewall cercandolo dalla barra di ricerca sull'Icona Windows in basso a sinistra oppure cercarlo sul pannello di controllo.

Andiamo su “Impostazioni Avanzate”



Andiamo a creare una nuova regola di connessione in entrata. Come da figura creiamo una regola personalizzata > tutti i programmi e lasciamo invariato fino alla sezione nome che daremo a nostra discrezione.



La nuova regola creata ci permetterà di ricevere il comando ping dalle altre macchine del nostro laboratorio virtuale, come vedremo nella prossima sezione.

Non è stato creato una regola più stringente sul firewall per agevolare futuri utilizzi e/o implementazione della macchina virtuale con altre macchine del laboratorio virtuale.

## Verifica e test sulla rete del laboratorio virtuale

Per verificare che le due macchine virtuali siano correttamente configurate e comunichino tra di loro utilizziamo il comando PING da terminale.

Da Kali a Windows usiamo il comando “ping 192.168.32.101”

```
kali@kali: ~
File Actions Edit View Help
└─(kali㉿kali)-[~]
$ ping 192.168.32.101
PING 192.168.32.101 (192.168.32.101) 56(84) bytes of data.
64 bytes from 192.168.32.101: icmp_seq=1 ttl=128 time=1.45 ms
64 bytes from 192.168.32.101: icmp_seq=2 ttl=128 time=0.699 ms
64 bytes from 192.168.32.101: icmp_seq=3 ttl=128 time=1.51 ms
64 bytes from 192.168.32.101: icmp_seq=4 ttl=128 time=1.67 ms
64 bytes from 192.168.32.101: icmp_seq=5 ttl=128 time=0.848 ms
64 bytes from 192.168.32.101: icmp_seq=6 ttl=128 time=0.737 ms
64 bytes from 192.168.32.101: icmp_seq=7 ttl=128 time=1.24 ms
^C
--- 192.168.32.101 ping statistics ---
7 packets transmitted, 7 received, 0% packet loss, time 6053ms
rtt min/avg/max/mdev = 0.699/1.165/1.669/0.370 ms

└─(kali㉿kali)-[~]
$
```

Collegamento effettuato con successo, il pacchetto è stato correttamente inviato a Windows 7 che ha confermato di aver ricevuto il pacchetto.

Da Windows a Kali usiamo il comando “ping 192.168.32.100”

```
C:\Users\Cybersecurity>ping 192.168.32.100

Esecuzione di Ping 192.168.32.100 con 32 byte di dati:
Risposta da 192.168.32.100: byte=32 durata=1ms TTL=64
Risposta da 192.168.32.100: byte=32 durata<1ms TTL=64
Risposta da 192.168.32.100: byte=32 durata<1ms TTL=64
Risposta da 192.168.32.100: byte=32 durata<1ms TTL=64

Statistiche Ping per 192.168.32.100:
  Pacchetti: Trasmessi = 4, Ricevuti = 4,
  Persi = 0 <0% persi>,
Tempo approssimativo percorsi andata/ritorno in millisecondi:
  Minimo = 0ms, Massimo = 1ms, Medio = 0ms

C:\Users\Cybersecurity>
```

Collegamento effettuato con successo, il pacchetto è stato correttamente inviato a Kali Linux che ha confermato di aver ricevuto il pacchetto.

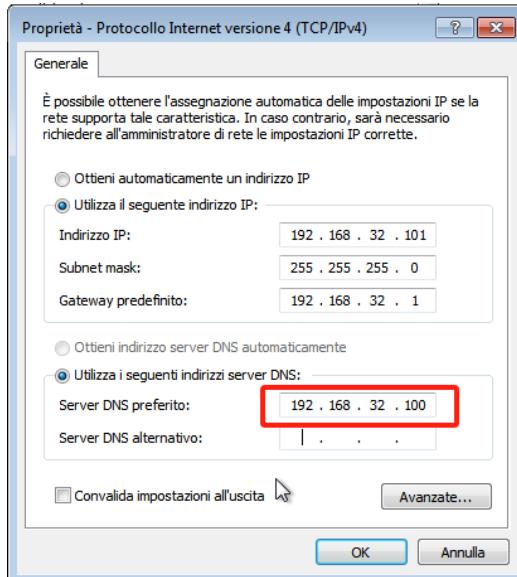
## Configurazione Client, HTTPS Server & DNS Server

Abbiamo bisogno di attivare i servizi di HTTPS come server con IP statico a 192.168.32.100 (corrispondente a Kali) per far visualizzare la pagina web e DNS per tradurre il dominio episode.internal all'indirizzo IP di Kali.

Come da traccia della consegna, quando su Windows 7 digitiamo sul browser episode.internal deve attivarsi il servizio DNS impostato su Inetsim di Kali Linux che lo indirizzerà a 192.168.32.100.

### Configurazione DNS di Windows 7 (Client)

Impostiamo l'indirizzo IP del server DNS dal client Windows 7. Torniamo sulla stessa finestra per impostare l'indirizzo IP statico, su "Server DNS preferito", immettiamo l'indirizzo IP 192.168.32.100.

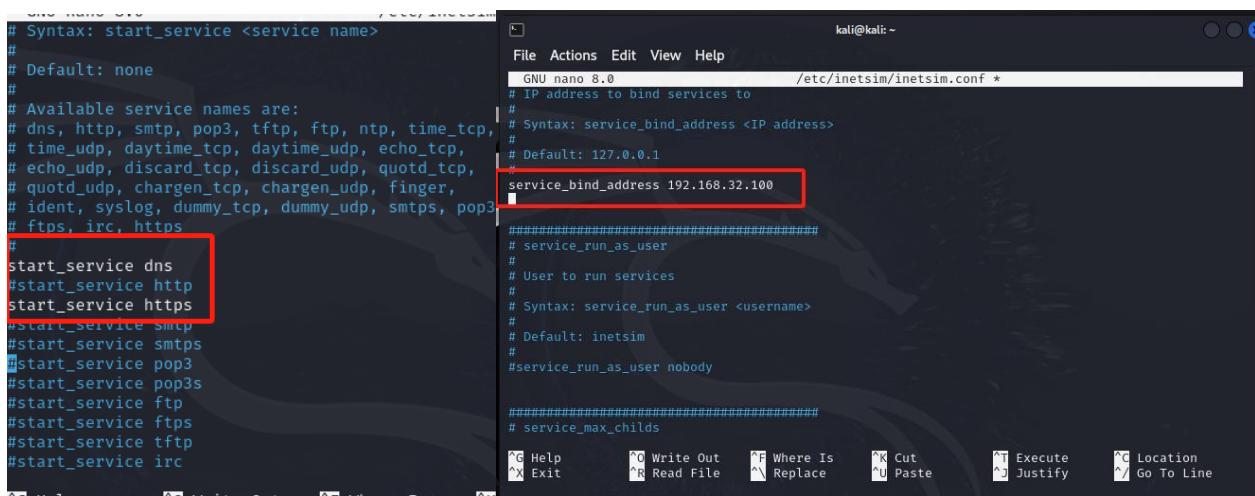


12

### Configurazione Inetsim su Kali Linux (Server HTTPS & DNS)

Apriamo il terminale in Kali Linux e avviamo la configurazione di Inetsim: dal terminale il comando "sudo nano /etc/inetsim/inetsim.conf"

