



# Progetto settimana 1 EPICODE

Giovanni Pisapia

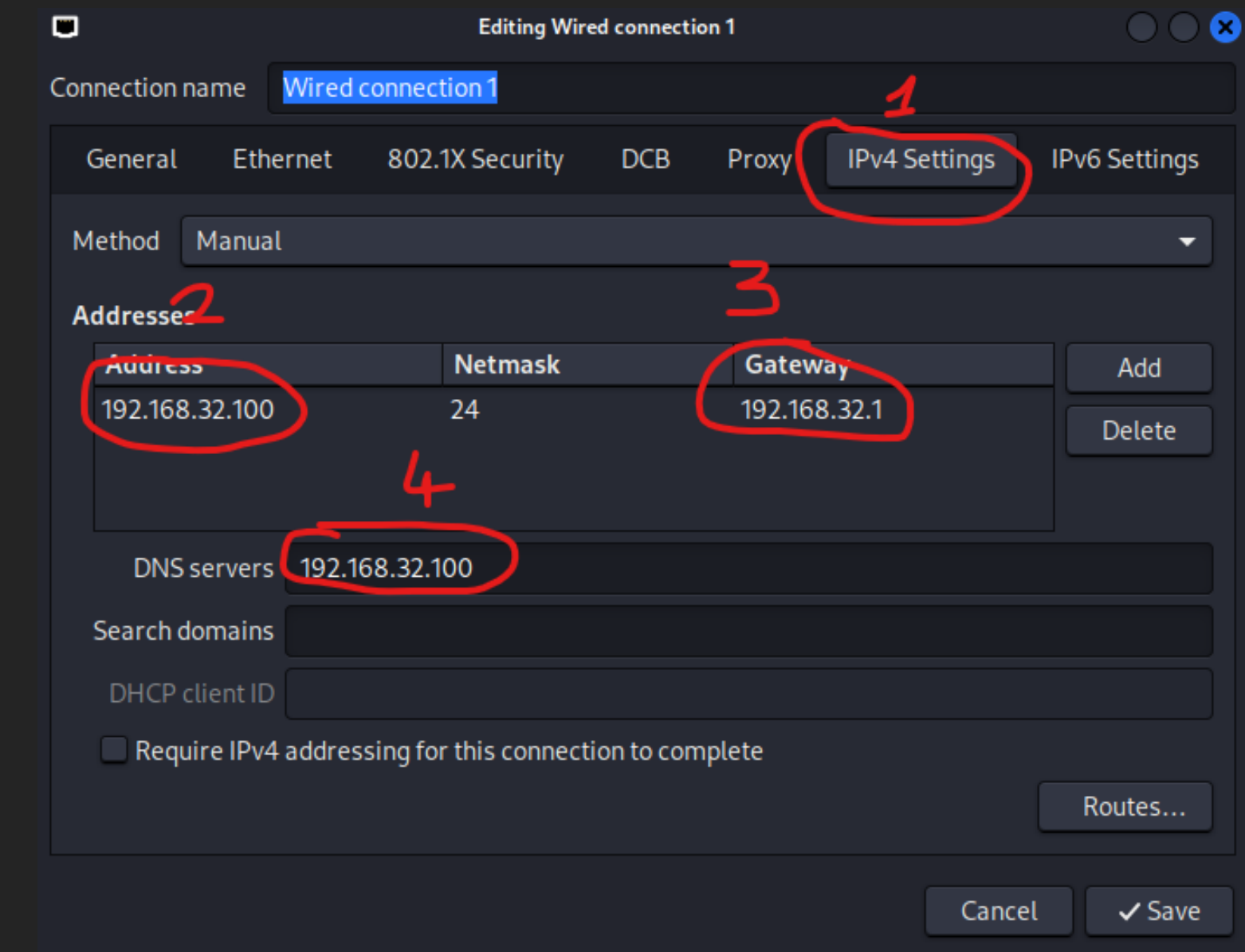
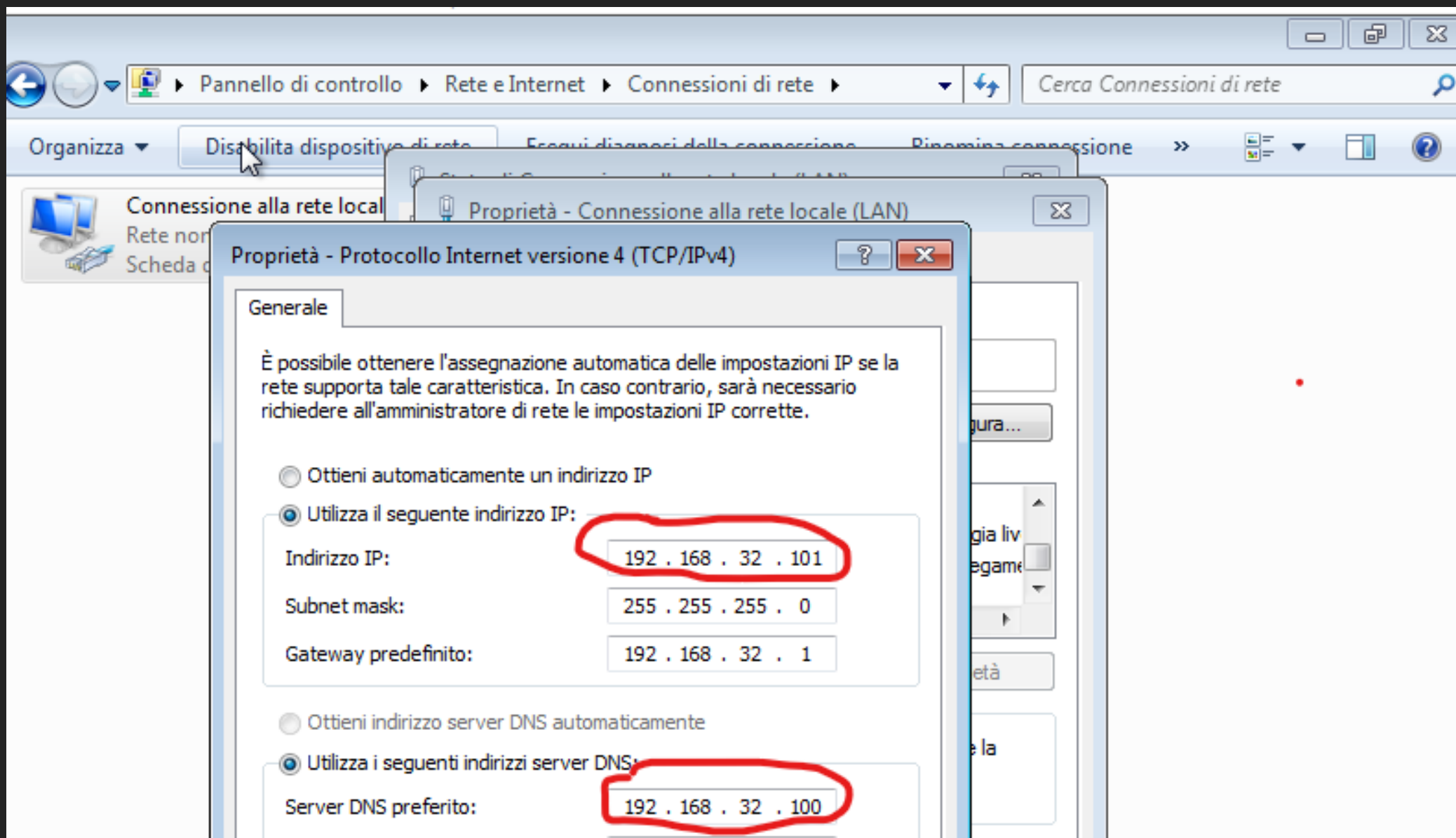


