

### Requisiti e servizi:

- Kali Linux: IP 192.168.32.100
- Windows: 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) 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.

*In questo esercizio, devo simulare un'architettura client-server in un ambiente virtuale usando VirtualBox e poi analizzare il traffico di rete con Wireshark. Vi racconto passo-passo tutto quello che ho fatto, in ogni dettaglio.*

## 1. Preparazione delle macchine virtuali

### Scarico e installo VirtualBox

- Vado sul sito ufficiale di VirtualBox (<https://www.virtualbox.org>) e scarico l'ultima versione (7.1.6) per il mio sistema operativo Windows 11 Pro (architettura 64 bit). Dopo averlo installato, lo avvio.

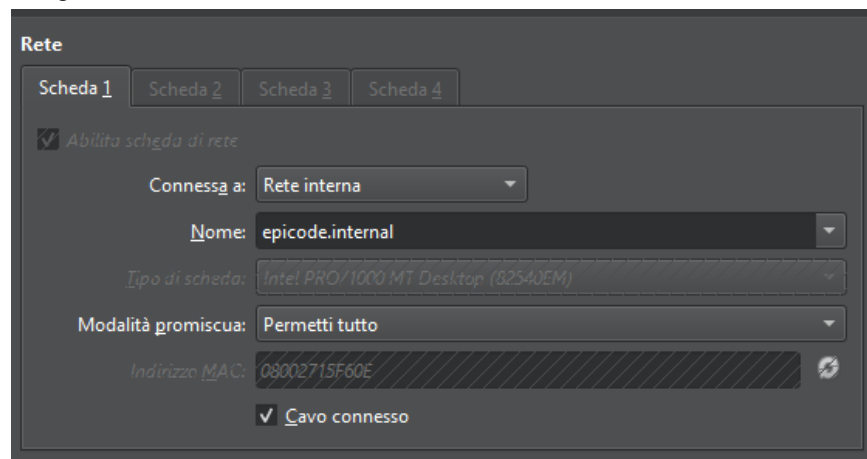
### Creo le macchine virtuali:

- Clicco su "Nuova" e creo la prima macchina virtuale, che chiamo "Host".
  - Scelgo il tipo di sistema operativo e la versione (Ubuntu 64-bit).
  - Assegno almeno 2 GB di RAM e 20 GB di spazio su disco
- Ripeto lo stesso processo per creare la seconda macchina virtuale, che chiamo "Guest".

### Vado a configurare le impostazioni di rete

- vado su "Impostazioni" > "Rete" per entrambe le macchine e scelgo "Scheda 1"
- imposto il "Tipo di connessione" su Rete interna (Internal Network).
- Assegno lo stesso nome alla rete interna per entrambe le macchine (epicode.internal).

- Questo mi permette di far comunicare le due macchine direttamente tra loro senza bisogno di una connessione esterna.



