

EPICODE

TEST FINE MODULO 1

CONSEGNA:

Nell'esercizio di oggi metteremo insieme le competenze acquisite finora. Lo studente verrà valutato sulla base della risoluzione al problema seguente.

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) richiede tramite web browser una risorsa all'hostname **epicode.internal** che risponde 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 HTTPS.

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.

(PAG. TOT. 12)

Apriamo su VirtualBox le macchine virtuali di Kali Linux e Windows 7

Iniziamo modificando gli indirizzi IP

Per svolgere questo procedimento su Kali:

- apriamo il terminale
- digitiamo e lanciamo il comando *sudo nano /etc/network/interfaces*
- modifichiamo l'indirizzo IP (192.168.32.100)

```
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
```

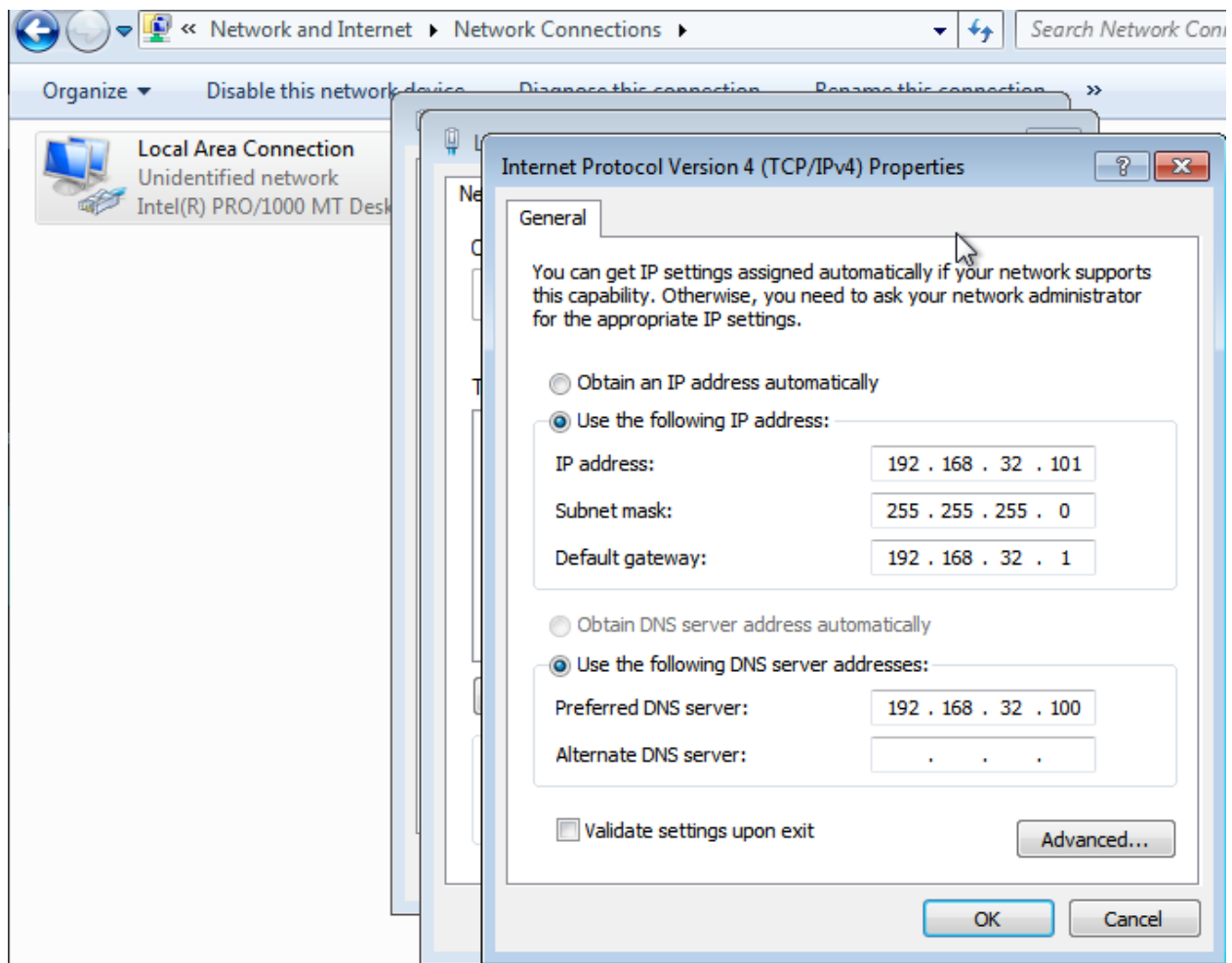
- per far sì che le impostazioni cambiate siano valide dobbiamo riavviare la macchina
- all'avvio lanciamo il comando *ifconfig*

```
L$ 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::a00:27ff:feeb:7ef5 prefixlen 64 scopeid 0x20<link>
        ether 08:00:27:cb:7e:f5 txqueuelen 1000 (Ethernet)
        RX packets 0 bytes 0 (0.0 B)
        RX errors 0 dropped 0 overruns 0 frame 0
        TX packets 15 bytes 2344 (2.2 KiB)
        TX errors 0 dropped 0 overruns 0 carrier 0 collision
    ns 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 collision
    ns 0
```