# STEP 1

## Cambio indirizzo ip su macchina Windows 7 e Linux

### Windows 7

### Linux

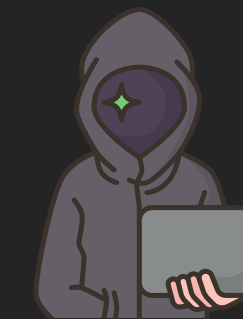


1. Ho cliccato sul pulsante Start.
2. Ho selezionato il Pannello di Controllo.
3. Ho cercato la voce Centro connessioni di Rete e Condivisione e ho cliccato su Modifica impostazioni scheda.
4. Ho individuato la connessione di rete per la quale volevo cambiare l'indirizzo IP.
5. Ho fatto clic destro sulla connessione selezionata e ho scelto Proprietà.
6. Nella finestra che si è aperta, ho selezionato Protocollo Internet versione 4 (TCP/IPv4).
7. Ho fatto clic su Proprietà.
8. Ho selezionato l'opzione di utilizzo degli indirizzi IP statici.
9. Ho inserito l'indirizzo IP (192.168.32.101), la subnet mask (255.255.255.0), il gateway predefinito (192.168.32.1) e il server DNS (192.168.32.100).
10. Ho fatto clic su OK per salvare le modifiche.
11. Ho chiuso tutte le finestre.

1. Accedere alle impostazioni di rete del sistema operativo
2. Selezionare la scheda di rete per cui si vuole modificare l'indirizzo IP
3. Fare clic sul pulsante "Impostazioni IPv4" (o "IPv6" se si vuole configurare un indirizzo IPv6)
4. Selezionare l'opzione "Manuale"
5. Inserire l'indirizzo IP desiderato nel campo "Indirizzo"
6. Inserire la subnet mask nel campo "Maschera di rete"
7. Inserire il gateway predefinito nel campo "Gateway"
8. Inserire uno o più server DNS nell'elenco "Server DNS"
9. Fare clic sul pulsante "Applica" per salvare le modifiche

# STEP 2

## Configurazione del server HTTP/HTTPS



Prima di tutto, ho aperto il terminale di Kali Linux.

Per configurare Inetsim come simulatore della mia rete, ho eseguito il comando "sudo nano etc/inetsim/inetsim.conf".

Di default, tutti i comandi in Inetsim sono preceduti dal simbolo #, che in programmazione indica che il comando è commentato e quindi non viene letto dall'interprete. Per attivare i servizi necessari per l'esercizio, ho dovuto rimuovere il simbolo #.

Come primo passaggio, ho modificato la riga "service\_bind\_address" a "192.168.32.100", in modo che tutto il traffico inviato a Inetsim venisse indirizzato a questo indirizzo IP.

In seguito, ho impostato la modalità di risoluzione del DNS con la riga "#dns\_default\_ip 192.168.32.100".

Per configurare un DNS statico che resolvesse il mio nome di dominio in un indirizzo IP specifico, ho usato la riga "dns\_static epicode.internal 192.168.32.100".

Infine, ho attivato i servizi HTTP e HTTPS decommentando le righe corrispondenti alle porte 80 e 443. Come sappiamo, HTTP funziona sulla porta 80 e HTTPS sulla porta 443.

Una volta terminata la configurazione di Inetsim, ho salvato le modifiche e chiuso il file.

```
#  
# Syntax: service_bind_address <IP address>  
#  
# Default: 127.0.0.1
```

```
service_bind_address 192.168.32.100
```

```
# Default: 127.0.0.1  
#  
dns_default_ip 192.168.32.100
```

```
#####  
# dns_default_hostname
```

```
#  
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
```

```
#  
# Default: 80  
#  
http_bind_port 80
```

```
#####  
# http_version
```

# STEP 3

## Prova funzionamento HTTP/HTTPS



Per verificare se ho configurato inetsim correttamente, ho seguito i seguenti passaggi:

Ho aperto il terminale Linux e ho eseguito il comando "sudo su" per accedere ai permessi di root.

Ho inserito il comando "inetsim" per avviare la simulazione di rete.