## 2. Installazione dei sistemi operativi

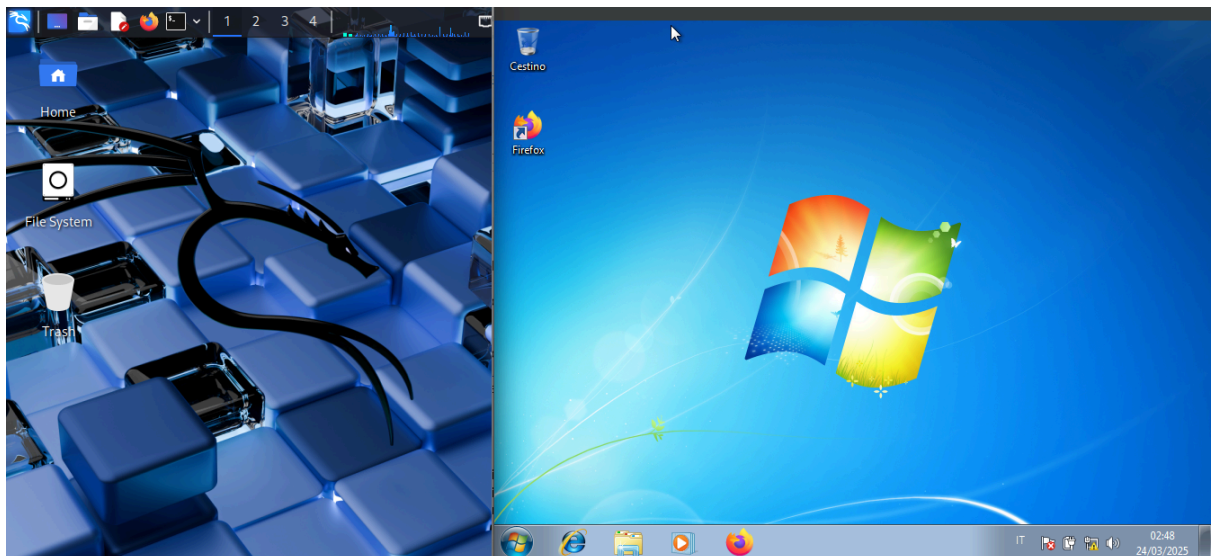
Scarico la Macchina virtuale preconstituita di Kali Linux

- Vado sul sito <https://www.kali.org/get-kali/#kali-virtual-machines> dove trovo la macchina virtuale già installata e configurata

Scarico l' Iso di Windows 7

- Seguo la Guida del percorso all'indirizzo <https://archive.org/details/win-7-aio-32x-64x>, avvio la macchina montando l'immagine ISO scaricata e seguo la procedura di installazione di Windows 7 Pro

In meno di 2 ore, ho pronto il mio Laboratorio



### 3. Configurazione degli Ambienti Virtuali

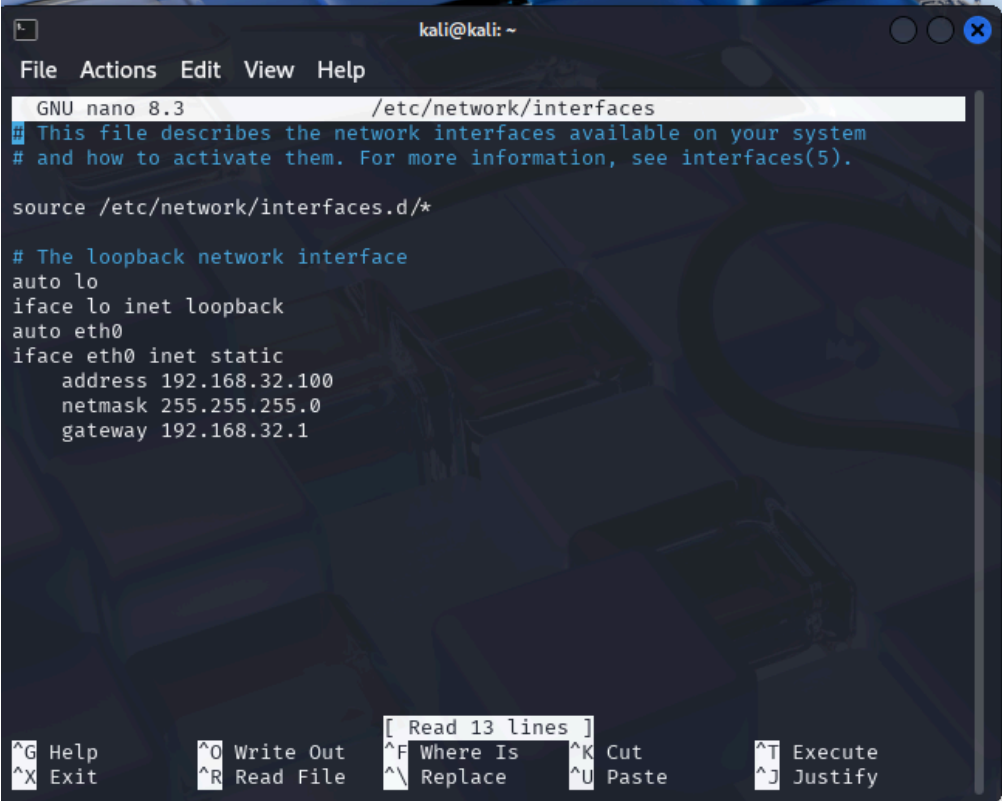
- Configurazione di Kali Linux

Assegnazione IP statico (192.168.32.100 )