Attiviamo i servizi di https e dns togliendo il commento # e impostiamo l'indirizzo IP su 192.168.32.100 poiché è l'indirizzo IP sulla quale faremo l'analisi con Wireshark. Lo stesso indirizzo IP lo impostiamo anche su dns\_default per attivare il servizio di DNS e "dns\_static episode.internal 192.168.32.100" affinché traduca solo il dominio episode.internal nell'indirizzo IP, appunto, 192.168.32.100.



```
# Syntax: start_service <service name>
#
# 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, quodt_tcp,
# quodt_udp, chargen_tcp, chargen_udp, finger,
# ident, syslog, dummy_tcp, dummy_udp, smtpt, pop3
# ftps, irc, https
#
start_service dns
#start_service http
start_service https
#start_service smtp
#start_service smtpt
#start_service pop3
#start_service pop3s
#start_service ftp
#start_service ftpt
#start_service tftp
#start_service irc
#
# IP address to bind services to
# Syntax: service_bind_address <IP address>
# Default: 127.0.0.1
#
service_bind_address 192.168.32.100
#####
# service_run_as_user
#
# User to run services
#
# Syntax: service_run_as_user <username>
# Default: inetsim
#
#service_run_as_user nobody
#####
# service_max_childs
#
# service_max_childs
```

```

GNU nano 8.0          /etc/inetsim/
# Syntax: dns_bind_port <port number>
#
# Default: 53
#
#dns_bind_port 53

#####
# 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

#####

# dns_default_hostname
#
# Default hostname to return with DNS replies

^G Help      ^O Write Out    ^F Where Is    ^K Cut
^X Exit      ^R Read File   ^M Replace   ^U Paste

```

```

GNU nano 8.0          /etc/inetsim/inetsim
#dns_default_domainname epicode.internal

#####
# dns_static
#
# Static mappings for DNS
#
# Syntax: dns_static <fqdn hostname> <IP address>
#
# Default: none
#
#dns_static www.foo.com 10.10.10.10
#dns_static ns1.foo.com 10.70.50.30
#dns_static ftp.bar.net 10.10.20.30
dns_static epicode.internal 192.168.32.100

#####

# dns_version
#
# DNS version
#
# Syntax: dns_version <version>

^G Help      ^O Write Out    ^F Where Is    ^K Cut
^X Exit      ^R Read File   ^M Replace   ^U Paste

```

Non è necessario configurare “dns\_default\_domainname epicode.internal” perché quanto già configurato è soddisfacente per la traccia. Se avessimo configurato questa impostazione, qualsiasi dominio sarebbe stato reindirizzato a 192.168.32.100, invece nel nostro caso è solo il dominio della consegna **epicode.internal**.

Per quanto riguarda il caricamento della pagina fake html sia per i protocolli https sia http sono attivi di default.

```
# https_default_fakefile sample.html text/html
```

Premiamo CTRL+O e invio per salvare e CTRL+X per tornare al terminale dove avviamo inetsim con il comando “sudo inetsim”.

```

kali㉿kali:~$ sudo inetsim
[...]
Session ID: 15610
Listening on: 0.0.0.0
Real Date/Time: 2024-07-19 17:20:52
Fake Date/Time: 2024-07-19 17:20:52 (Delta: 0 seconds)
Forking services...
  * dns_53_tcp_udp - started (PID 15610)
  * http_80_tcp - started (PID 15613)
  * https_443_tcp - started (PID 15614)
done.
Simulation running.
^C * https_443_tcp - stopped (PID 15614)
  * http_80_tcp - stopped (PID 15613)
  * http_80_tcp - stopped (PID 15610)
Simulation stopped.
== INetSim main process stopped (PID 15610) ==

Attempt to start Net::DNS::Nameserver in a subprocess at /usr/share/perl5/INetSim/DNS.pm line 69.
* http_80_tcp - started (PID 15613)
* https_443_tcp - started (PID 15614)
done.

File Actions Edit View Help
GNU nano 8.0          /usr/share/perl5/INetSim/DNS.pm *
my $uid = getpwnam($runasuser); # no servers could be reached
my $gid = getgrnam($runasgroup);
POSIX::setgid($gid); # no setup with fd00::1:eed:6fff:fe42:3b53
my $newgid = POSIX::getegid(); #epicode.internal failed: network unreachable
if ($newgid != $gid) { #no servers could be reached
    INetSim::Log::MainLog("failed! (Cannot switch group)", INetSim::Config);
    exit 0; #no setup with fd00::1:eed:6fff:fe42:3b53
} #no servers could be reached

POSIX::setuid($uid);
if ($< != $uid || $> != $uid) {
    $< = $> = $uid; # try again - reportedly needed by some Perl 5.8.0 L
    if ($< != $uid) {
        INetSim::Log::MainLog("failed! (Cannot switch user)", INetSim::Config);
        exit 0;
    }
}

$0 = 'inetns.' . INetSim::Config::getConfigParameter("DNS_ServiceName");
INetSim::Log::MainLog("started (PID $CPID)", INetSim::Config::getConfigParameter("DNS_ServiceName"));
$server->start_server(); #INetSim::Log::MainLog("started (PID $CPID)", INetSim::Config::getConfigParameter("DNS_ServiceName"));
exit 0;
}

^G Help      ^O Write Out    ^F Where Is    ^K Cut
^X Exit      ^R Read File   ^M Replace   ^U Paste

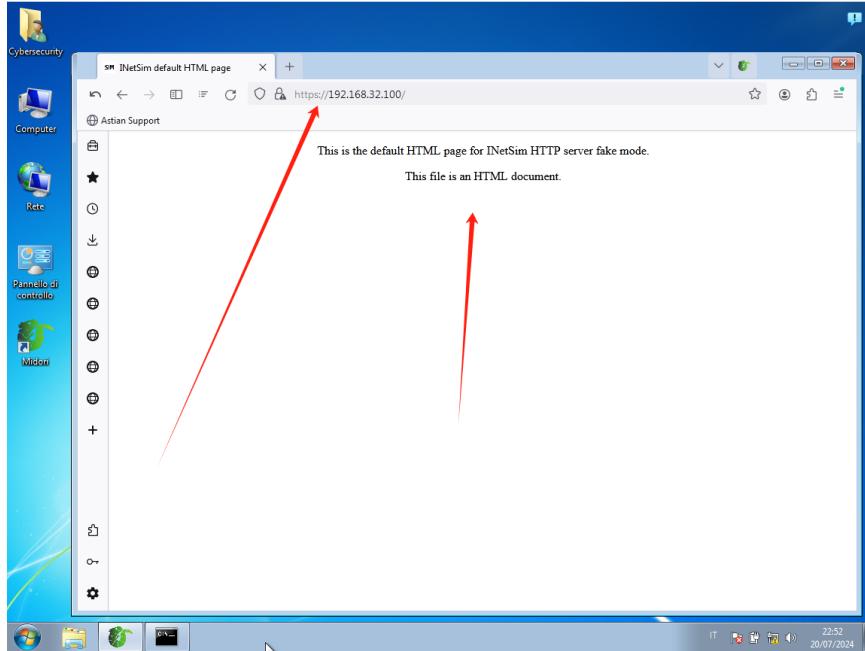
```

Dopo aver avviato c’è un suggerimento che ci consiglia di aggiornare il metodo obsoleto con quello preferenziale “start\_server()”. Per cui come suggerito apriamo il file di configurazione “sudo nano /usr/share/perl5/INetSim/DNS.pm” (fare attenzione alle maiuscole e minuscole). Modifichiamo come da immagine e salviamo CTRL+O. Facciamo questa operazione anche per non avere problemi con futuri aggiornamenti, in quanto è stato dichiarato “deprecated method”.

## Test sulla configurazione di INetSim

Andiamo su Windows 7 e avviamo il browser. Nel nostro caso ho aggiornato manualmente Windows 7, per supportare un browser veloce e leggero, in sostituzione del browser di default Internet Explorer, Midori.