Successivamente, ho aperto la macchina virtuale Windows 7 e ho avviato il browser Internet Explorer.

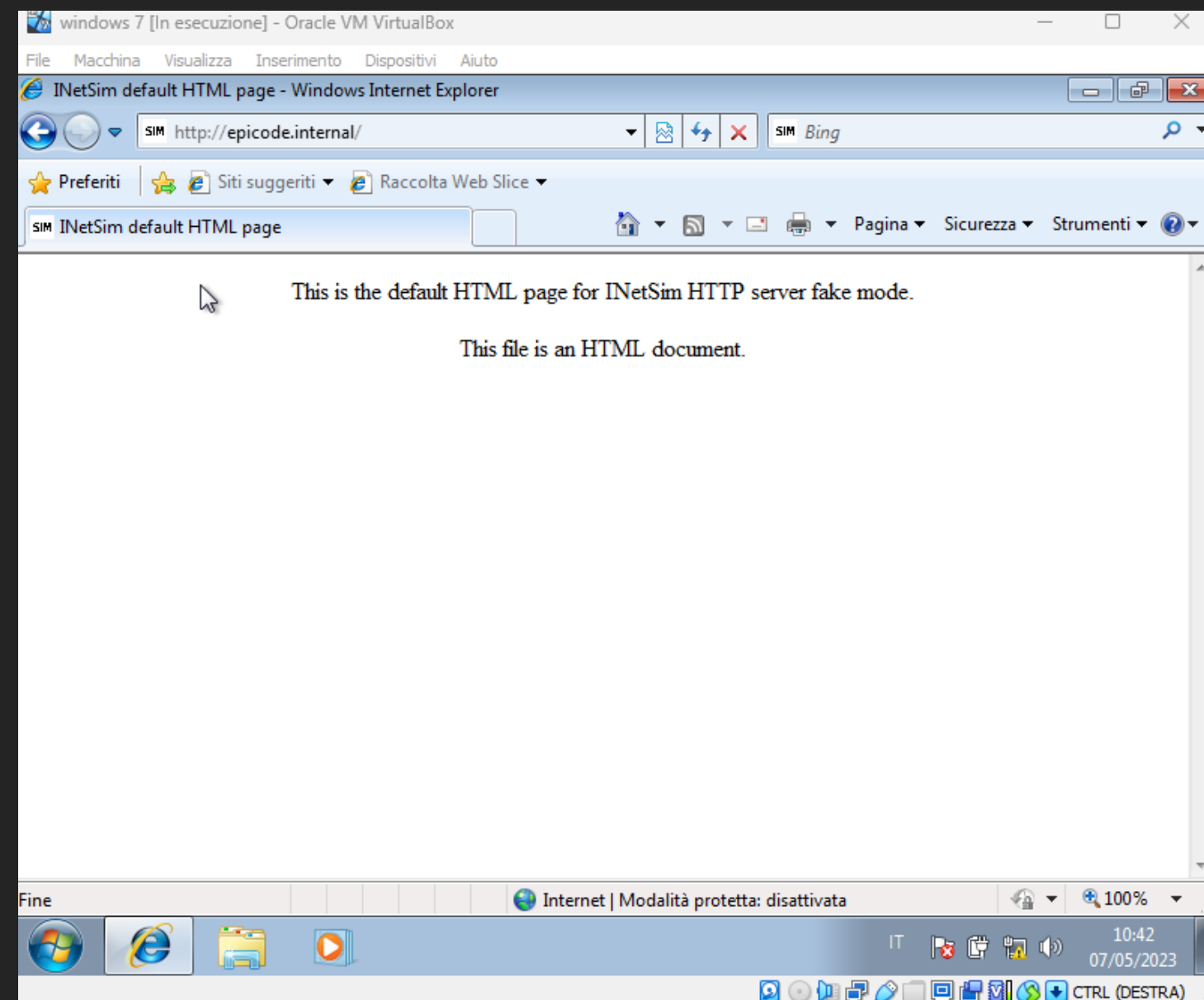
Nella barra di ricerca del browser, ho inserito il link "epicode.internal" e ho premuto il tasto invio.