Configurazione Windows 7:

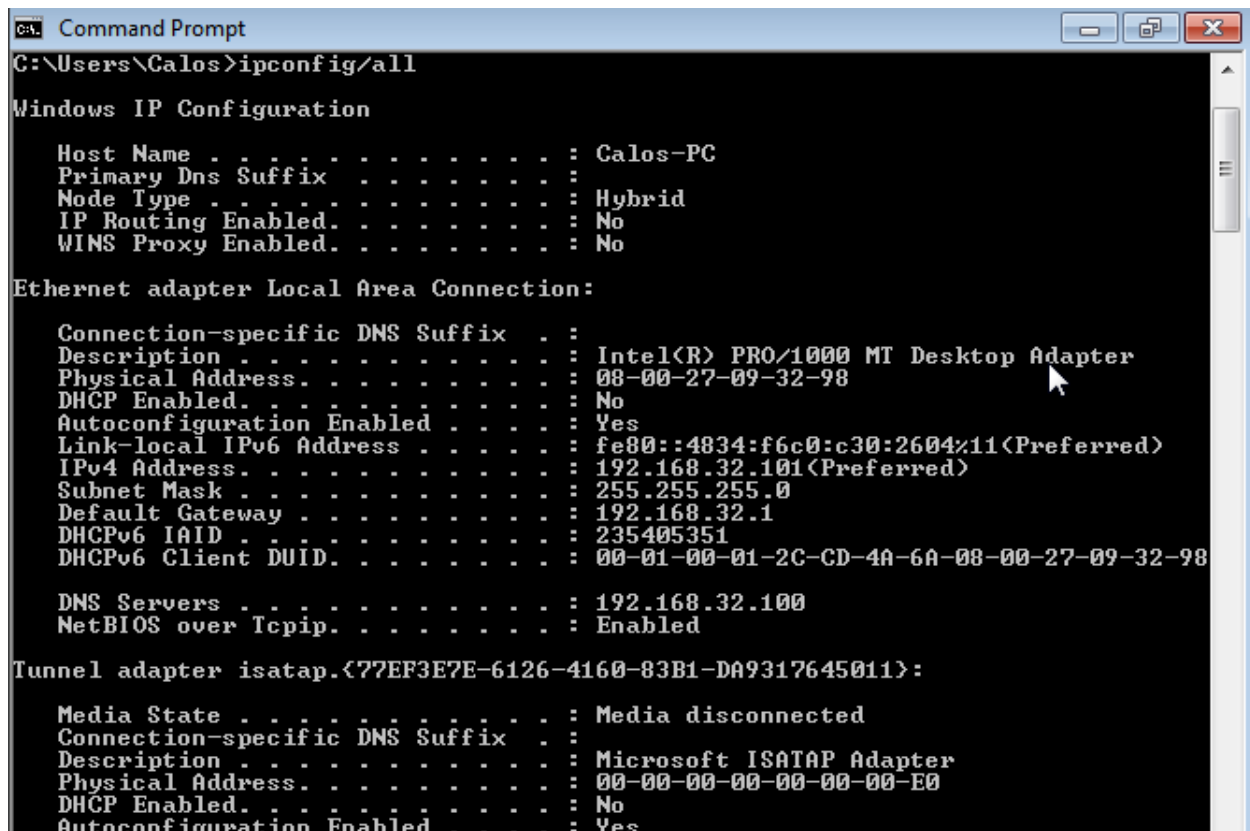
- apriamo il pannello di controllo
- andiamo su Rete e Internet
- centro di rete e condivisione
- modifica impostazioni scheda di rete
- doppio click sulla nostra rete
- proprietà
- protocollo internet versione 4 (TCP/IPv4) ed impostiamo l'indirizzo *IP* 192.168.32.101