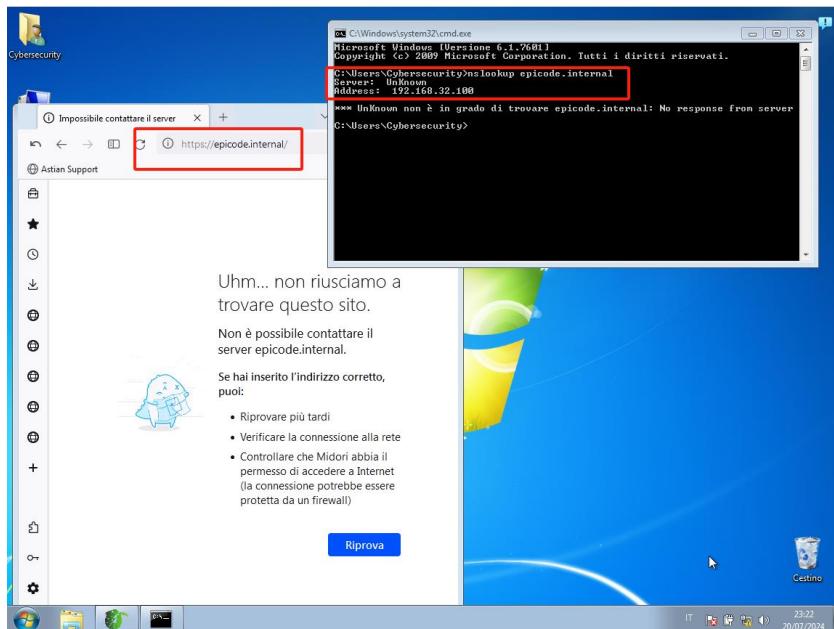
Inseriamo nell'url l'indirizzo ip 192.168.32.100



Qui concludiamo che funziona perfettamente, il collegamento e il caricamento della pagina fake HTML di Inetsim. Per cui a livello teorico il server HTTP funziona. Possiamo passare al prossimo test.

14

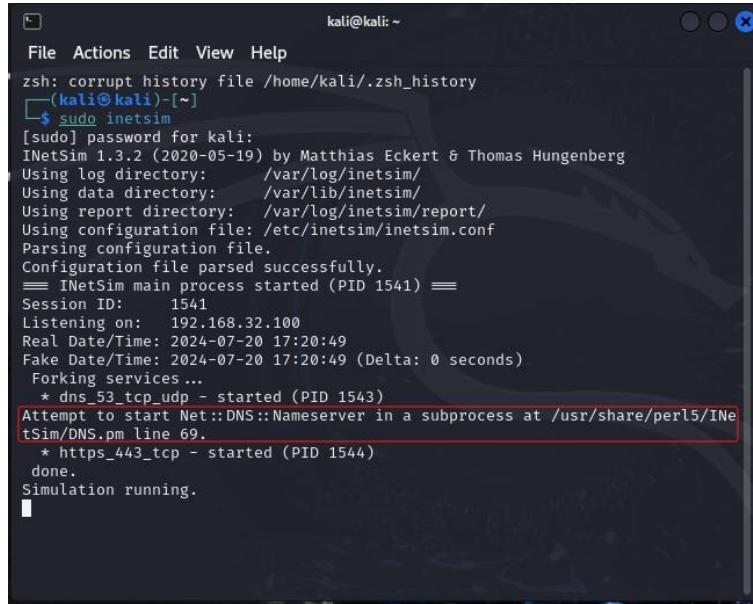
Inseriamo nell'url l'indirizzo il dominio **epicode.internal**



Purtroppo notiamo che la pagina non carica. Interrogiamo il server DNS con il comando sul terminale “nslookup epicode.internal” e scopriamo che il DNS è configurato correttamente, vedi riferimento sezione “Configurazione DNS di Windows 7”, tuttavia il server è “Unknown” per cui capiamo che non carica server, c’è un probabile errore di collegamento/ configurazione DNS lato Kali Linux/ Inetsim.

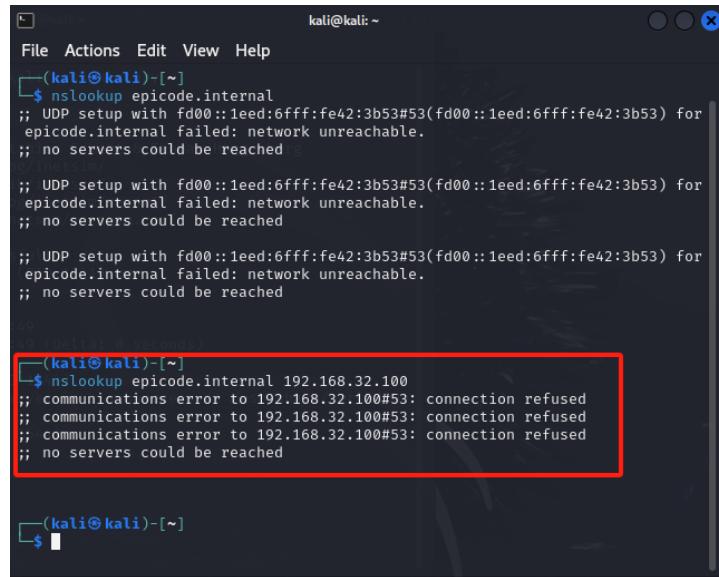
## Analisi e risoluzione del mancato funzionamento del server DNS

Una volta accertatoci che il problema non dipende da Windows 7, torniamo a Kali Linux ed a Inetsim



```
kali@kali: ~
File Actions Edit View Help
zsh: corrupt history file /home/kali/.zsh_history
└─(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 1541) ==
Session ID:      1541
Listening on:    192.168.32.100
Real Date/Time: 2024-07-20 17:20:49
Fake Date/Time: 2024-07-20 17:20:49 (Delta: 0 seconds)
Forking services...
* dns_53_tcp_udp - started (PID 1543)
Attempt to start Net::DNS::Nameserver in a subprocess at /usr/share/perl5/INe
tSim/DNS.pm line 69.
* https_443_tcp - started (PID 1544)
done.
Simulation running.
```