Ho verificato se il server DNS appena configurato risolve correttamente il nome di dominio testando sia la connessione http che https.

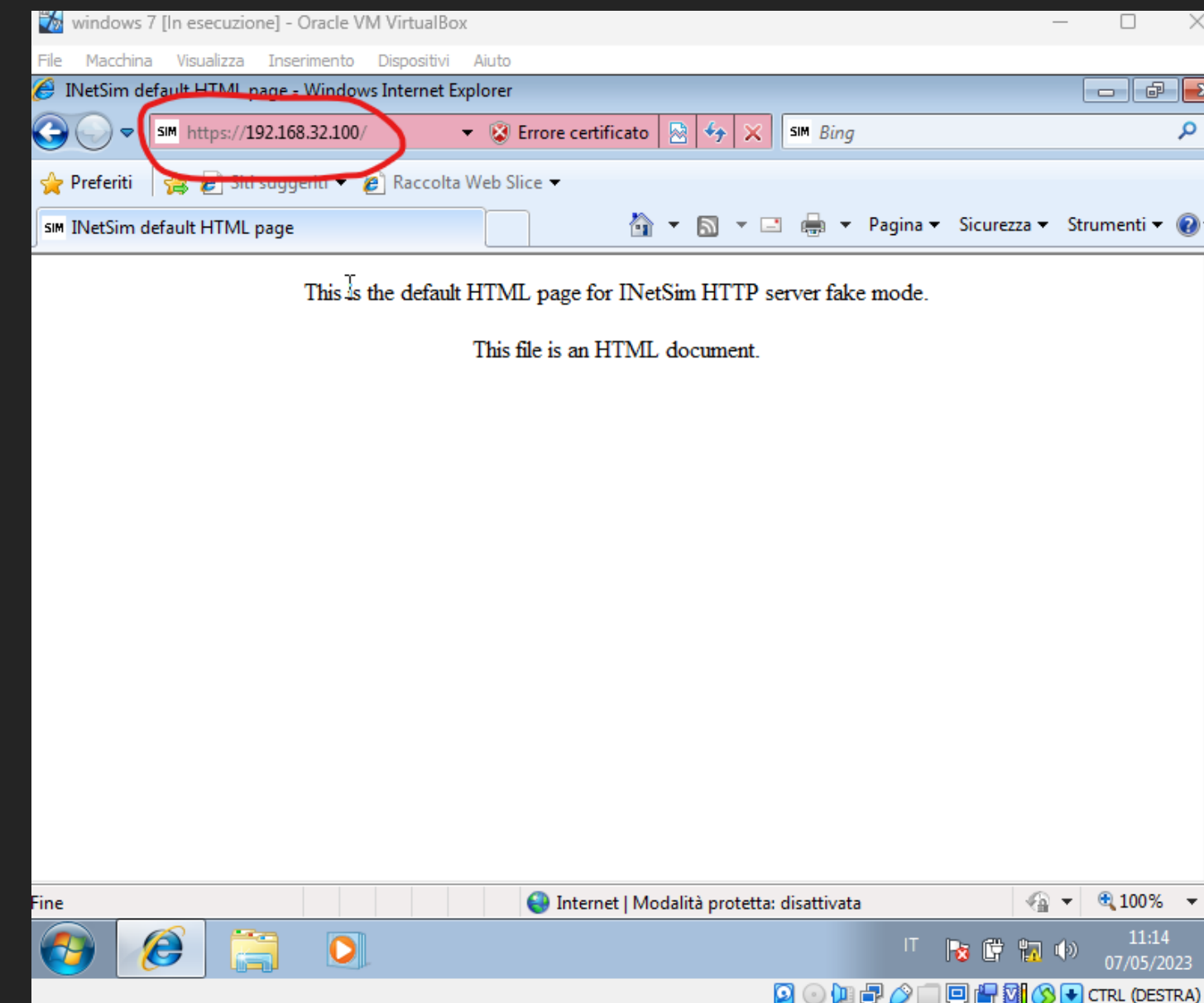
Se la configurazione è stata effettuata correttamente, avrei dovuto visualizzare una pagina predefinita di inetsim a conferma del corretto funzionamento del simulatore di rete.