Mentre siamo in questa schermata modifichiamo anche il *Defaul gateway* (192.168.32.1) e *Preferred DNS server* (192.168.32.100) che ci serviranno per lanciare tramite web browser la richiesta epicode.internal;

NB: il DNS inserito è l'IP di Kali

successivamente apriamo il Prompt dei comandi e lanciamo il comando *ipconfig/all*,

A screenshot of a Windows Command Prompt window titled "Command Prompt". The command "ipconfig/all" has been entered and executed. The output shows the Windows IP Configuration for a system named "Calos-PC". It lists the primary DNS suffix as empty, node type as "Hybrid", and IP routing and WINS proxy as disabled. Under "Ethernet adapter Local Area Connection:", it shows the connection-specific DNS suffix is empty, description is "Intel(R) PRO/1000 MT Desktop Adapter", physical address is "08-00-27-09-32-98", DHCP is enabled, autoconfiguration is enabled, link-local IPv6 address is "fe80::4834:f6c0:c30:2604%11 (Preferred)", IPv4 address is "192.168.32.101 (Preferred)", subnet mask is "255.255.255.0", default gateway is "192.168.32.1", DHCPv6 IAID is "235405351", and DHCPv6 client DUID is "00-01-00-01-2C-CD-4A-6A-08-00-27-09-32-98". It also lists DNS servers as "192.168.32.100" and NetBIOS over Tcpip as enabled. Under "Tunnel adapter isatap.{77EF3E7E-6126-4160-83B1-DA9317645011}:", it shows the media state as disconnected, description as "Microsoft ISATAP Adapter", physical address as "00-00-00-00-00-00-E0", DHCP is disabled, and autoconfiguration is enabled.

```
Command Prompt
C:\Users\Calos>ipconfig/all

Windows IP Configuration

Host Name . . . . . : Calos-PC
Primary Dns Suffix . . . . . :
Node Type . . . . . : Hybrid
IP Routing Enabled. . . . . : No
WINS Proxy Enabled. . . . . : No

Ethernet adapter Local Area Connection:

Connection-specific DNS Suffix . :
Description . . . . . : Intel(R) PRO/1000 MT Desktop Adapter
Physical Address. . . . . : 08-00-27-09-32-98
DHCP Enabled. . . . . : No
Autoconfiguration Enabled . . . . : Yes
Link-local IPv6 Address . . . . . : fe80::4834:f6c0:c30:2604%11(Preferred)
IPv4 Address. . . . . : 192.168.32.101(Preferred)
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.32.1
DHCPv6 IAID . . . . . : 235405351
DHCPv6 Client DUID. . . . . : 00-01-00-01-2C-CD-4A-6A-08-00-27-09-32-98

DNS Servers . . . . . : 192.168.32.100
NetBIOS over Tcpip. . . . . : Enabled

Tunnel adapter isatap.{77EF3E7E-6126-4160-83B1-DA9317645011}:

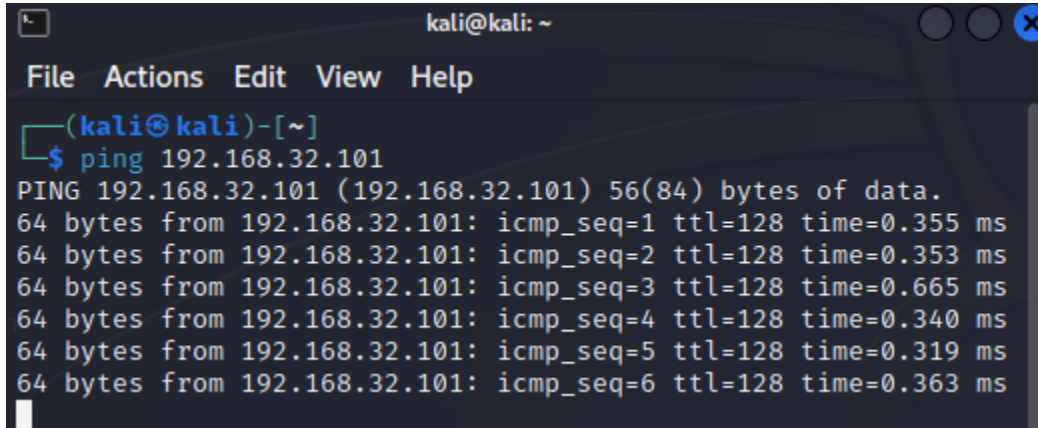
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :
Description . . . . . : Microsoft ISATAP Adapter
Physical Address. . . . . : 00-00-00-00-00-00-E0
DHCP Enabled. . . . . : No
Autoconfiguration Enabled . . . . : Yes
```