Modifico il File di Configurazione di rete con

*sudo nano /etc/network/interfaces* aggiungendo le righe

```
auto eth0
iface eth0 inet static
    address 192.168.32.100
    netmask 255.255.255.0
    gateway 192.168.32.1
```



Ricordandomi di riavviare il servizio con

*sudo /etc/init.d/networking restart*

Dopodiché verifico con *ip* a se si è assegnato correttamente

```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ sudo nano /etc/network/interfaces  
[sudo] password for kali:  
(kali@kali)-[~]  
$ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group def  
ault qlen 1000  
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00  
    inet 127.0.0.1/8 scope host lo  
        valid_lft forever preferred_lft forever  
    inet6 ::1/128 scope host noprefixroute  
        valid_lft forever preferred_lft forever  
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP g  
roup default qlen 1000  
    link/ether 08:00:27:04:42:0f brd ff:ff:ff:ff:ff:ff  
    inet 192.168.32.100/24 brd 192.168.32.255 scope global eth0  
        valid_lft forever preferred_lft forever  
    inet6 fe80::a00:27ff:fe04:420f/64 scope link proto kernel_ll  
        valid_lft forever preferred_lft forever  
(kali@kali)-[~]  
$
```

## Installazione Apache e Configurazione del Server Https

Con `sudo apt install apache2`, procedo con l'installazione di Apache,  
con `sudo a2enmod ssl` ho abilitato il modulo SSL  
e con `sudo systemctl restart apache2` riavvio il servizio

Dopodiché creo un certificato ssl autofirmato con  
`sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout  
/etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt`

Configuro Apache per utilizzare il certificato per utilizzare il modulo ssl,  
aggiungendo al file di configurazione **default-ssl.conf** le righe

```
<VirtualHost *:443>  
    ServerAdmin webmaster@localhost  
    ServerName epicode.internal  
    DocumentRoot /var/www/html  
    SSLEngine on  
    SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt  
    SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key  
    <Directory /var/www/html>  
        Options Indexes FollowSymLinks  
        AllowOverride All  
        Require all granted  
    </Directory>  
</VirtualHost>
```

Infine eseguo lo script a2ensite : `sudo a2ensite default-ssl.conf` e concludo la l'installazione di Apache e del server https con un bel reload : `sudo systemctl reload apache2`