In questo modo, ho potuto verificare la corretta configurazione di inetsim e l'abilitazione dei servizi necessari.

## HTTP



## HTTPS





# STEP 4

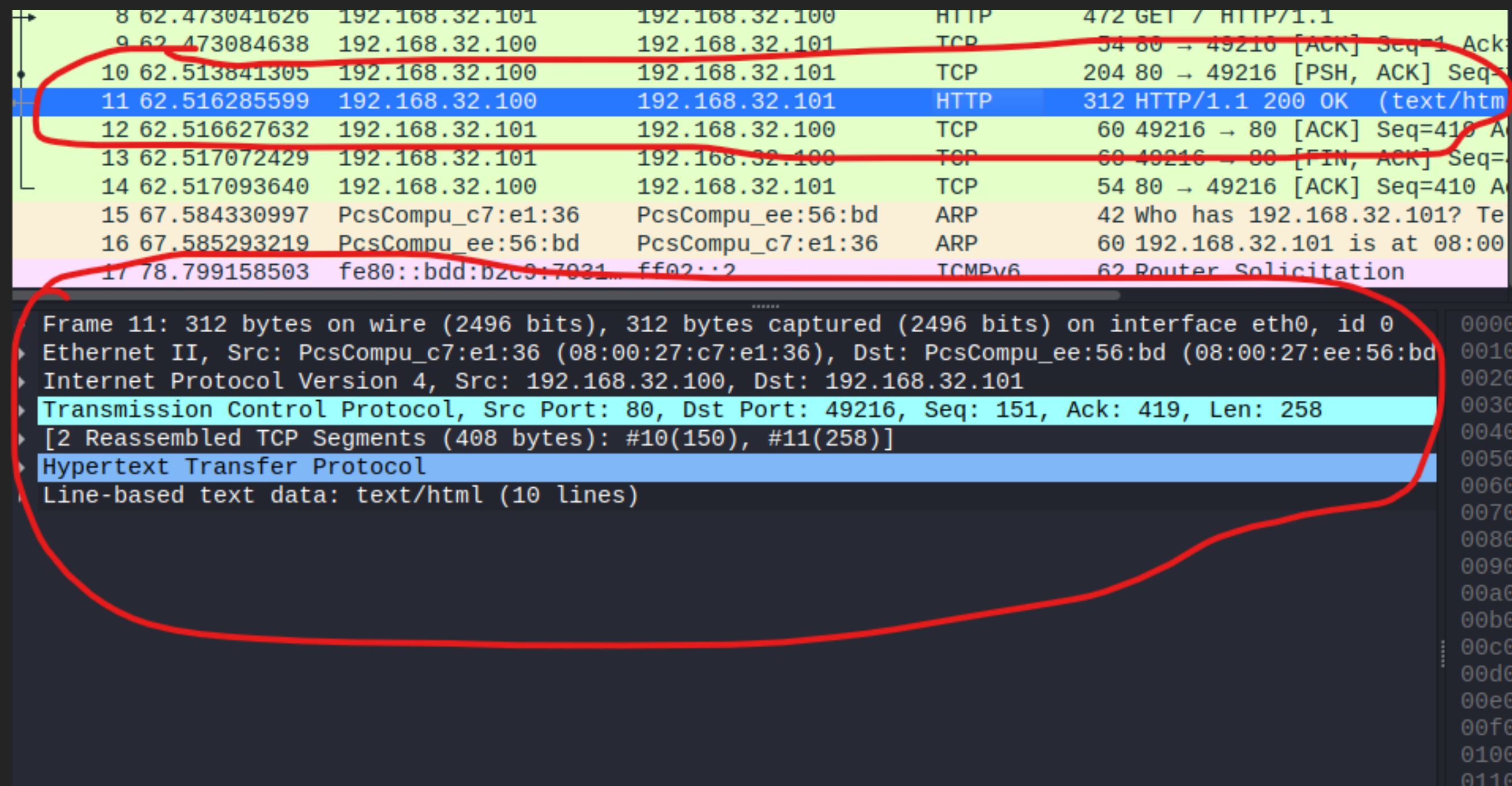
## Intercettazione del traffico http tramite Wireshark