notiamo che le impostazioni che abbiamo applicato sono salvate correttamente.

Proviamo il ping tra Kali e Win7

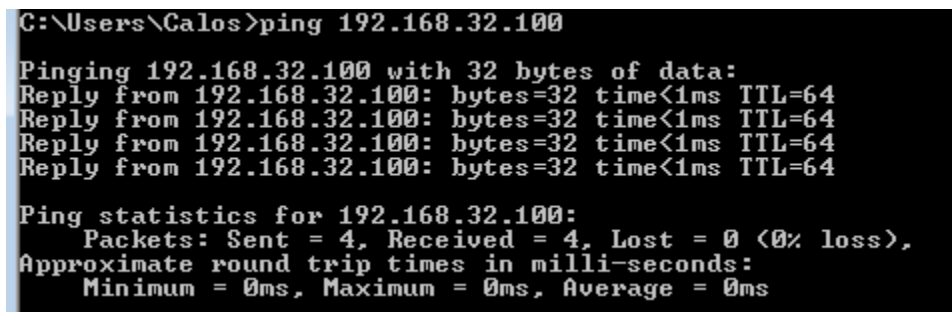
Per assicurarci che le due macchine comunicano tra loro:

da Kali, apriamo il terminale e digitiamo *ping 192.168.32.101*

A screenshot of a Kali Linux terminal window. The title bar shows 'kali@kali: ~'. The menu bar includes 'File', 'Actions', 'Edit', 'View', and 'Help'. The prompt is '(kali@kali)-[~]'. The user has entered '\$ ping 192.168.32.101'. The output shows a successful ping to 192.168.32.101 with 56(84) bytes of data. Six replies are shown, each with a sequence number from 1 to 6, a TTL of 128, and a time ranging from 0.319 ms to 0.665 ms.

```
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=0.355 ms  
64 bytes from 192.168.32.101: icmp_seq=2 ttl=128 time=0.353 ms  
64 bytes from 192.168.32.101: icmp_seq=3 ttl=128 time=0.665 ms  
64 bytes from 192.168.32.101: icmp_seq=4 ttl=128 time=0.340 ms  
64 bytes from 192.168.32.101: icmp_seq=5 ttl=128 time=0.319 ms  
64 bytes from 192.168.32.101: icmp_seq=6 ttl=128 time=0.363 ms
```