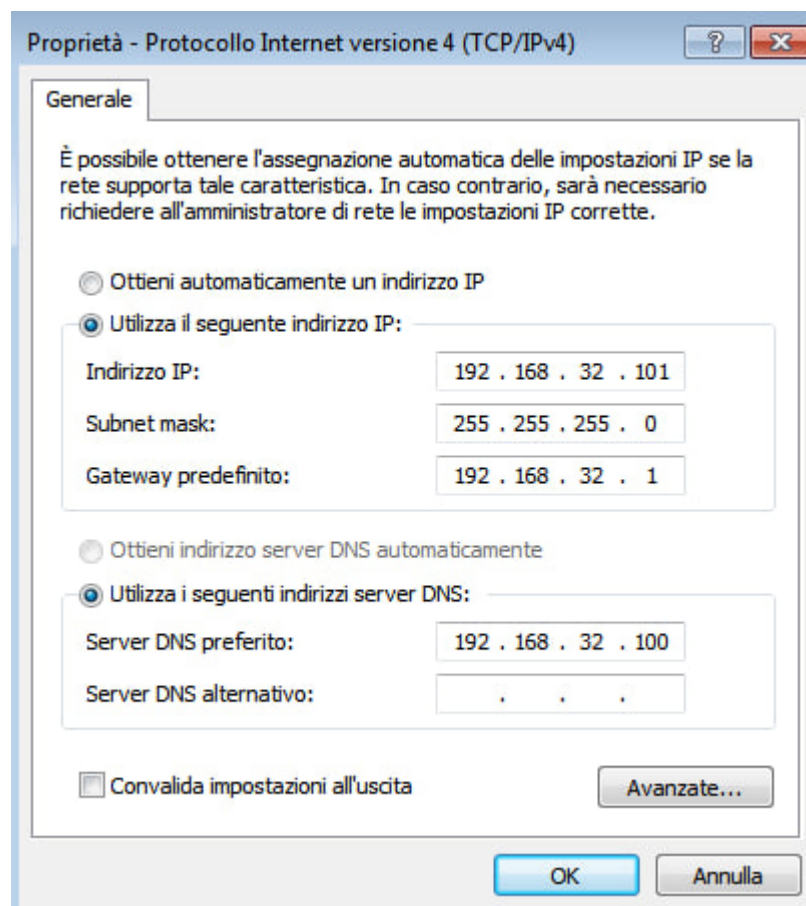
## Configurazione del DNS

Per concludere dal momento che inetsim non funziona e non posso accedere ad internet configuro il dns modificando il file hosts, per risolvere `epicode.internal` attraverso `sudo nano /etc/hosts` e inserendo la riga