Come primo passo, ho avviato inetsim tramite il terminale di Kali Linux. Successivamente, ho avviato Wireshark, selezionato l'interfaccia di rete corrispondente e avviato la scansione del traffico di rete.

In seguito, mi sono spostato sulla macchina virtuale Windows 7 e ho aperto il browser. All'interno della barra di ricerca, ho digitato "http://epicode.internal" per richiedere la risorsa al server HTTP.

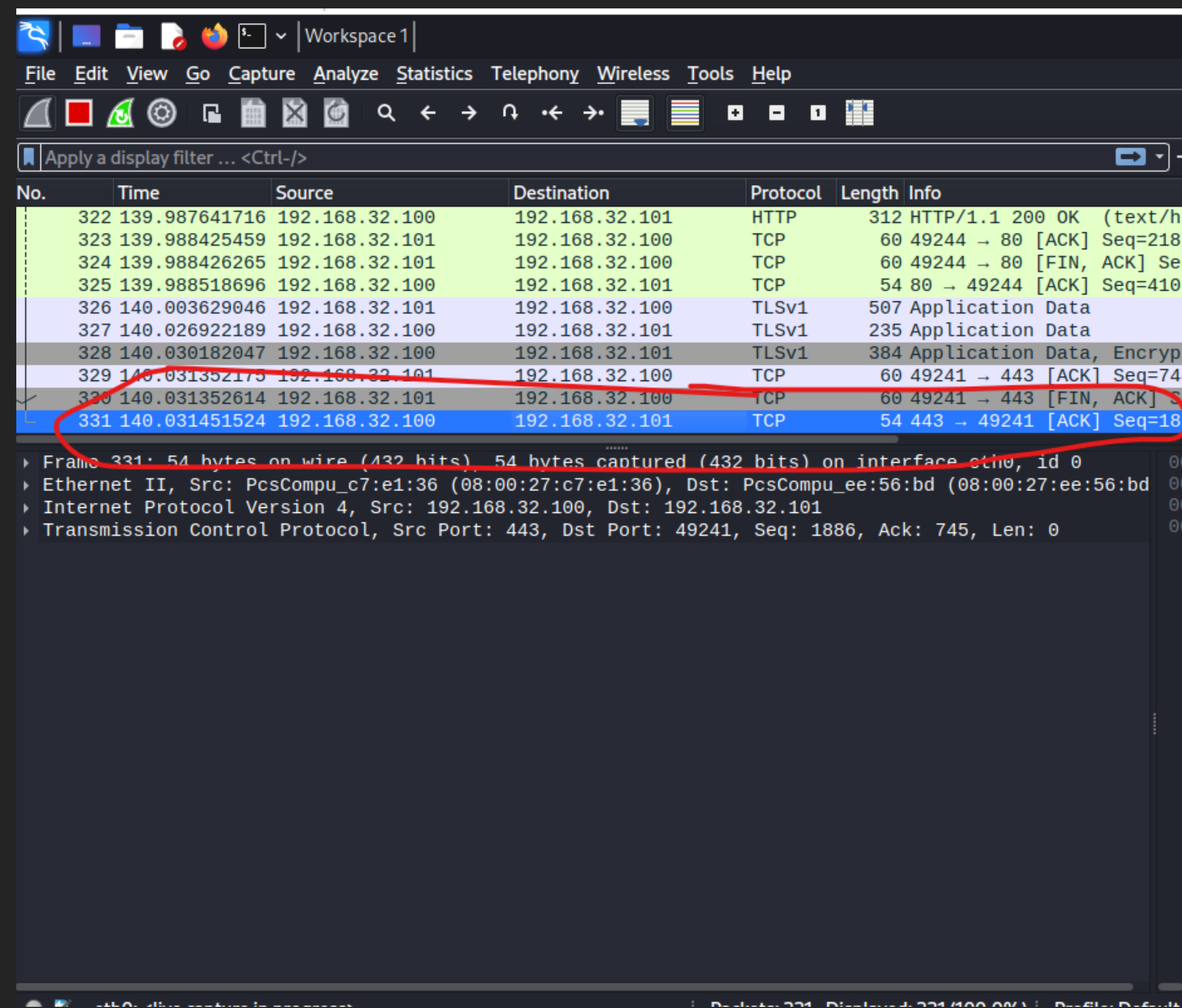
Infine, ho controllato i pacchetti ricevuti su Wireshark sulla macchina Kali Linux, evidenziando i MAC address di sorgente e destinazione e il contenuto della richiesta HTTP.