da Windows 7, apriamo il prompt e digitiamo *192.168.32.100*

A screenshot of a Windows 7 command prompt window. The title bar shows 'C:\Users\Calos>'. The user has entered 'ping 192.168.32.100'. The output shows a successful ping to 192.168.32.100 with 32 bytes of data. Four replies are shown, each with a time less than 1ms and a TTL of 64. Ping statistics are also displayed, showing 4 packets sent, 4 received, and 0% loss.

```
C:\Users\Calos>ping 192.168.32.100  
  
Pinging 192.168.32.100 with 32 bytes of data:  
Reply from 192.168.32.100: bytes=32 time<1ms TTL=64  
Reply from 192.168.32.100: bytes=32 time<1ms TTL=64  
Reply from 192.168.32.100: bytes=32 time<1ms TTL=64  
Reply from 192.168.32.100: bytes=32 time<1ms TTL=64  
  
Ping statistics for 192.168.32.100:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
    Approximate round trip times in milli-seconds:  
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```


Così abbiamo conferma che c'è comunicazione tra le due macchine.

Inetsim e Configurazione DNS, HTTP e HTTPS

Inetsim è un software gratuito e open source che consente di simulare servizi Internet comuni in un ambiente di laboratorio

Procediamo con la configurazione:

- Apriamo il terminale di Kali
- `cd /etc/inetsim` → `Ls` → `sudo nano inetsim.conf`



```
(kali㉿kali)-[~]
$ cd /etc/inetsim

(kali㉿kali)-[/etc/inetsim]
$ ls
inetsim.conf

(kali㉿kali)-[/etc/inetsim]
$ sudo nano inetsim.conf
```

- lanciamo quindi il comando `sudo nano inetsim.conf` inseriamo la password di kali e si aprirà il file dove dobbiamo *commentare i servizi DNS, HTTP e HTTPS*, impostare *dns default ip (192.168.32.100)*, *bind address (0.0.0.0)* e *dns static (epicode.internal 192.168.32.100)*

```

GNU nano 7.2          inetsim.conf *
# 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

```

```

#####
# 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_static
#
# Static mappings for DNS
#
# Syntax: dns_static <fqdn hostname> <IP address>
#
# Default: none
#
#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