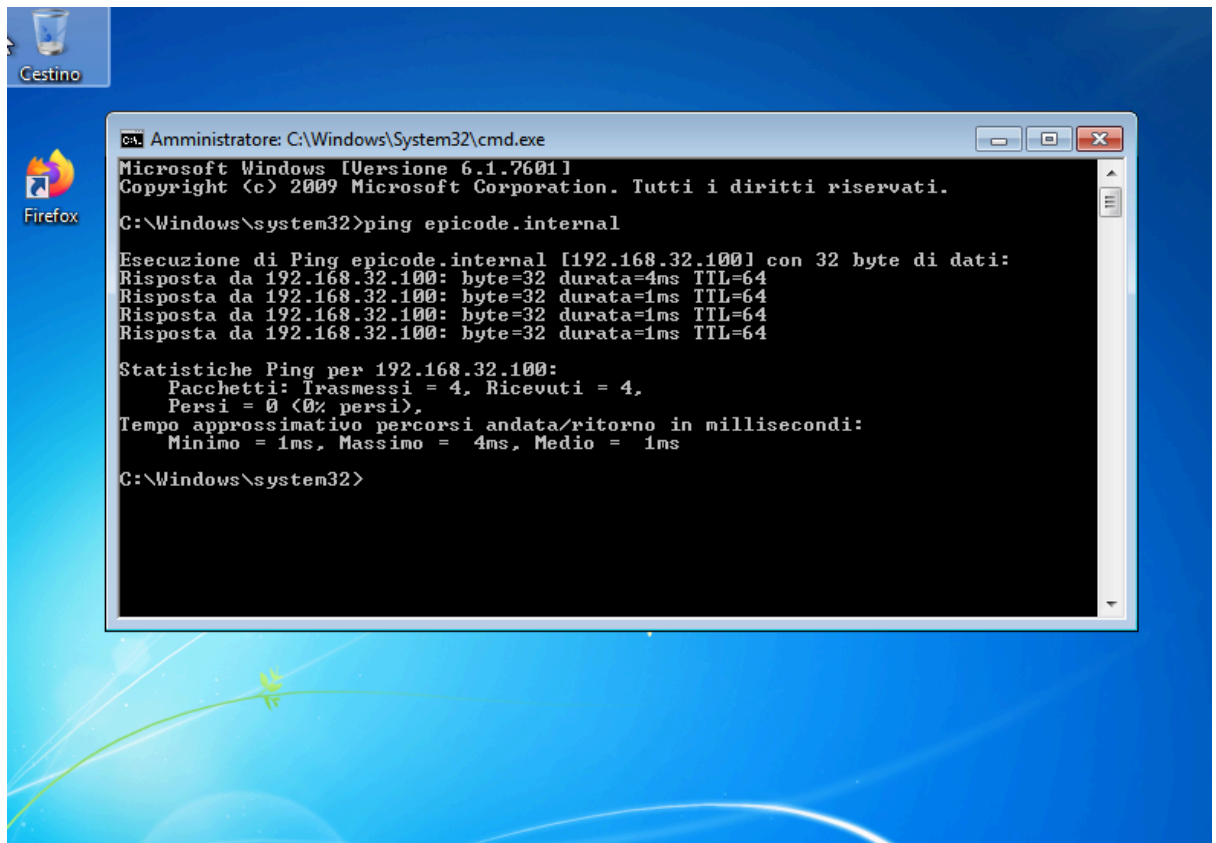
192.168.32.100 epicode.internal

- **Configurazione di Windows 7 Pro**

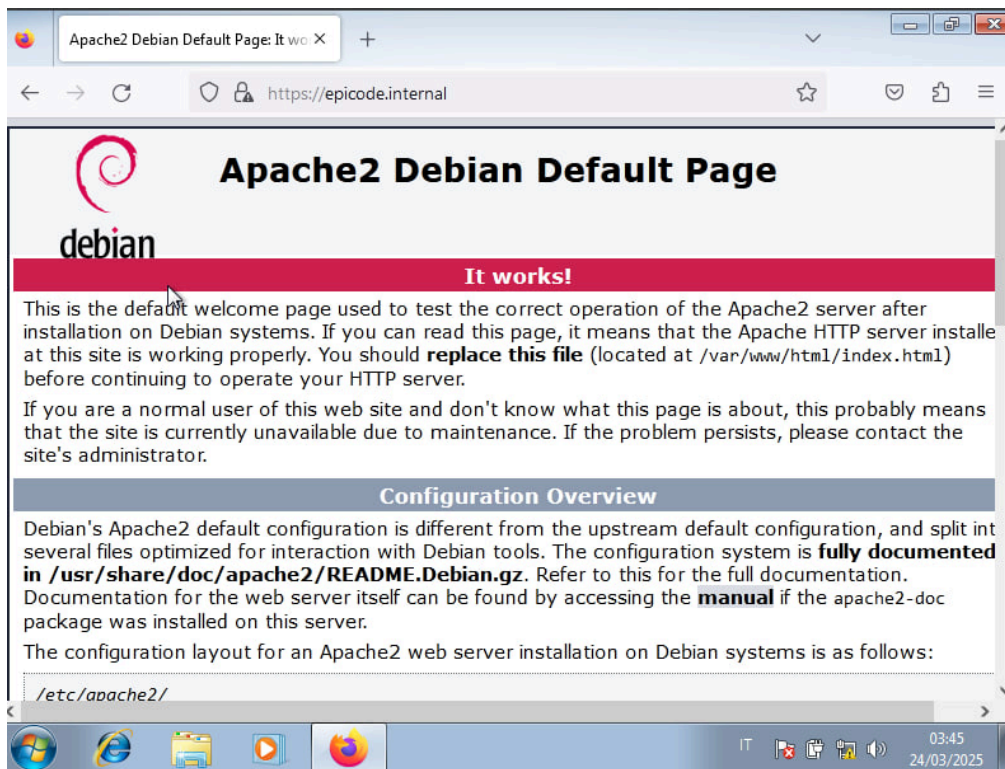
Infine configuro Windows 7, modificando le proprietà del protocollo TCP/IPv4 come da schermata :