Una volta riavviato Inetsim, notiamo subito che c'è stato un tentativo di aprire la porta 53 relativo al server DNS. Per capire meglio utilizziamo lo stesso comando su un nuovo terminale di Kali Linux "nslookup episode.internal 192.168.32.100"



```
kali@kali: ~
File Actions Edit View Help
└─(kali㉿kali)-[~]
$ nslookup episode.internal
;; UDP setup with fd00::1eed:6fff:fe42:3b53#53(fd00::1eed:6fff:fe42:3b53) for
episode.internal failed: network unreachable.
;; no servers could be reached

;; UDP setup with fd00::1eed:6fff:fe42:3b53#53(fd00::1eed:6fff:fe42:3b53) for
episode.internal failed: network unreachable.
;; no servers could be reached

;; UDP setup with fd00::1eed:6fff:fe42:3b53#53(fd00::1eed:6fff:fe42:3b53) for
episode.internal failed: network unreachable.
;; no servers could be reached

;; (Delta: 0 seconds)
└─(kali㉿kali)-[~]
$ nslookup episode.internal 192.168.32.100
;; communications error to 192.168.32.100#53: connection refused
;; communications error to 192.168.32.100#53: connection refused
;; communications error to 192.168.32.100#53: connection refused
;; no servers could be reached

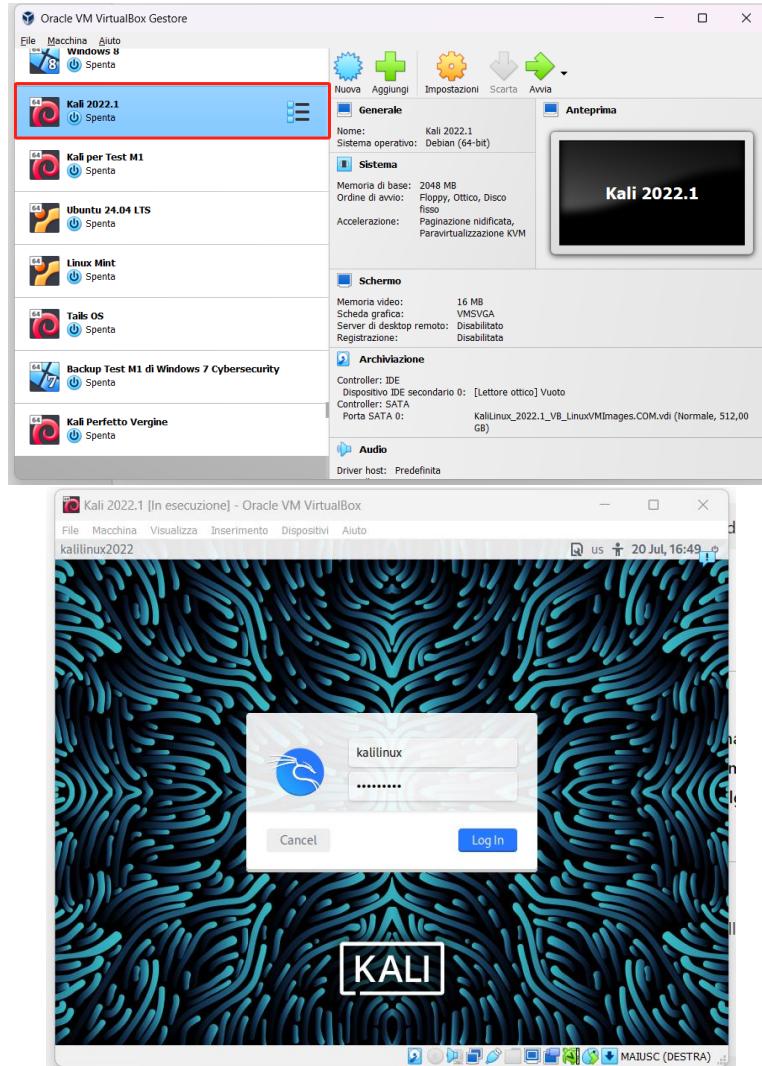
└─(kali㉿kali)-[~]
$
```

15

Purtroppo da questo test, abbiamo la conferma che la porta 53, necessaria per far funzionare il server DNS, non è attiva ed è questo il motivo per la quale non ha funzionato il test con il dominio episode.internal. Purtroppo nemmeno facendo gli ultimi aggiornamenti risolve il problema.

Ricontrollando tutte le configurazioni, facendo varie ricerche, consultazioni e ragionamenti concludo che è un problema di compatibilità. Kali Linux è arrivato al momento alla versione 2024.1 mentre Inetsim la versione corrente 1.3.2, che è stata rilasciata il 19 maggio 2020 e sono ormai 4 anni che non è stato aggiornato (fonti <https://www.inetsim.org/>). A conferma del mio sospetto, ho trovato prove che INetSim funzionava correttamente con una configurazione simile alla mia, in cui il server DNS operava perfettamente e il dominio veniva interpretato in modo corretto. Tuttavia, i test che ho consultato risalgono a sei anni fa.

Si prosegue quindi con l'installazione e configurazione identica al paragrafo "Configurazione di Kali Linux" di una nuova macchina virtuale Kali Linux ma versione 2022.1. L'unica differenza è che le credenziali di accesso sono "kalilinux" sia per nome utente che password.



16

Una volta avviato inetsim possiamo notare che la porta 53 è correttamente avviata.

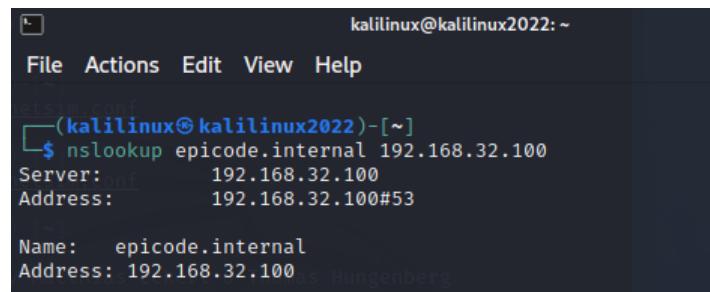
```

kalilinux@kalilinux2022: ~
File Actions View Help
-(kalilinux@kalilinux2022)-[~]
$ sudo 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
Parsin
Configuration file parsed successfully.
== INetSim main process started (PID 1237) ==
Session ID: 1237
Listening on: 192.168.32.100
Real Date/Time: 2024-07-20 16:50:43
Fake Date/Time: 2024-07-20 16:50:43 (Delta: 0 seconds)
Forking services ...
* dns_53_tcp_udp - started (PID 1239)
* https_443_tcp - started (PID 1240)
done.
Simulation running.

```

## Test sulla risoluzione DNS del dominio epicode.internal

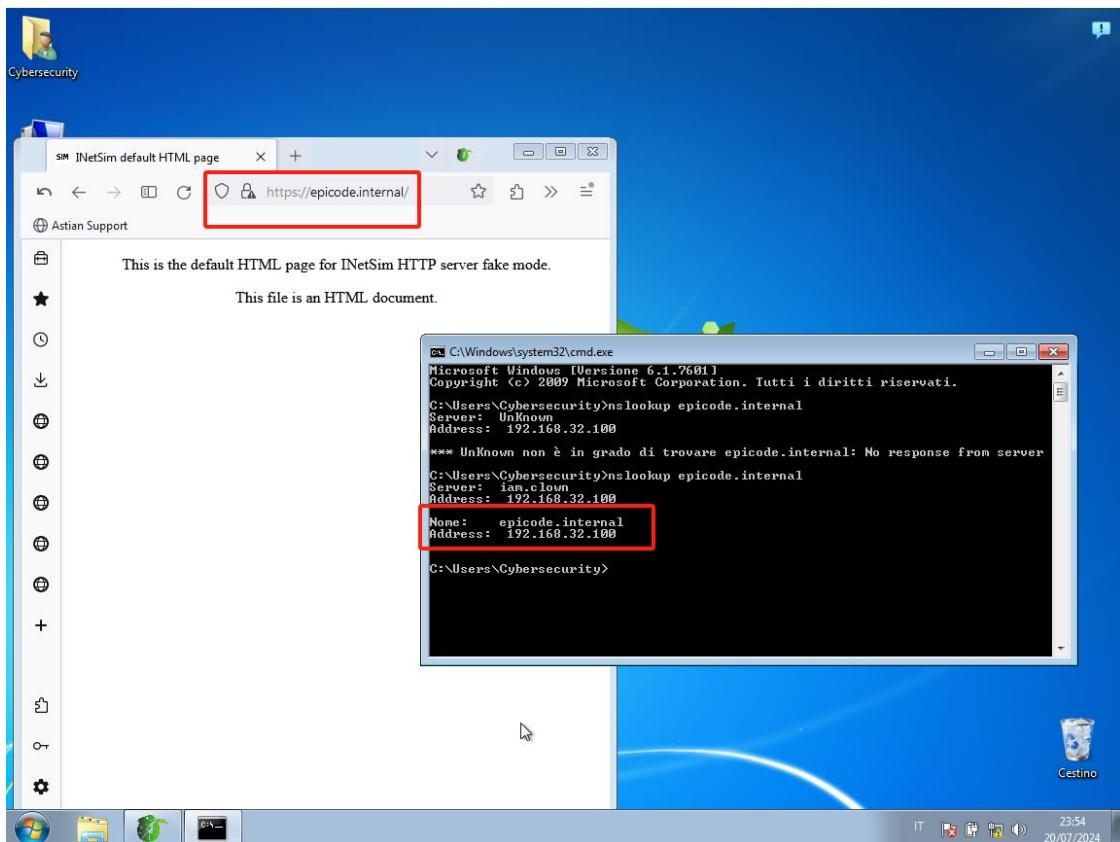
Testiamo con il comando “nslookup epicode.internal 192.168.32.100” su un altro terminale di Kali Linux la risoluzione del dominio. Tutto corretto.



```
kalilinux@kalilinux2022:~  
File Actions Edit View Help  
netSim.conf  
(kalilinux@kalilinux2022)-[~]  
$ nslookup epicode.internal 192.168.32.100  
Server: 192.168.32.100  
Address: 192.168.32.100#53  
  
Name: epicode.internal  
Address: 192.168.32.100
```