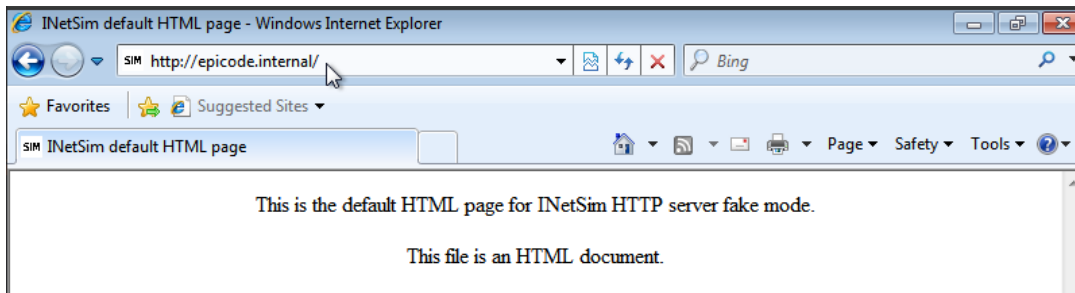
```

```
GNU nano 7.2      inetsim.conf *
#
# IP address to bind services to
#
# Syntax: service_bind_address <IP address>
#
# Default: 127.0.0.1
#
#
service_bind_address 0.0.0.0
```

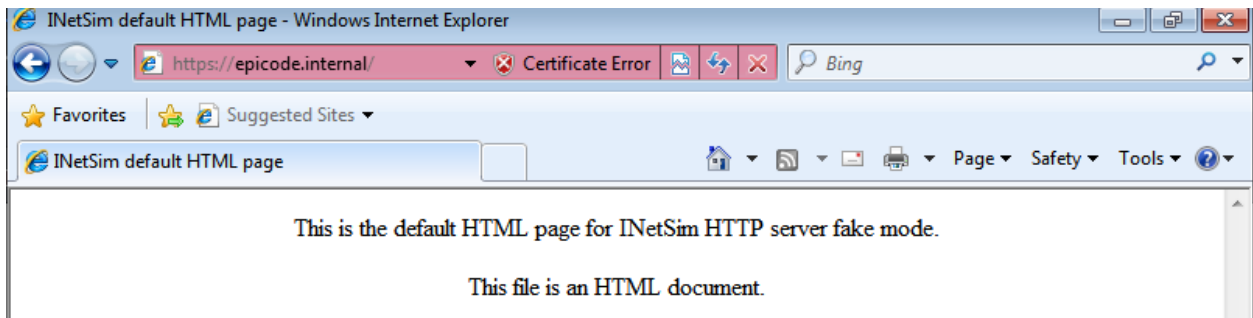
- riavviamo la macchina
- all'avvio da terminale lanciamo *sudo inetsim*