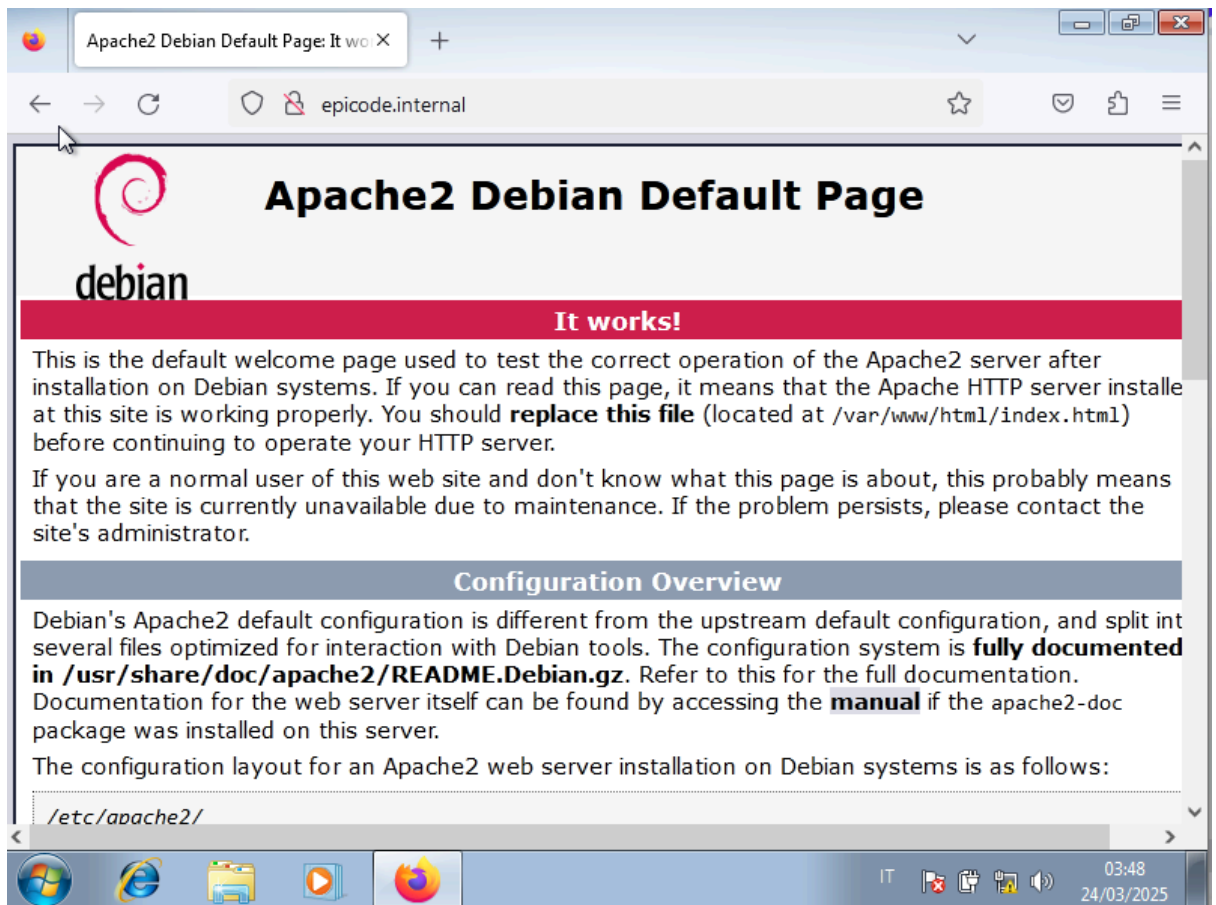
Per vedere se è tutto configurato correttamente attraverso il prompt dei comandi di Windows eseguo un ping : *ping epicode.internal*



Infine concludo la configurazione verificando se dal Browser di Windows 7 raggiungo Apache sia tramite il protocollo https che tramite http:





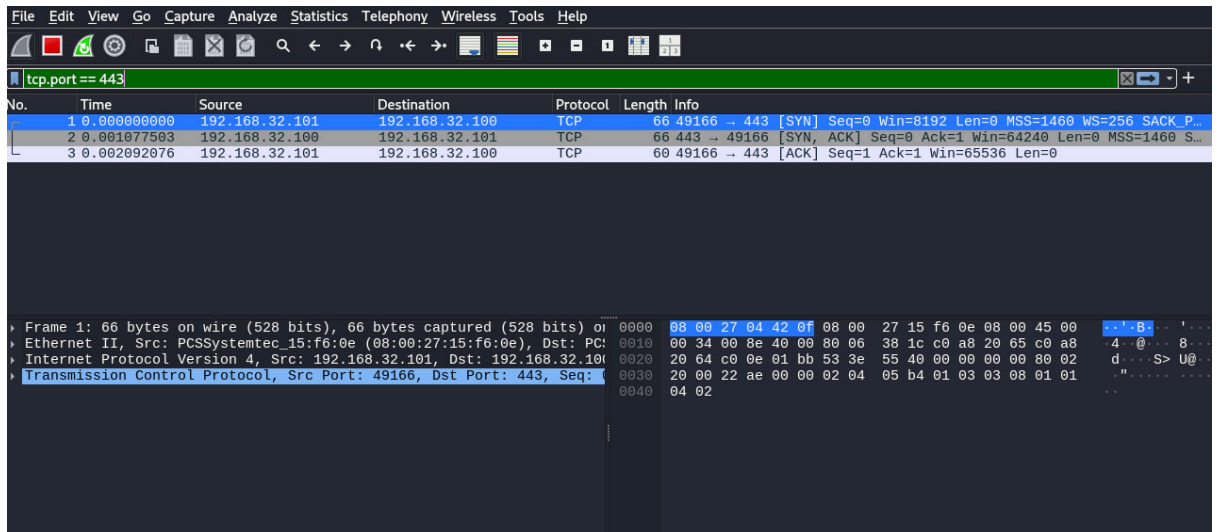


#### 4. Cattura del Traffico con Wireshark

Come richiesto avvio Wireshark e applico il filtro `tcp.port == 80`, dopodiché da Windows 7, apro un browser e visito <http://epicode.internal>