# STEP 5

## Intercettazione del traffico https tramite Wireshark

Per effettuare il secondo test, ho sostituito il server HTTPS con un server HTTP e ho ripetuto la procedura di richiesta della risorsa tramite il browser. Ho nuovamente controllato i pacchetti ricevuti su Wireshark, evidenziando le differenze tra il traffico HTTP e quello HTTPS.



No.	Time	Source	Destination	Protocol	Length	Info
322	139.987641716	192.168.32.100	192.168.32.101	HTTP	312	HTTP/1.1 200 OK (text/html)
323	139.988425459	192.168.32.101	192.168.32.100	TCP	60	49244 → 80 [ACK] Seq=218 A
324	139.988426265	192.168.32.101	192.168.32.100	TCP	60	49244 → 80 [FIN, ACK] Seq=
325	139.988518696	192.168.32.100	192.168.32.101	TCP	54	80 → 49244 [ACK] Seq=410 A
326	140.003629046	192.168.32.101	192.168.32.100	TLSv1	507	Application Data
327	140.026922189	192.168.32.100	192.168.32.101	TLSv1	235	Application Data
328	140.030182047	192.168.32.100	192.168.32.101	TLSv1	384	Application Data, Encrypted
329	140.031352175	192.168.32.101	192.168.32.100	TCP	60	49241 → 443 [ACK] Seq=744
330	140.031352614	192.168.32.101	192.168.32.100	TCP	60	49241 → 443 [FIN, ACK] Seq=
331	140.031451524	192.168.32.100	192.168.32.101	TCP	54	443 → 49241 [ACK] Seq=1886

Frame 331: 54 bytes on wire (432 bits) 54 bytes captured (432 bits) on interface eth0, id 0

Ethernet II, Src: PcsCompu\_c7:e1:36 (08:00:27:c7:e1:36), Dst: PcsCompu\_ee:56:bd (08:00:27:ee:56:bd)

Internet Protocol Version 4, Src: 192.168.32.100, Dst: 192.168.32.101

Transmission Control Protocol, Src Port: 443, Dst Port: 49241, Seq: 1886, Ack: 745, Len: 0





# CONCLUSIONI

In conclusione, questo esercizio ci ha permesso di applicare le nostre conoscenze acquisite sull'architettura client-server, sulla configurazione di un server HTTPS e di un servizio DNS, e sull'intercettazione del traffico utilizzando Wireshark. L'analisi dei pacchetti ricevuti ci ha permesso di comprendere le differenze tra i protocolli HTTPS e HTTP, in particolare riguardo alla sicurezza delle comunicazioni. Infatti, HTTPS garantisce una maggiore sicurezza dei dati scambiati grazie all'utilizzo di un certificato SSL/TLS per criptare le comunicazioni e garantire l'autenticità del sito web. Al contrario, HTTP non prevede alcun tipo di cifratura, esponendo i dati degli utenti a potenziali attacchi informatici.