```
(kali㉿kali)-[/etc/inetsim]
$ 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
Parsing configuration file.
Configuration file parsed successfully.
== INetSim main process started (PID 16923) ==
Session ID:      16923
Listening on:    127.0.0.1
Real Date/Time:  2023-11-14 09:28:27
Fake Date/Time:  2023-11-14 09:28:27 (Delta: 0 seconds)
Forking services ...
  * https_443_tcp - started (PID 16926)
  * http_80_tcp  - started (PID 16925)
done.
Simulation running.
```

- manteniamo aperto kali con inetsim
- passiamo a windows e facciamo la richiesta di HTTP e HTTPS sul browser, quindi nella barra di ricerca inseriamo <http://epicode.internal>



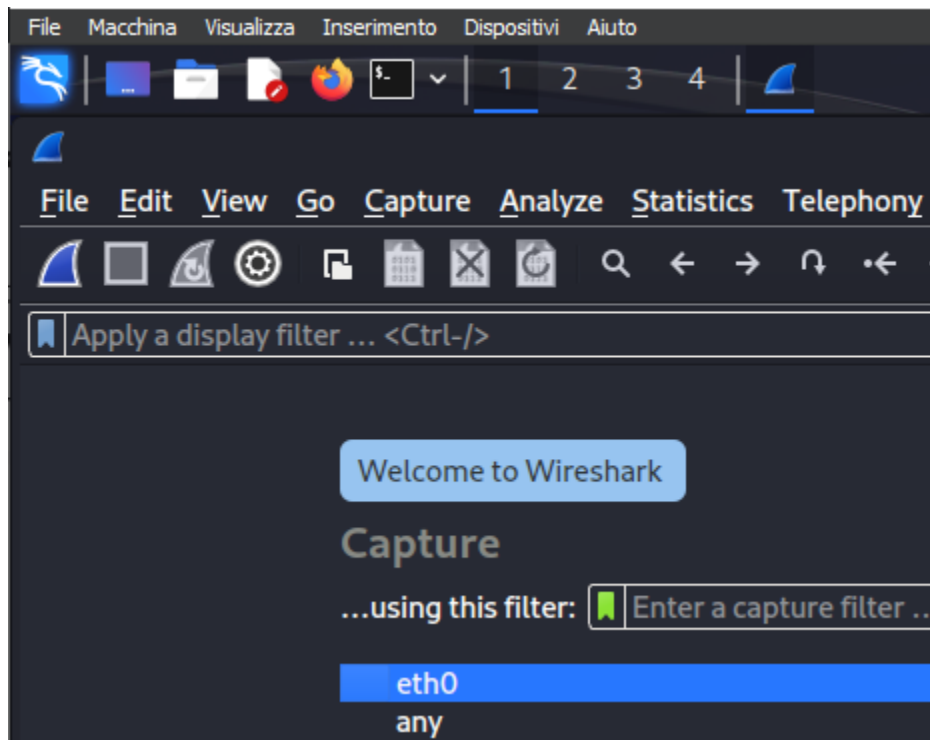
in una seconda pagina *https://epicode.internal*



Wireshark

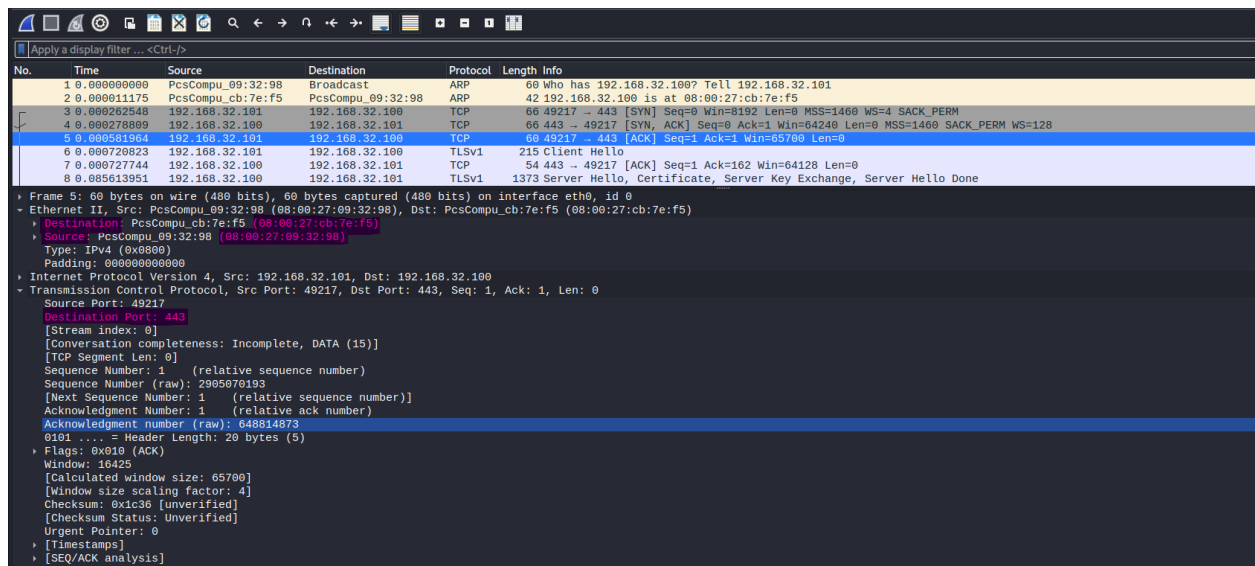
Wireshark permette di analizzare qualsiasi pacchetto, flusso di traffico o connessione che passa attraverso la scheda di rete del pc.

Per avviare Wireshark su Kali, apriamo il menù *09 - Sniffing & Spoofing*, e quindi avviamo Wireshark. Apriamo *eth0* che è la NIC che utilizzerà Kali per interagire con Windows 7



Intercettiamo HTTPS quindi, lanciando `sudo inetsim`, passiamo su Win7 e nella ricarichiamo la pagina HTTPS precedentemente aperta:

- Uno scambio di pacchetti *ARP* dove nel primo, Windows, chiede tramite messaggio *Broadcast* di chi siamo l'indirizzo IP; nel secondo, *Linux*, risponde fornendo il suo **MAC Address**
- Avviene la ***three-way-handshake***