No.	Time	Source	Destination	Protocol	Length	Info
3	0.001093408	192.168.32.101	192.168.32.100	TCP	66	49175 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM
4	0.001279444	192.168.32.100	192.168.32.101	TCP	66	80 → 49175 [ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM
5	0.002515902	192.168.32.101	192.168.32.100	TCP	60	49175 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
8	13.650283296	192.168.32.101	192.168.32.100	TCP	60	49175 → 80 [PSH, ACK] Seq=1 Ack=1 Win=65536 Len=1
9	13.650257818	192.168.32.100	192.168.32.101	TCP	54	80 → 49175 [ACK] Seq=1 Ack=2 Win=64256 Len=0
10	13.939722880	192.168.32.101	192.168.32.100	TCP	60	49175 → 80 [PSH, ACK] Seq=2 Ack=1 Win=65536 Len=1 [TCP PDU reassemble
11	13.940661151	192.168.32.100	192.168.32.101	TCP	54	80 → 49175 [ACK] Seq=1 Ack=3 Win=64256 Len=0
12	14.156057223	192.168.32.101	192.168.32.100	TCP	60	49175 → 80 [PSH, ACK] Seq=3 Ack=1 Win=65536 Len=1 [TCP PDU reassemble
13	14.156163248	192.168.32.100	192.168.32.101	TCP	54	80 → 49175 [ACK] Seq=1 Ack=4 Win=64256 Len=0
14	14.372540903	192.168.32.101	192.168.32.100	TCP	60	49175 → 80 [PSH, ACK] Seq=4 Ack=1 Win=65536 Len=1 [TCP PDU reassemble
15	14.372637282	192.168.32.100	192.168.32.101	TCP	54	80 → 49175 [ACK] Seq=1 Ack=5 Win=64256 Len=0
16	15.908439040	192.168.32.101	192.168.32.100	TCP	60	49175 → 80 [PSH, ACK] Seq=5 Ack=1 Win=65536 Len=1 [TCP PDU reassemble
17	15.908446630	192.168.32.100	192.168.32.101	TCP	54	80 → 49175 [ACK] Seq=1 Ack=6 Win=64256 Len=0
18	16.184644660	192.168.32.101	192.168.32.100	TCP	60	49175 → 80 [PSH, ACK] Seq=6 Ack=1 Win=65536 Len=1 [TCP PDU reassemble
19	16.184851513	192.168.32.100	192.168.32.101	TCP	54	80 → 49175 [ACK] Seq=1 Ack=7 Win=64256 Len=0
20	16.317970145	192.168.32.101	192.168.32.100	TCP	60	49175 → 80 [PSH, ACK] Seq=7 Ack=1 Win=65536 Len=1 [TCP PDU reassemble
21	16.318135140	192.168.32.100	192.168.32.101	TCP	54	80 → 49175 [ACK] Seq=1 Ack=8 Win=64256 Len=0
22	16.490326380	192.168.32.101	192.168.32.100	TCP	60	49175 → 80 [PSH, ACK] Seq=8 Ack=1 Win=65536 Len=1 [TCP PDU reassemble
23	16.490473711	192.168.32.100	192.168.32.101	TCP	54	80 → 49175 [ACK] Seq=1 Ack=9 Win=64256 Len=0
24	16.641298223	192.168.32.101	192.168.32.100	TCP	60	49175 → 80 [PSH, ACK] Seq=9 Ack=1 Win=65536 Len=1 [TCP PDU reassemble
25	16.641413865	192.168.32.100	192.168.32.101	TCP	54	80 → 49175 [ACK] Seq=1 Ack=10 Win=64256 Len=0

sempre dal Browser digito <https://epicode.internal> e visualizzo questa volta attraverso il filtro tcp.port == 443



## 5. Analisi e Conclusioni

*Partendo dall'esercizio che ho svolto sopra riesco ora a trarre diverse conclusioni approfondite sull'importanza di costruire un laboratorio virtuale per fare esperimenti, specialmente quando si tratta di analizzare protocolli di rete come HTTP e HTTPS.*

### 1. Importanza della cifratura nella comunicazioni di rete

Ho chiaramente compreso le differenze tra HTTP e HTTPS:

- HTTP: I dati sono trasmessi in chiaro, rendendoli vulnerabili a intercettazioni, attacchi man-in-the-middle e sniffing. Con strumenti come Wireshark, è possibile leggere il contenuto delle richieste e delle risposte, inclusi dati sensibili come credenziali o informazioni personali.
- HTTPS: I dati sono cifrati utilizzando protocolli come TLS/SSL, rendendo il traffico illeggibile a chiunque intercetti la comunicazione. Ciò a vantaggio della riservatezza.

In effetti, senza una cifratura ci si può esporre a rischi significativi, come il furto di dati o l'alterazione delle informazioni trasmesse.



## **2. Vantaggi e opportunità nel costruire un laboratorio virtuale**

Ho compreso che creare un ambiente virtuale con Windows 7 e Kali Linux offre numerosi vantaggi per fare esperimenti in modo sicuro e controllato:

- Isolamento e sicurezza
  - Un laboratorio virtuale mi permette di isolare l'ambiente di test dalla rete reale, evitando di esporre il mio sistema operativo a potenziali rischi. Posso sperimentare con configurazioni di rete, attacchi e difese senza compromettere l'integrità del mio computer di lavoro.
- Flessibilità e riproducibilità
  - Posso facilmente clonare, ripristinare o modificare l'ambiente virtuale permettendomi di testare diverse configurazioni in modo rapido e riproducibile. Inoltre posso simulare scenari complessi, come reti con più nodi, firewall, server e client, senza la necessità di comprare hardware fisico.