Test eseguito con successo.

Ripetiamo il test sul browser di Windows 7 con il link <https://epicode.internal/> e lo stesso comando sul terminale.

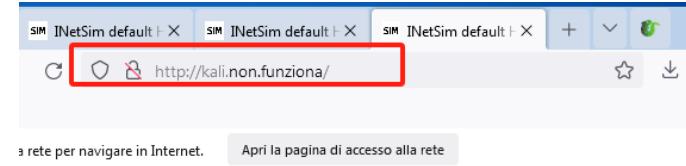
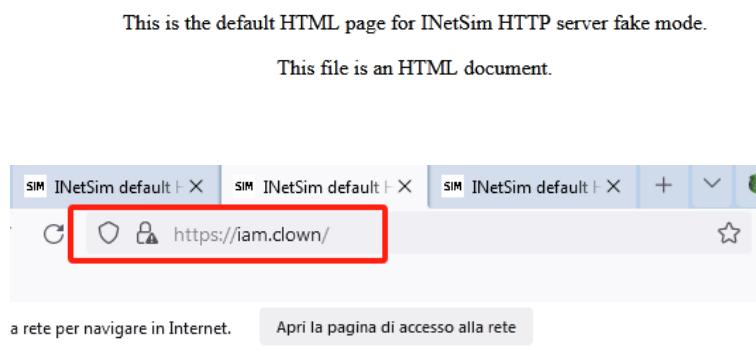
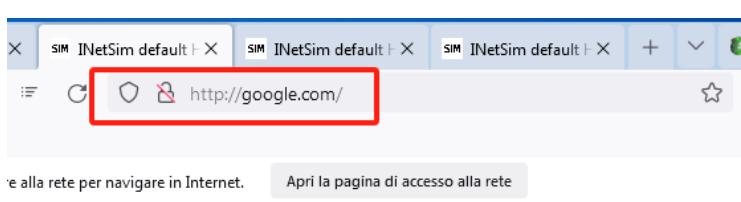
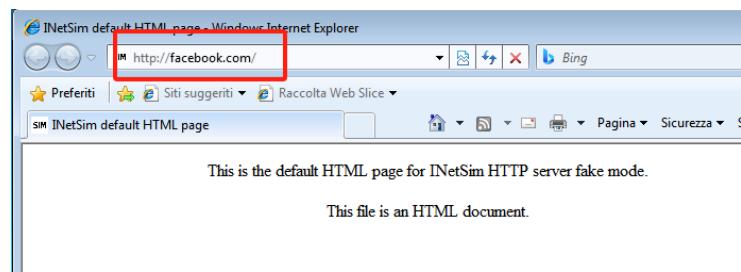


Test eseguito con successo.

Ulteriori verifiche sulla risoluzione dei domini personalizzati

Si sono inseriti ulteriori domini che risolvono in 192.168.32.100 per ulteriori verifiche.

```
#dns_static ftp.buji.net 192.168.32.100
dns_static episode.internal 192.168.32.100
dns_static facebook.com 192.168.32.100
dns_static google.com 192.168.32.100
dns_static iam.clown 192.168.32.100
dns_static kali.non.funziona 192.168.32.100
```



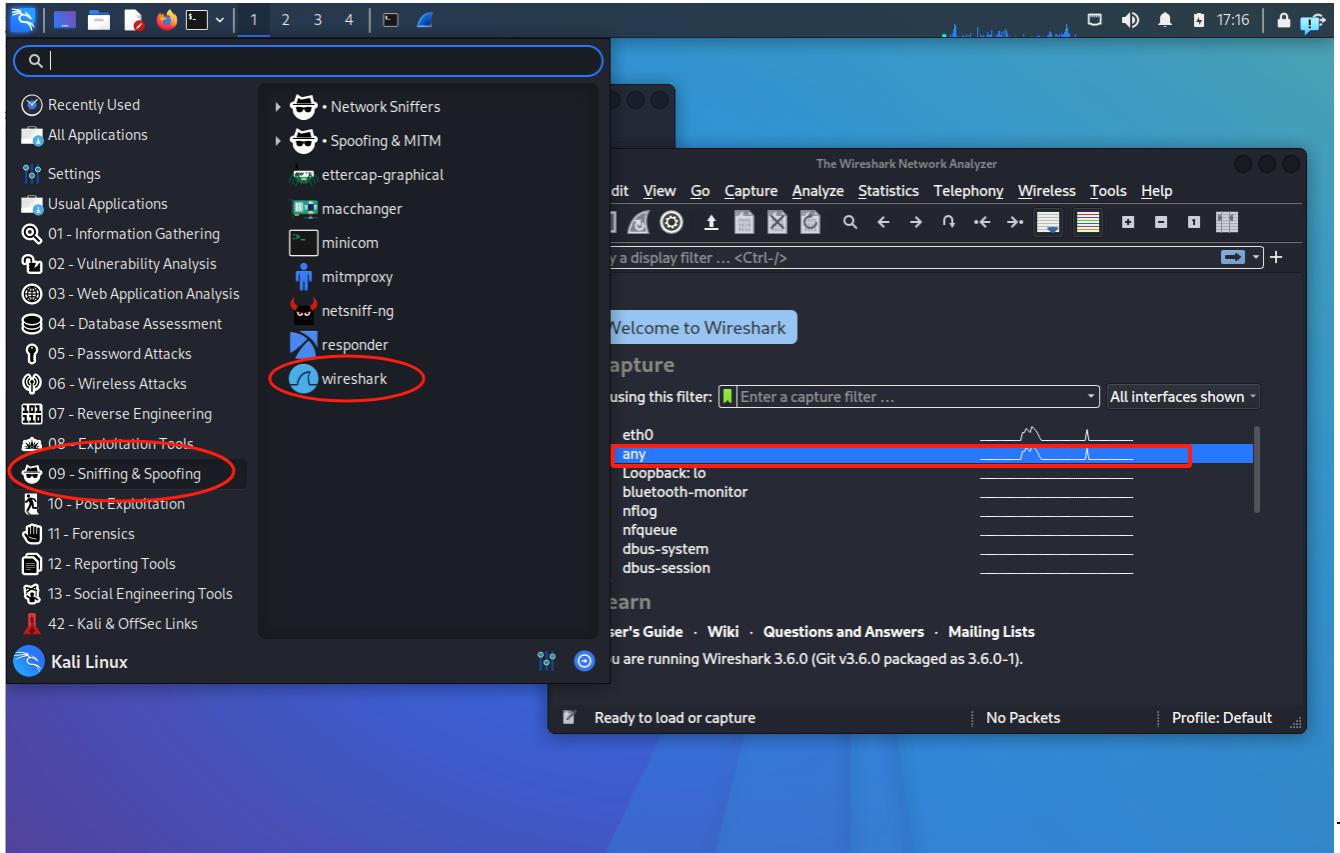
18

Tutti i test hanno avuto esito positivo, infatti hanno risolto i domini personalizzati con successo reindirizzandoli nella pagina fake di inetsim.