- subito dopo il client invia un pacchetto *TLSv1 - Client Hello*, il server risponde con un altro pacchetto *TLSv1 - Server Hello* (***TLS è un protocollo fondamentale per la sicurezza di Internet. Garantendo l'autenticazione, l'integrità dei dati e la confidenzialità, TLS aiuta a proteggere gli utenti da attacchi informatici***)

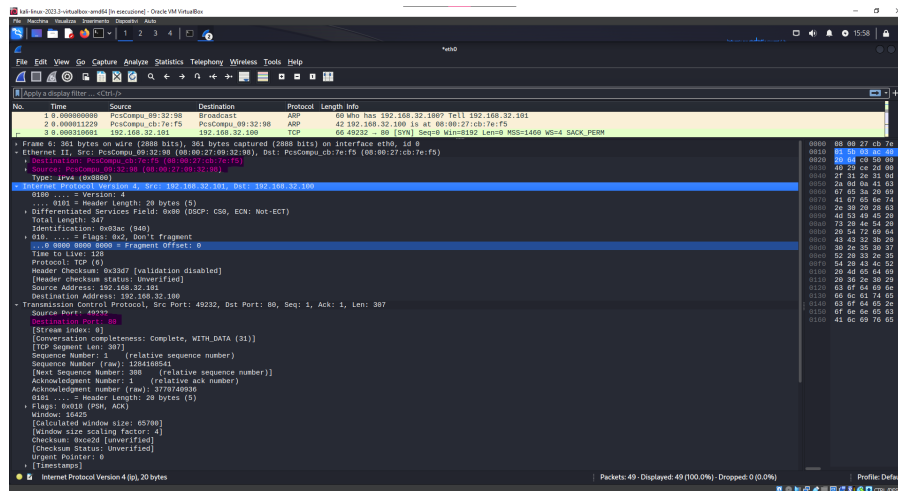


nell'immagine ho evidenziato gli indirizzi MAC di destinazione, quindi il MAC address di Kali (08:00:27:cb:7e:f5), e il MAC address della sorgente [Windows (08:00:27:09:32:98)];

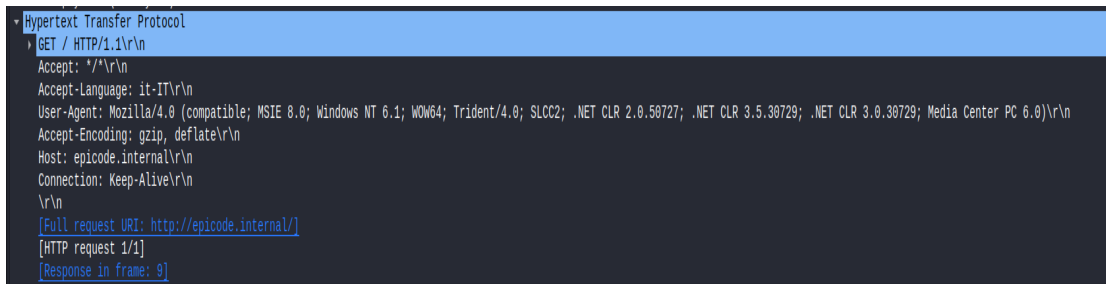
Notiamo che la porta di destinazione è **443**.

Ripetiamo l'operazione di sniffing con il server HTTP, quindi stavolta ricarichiamo la pagina di HTTP (ricordiamo che *Inetsim* deve essere lanciato sul terminale di Kali), così potremmo intercettare il traffico dati su Wireshark:

- Come per HTTPS avviene lo scambio di pacchetti **ARP**, quindi Windows chiede l'identità dell'IP 192.168.32.100 (IP di Kali) e Kali risponde fornendo il suo **MAC address**
- Anche nello sniffing di HTTP possiamo notare che rileva i MAC address di destinazione e di sorgente, e la porta **80**



- appare una “nuova voce” ***Hypertext Transfer Protocol*** dove all’interno possiamo notare la richiesta **GET**



Differenze tra HTTPS e HTTP

HTTPS dopo la **three-way-handshake** ha uno scambio di pacchetti **TLS**.

HTTP non consente una connessione sicura e infatti vengono mostrati nella richiesta GET tutte le informazioni, come HTML e il contenuto della pagina ricercata.