# Wireshark

## Analisi con protocollo HTTPS

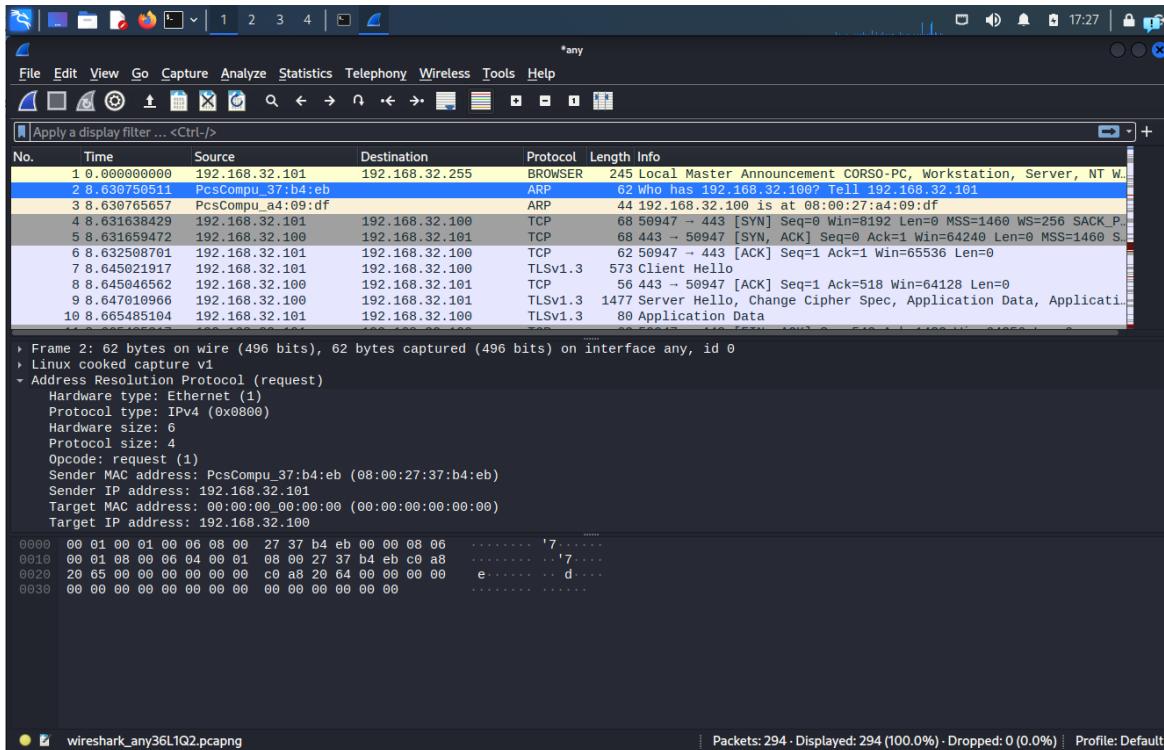
Tenendo attivo inetsim, apriamo Wireshark su Kali Linux e selezioniamo "Any".



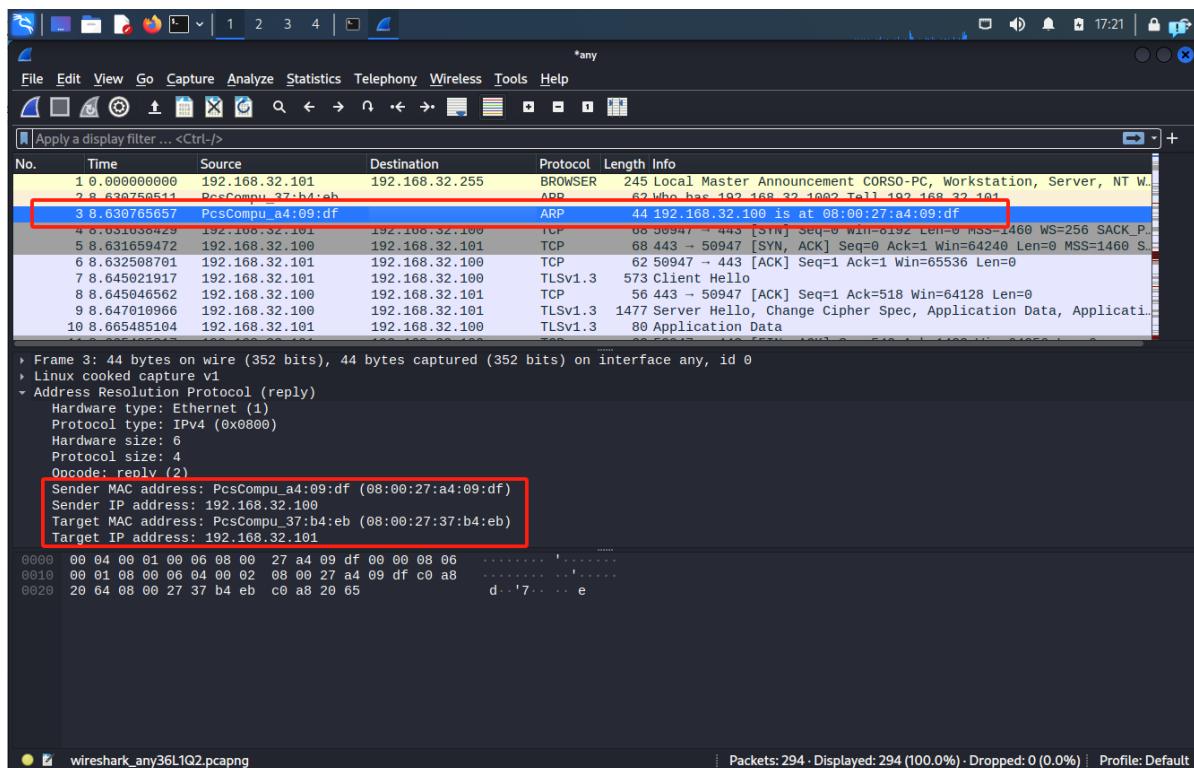
Premiamo la pinna blu e su Windows 7 ricarichiamo la pagina web <https://epicode.internal/> e fermiamo con il quadrato rosso.

## MAC Address

Il primo pacchetto è in protocollo ARP perché il mittente Windows 7 192.168.32.101 inizia con una richiesta broadcast alla rete, domandando a chi appartenga l'indirizzo IP 192.168.32.100 (risoluzione del dominio dal server DNS).

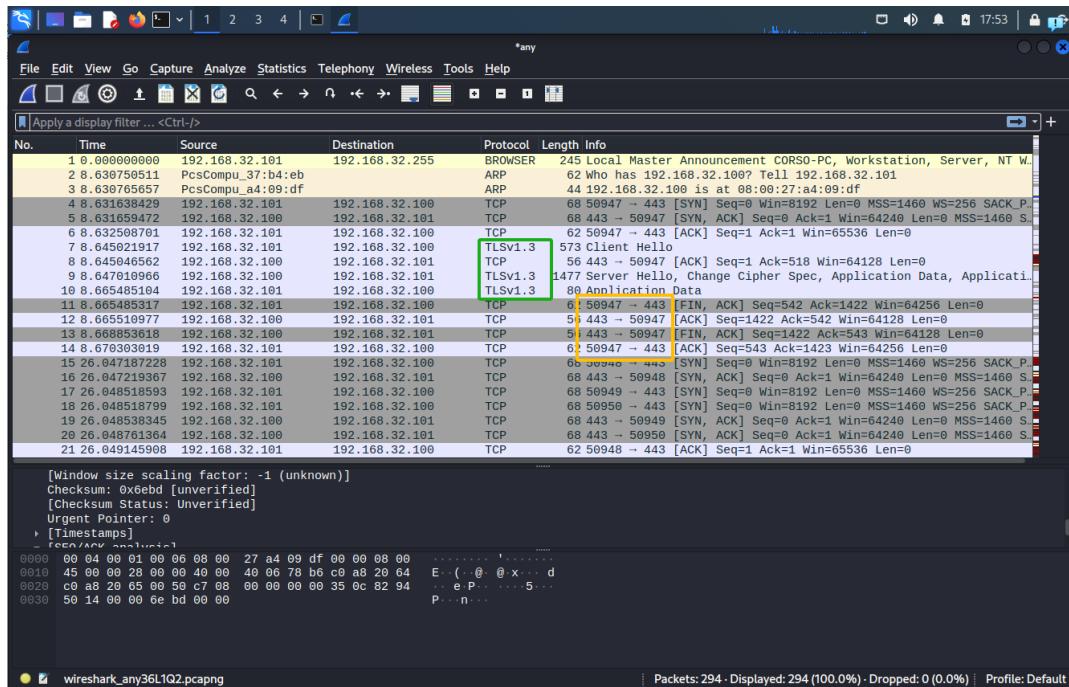


In risposta il server HTTPS 192.168.32.100 risponde inoltrando il proprio indirizzo MAC. A partire da questo punto avviene l'associazione IP | MAC ADDRESS che verrà salvata in un'apposita lista per agevolare la comunicazione senza dover ripetere la richiesta ARP. Appunto da questo pacchetto di ritorno possiamo, come evidenziato, trovare gli indirizzi MAC di Windows 7 e Kali Linux



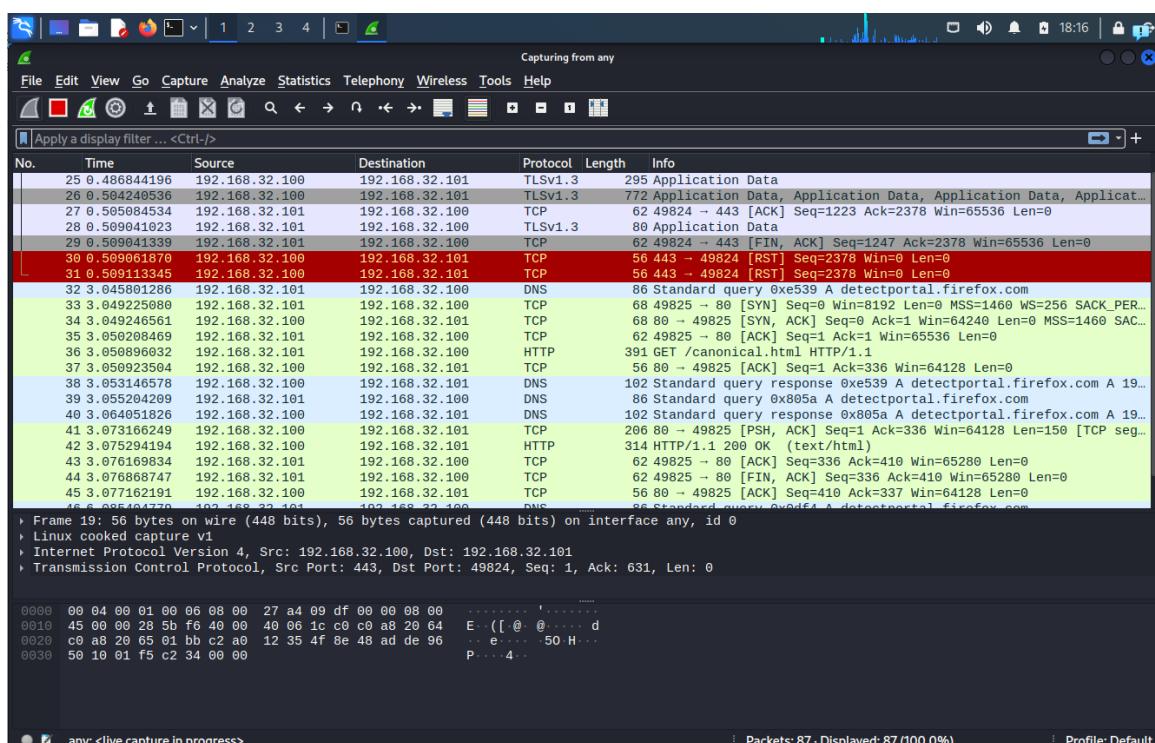
## Analisi contenuto richiesta HTTPS

Una volta trovati mittente e destinatario possiamo notare che il collegamento attraverso il protocollo TCP è avvenuto con successo.



Troviamo il protocollo di crittografia TLS (rettangolo verde) e la porta 443 (rettangolo giallo) convenzionalmente impostata per HTTPS, appunto ci conferma che il protocollo utilizzato è in HTTPS.

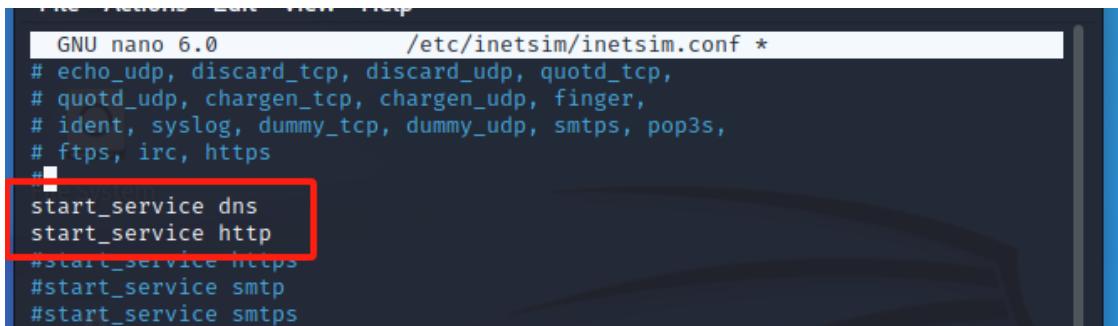
Per quanto riguarda il protocollo TCP possiamo trovare prove dell'utilizzo del principio del "three-way handshake" perché troviamo SIN, SIN-ACK e ACK. Questo processo garantisce che sia il client che il server siano sincronizzati e pronti a trasmettere dati in modo affidabile.



## Analisi con protocollo HTTP

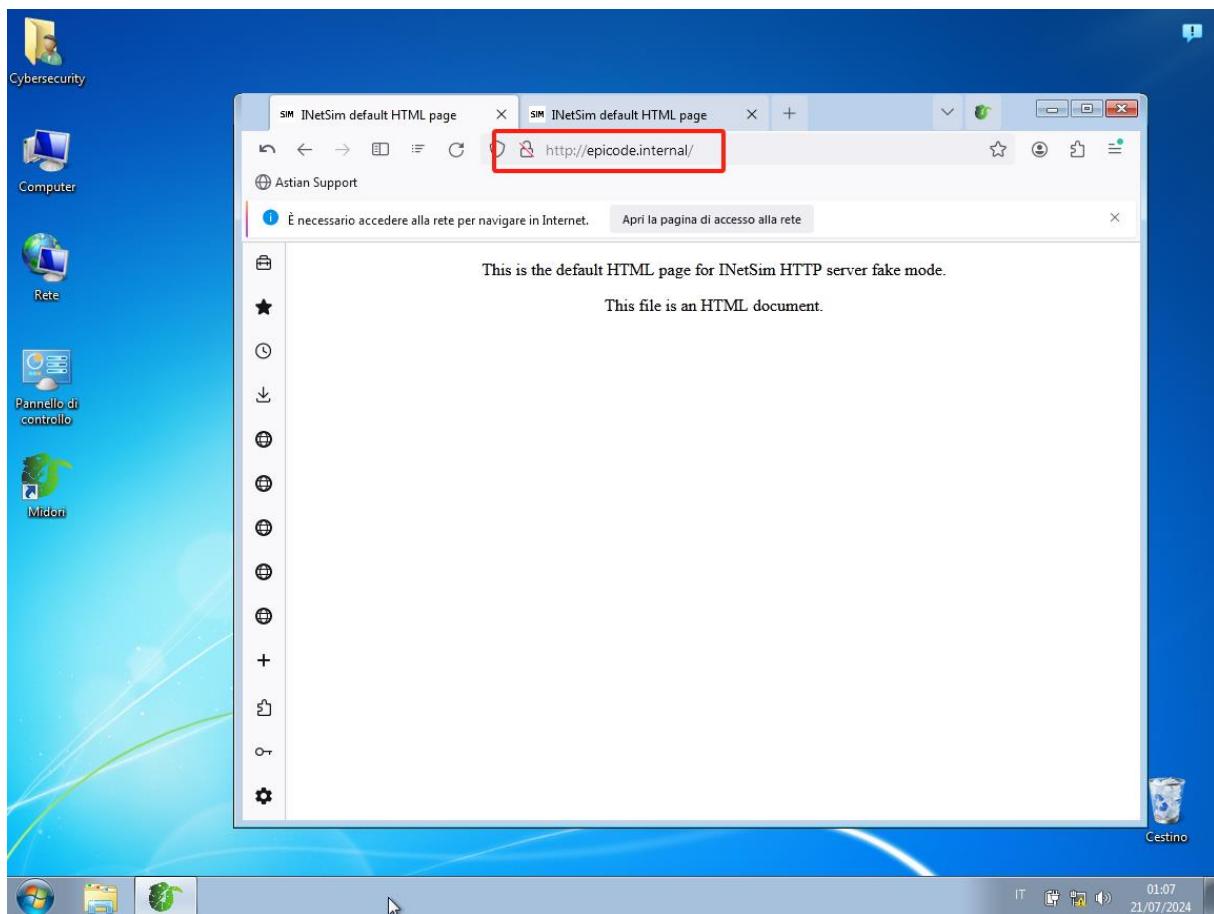
### Configurazione in HTTP

Per analizzare il tutto attraverso il protocollo HTTP. Riconfiguriamo Inetsim attivando il servizio http e, a discrezione, disattivare o meno il servizio https.



```
GNU nano 6.0          /etc/inetsim/inetsim.conf *
# 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
```

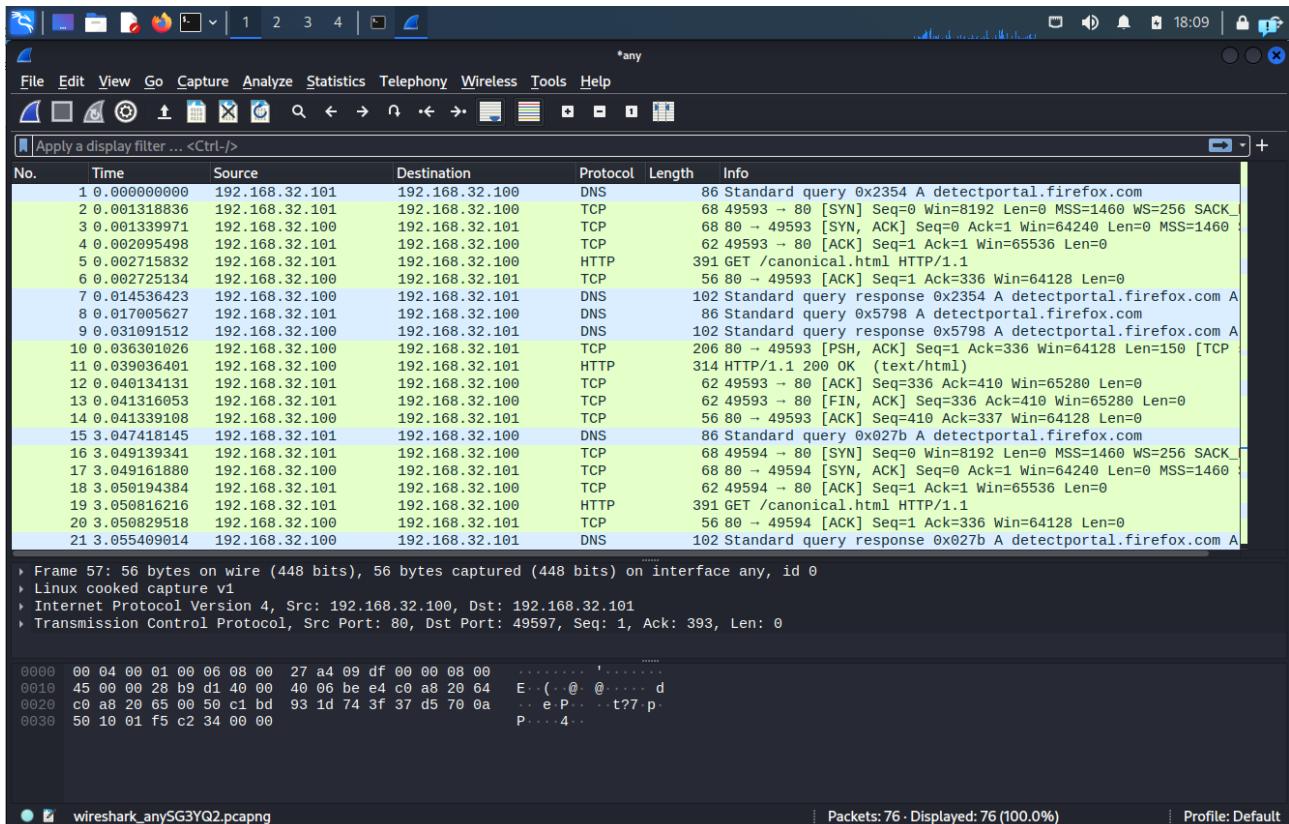
Ripetiamo quanto fatto precedente per l'analisi con Wireshark, aprendo tuttavia il link senza la "s" finale, quindi <http://epicode.internal/>



22

## Analisi contenuto richiesta HTTP

Come precedentemente effettuato ricarichiamo la pagina con Wireshark attivo.



Le principali differenze:

- l'assenza del protocollo di crittografia TLS perché il protocollo HTTP è privo di crittografia;
- non c'è nessuna richiesta ARP in quanto, non avendo spento le macchine, l'associazione indirizzi IP MAC è ancora salvata nella lista;
- la porta 80, convenzionalmente utilizzato per il protocollo HTTP e assenza della porta 443 riservata al protocollo HTTPS.

Si conferma l'utilizzo del protocollo TCP e dell'utilizzo del principio del "three-way handshake" perché ritroviamo SIN, SIN-ACK e ACK.

Per il resto non ci sono differenze significative.