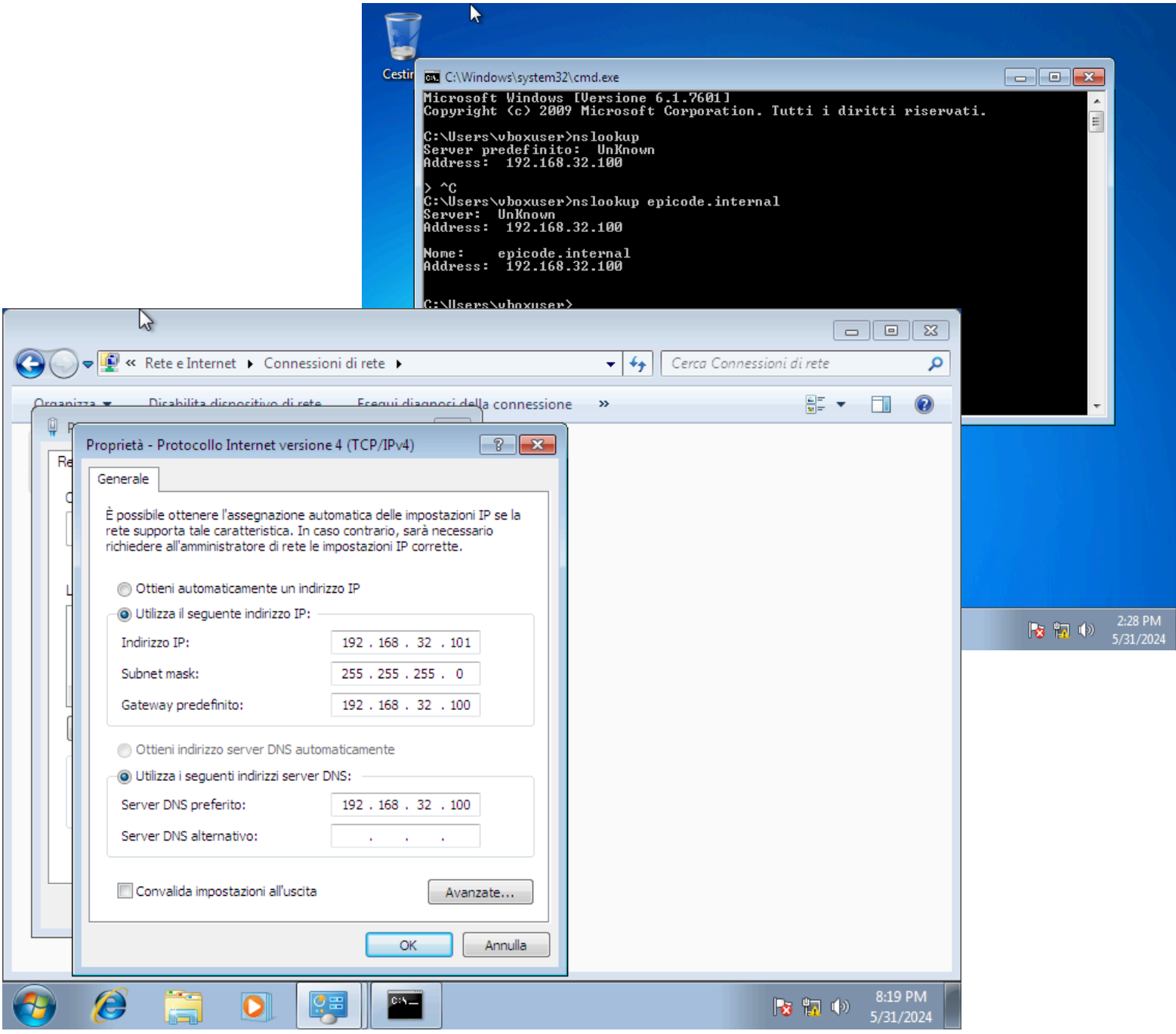
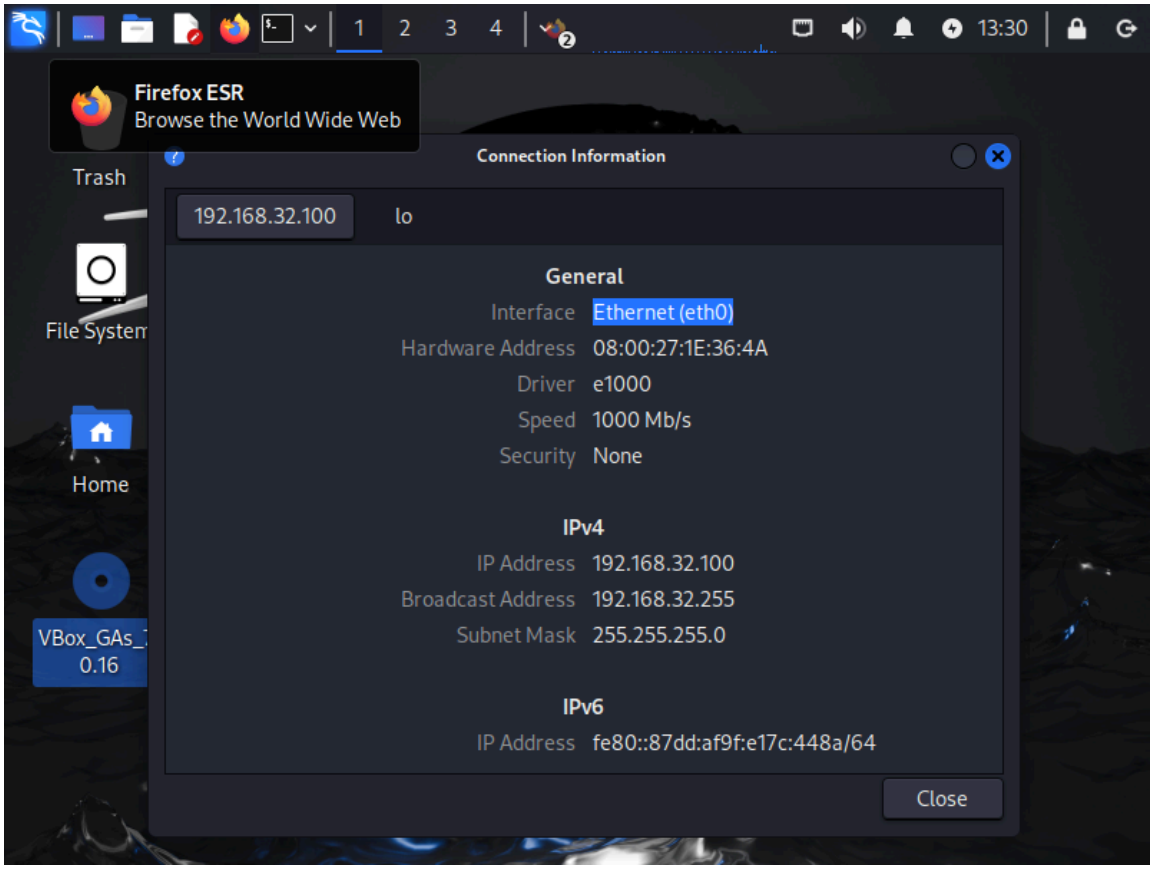
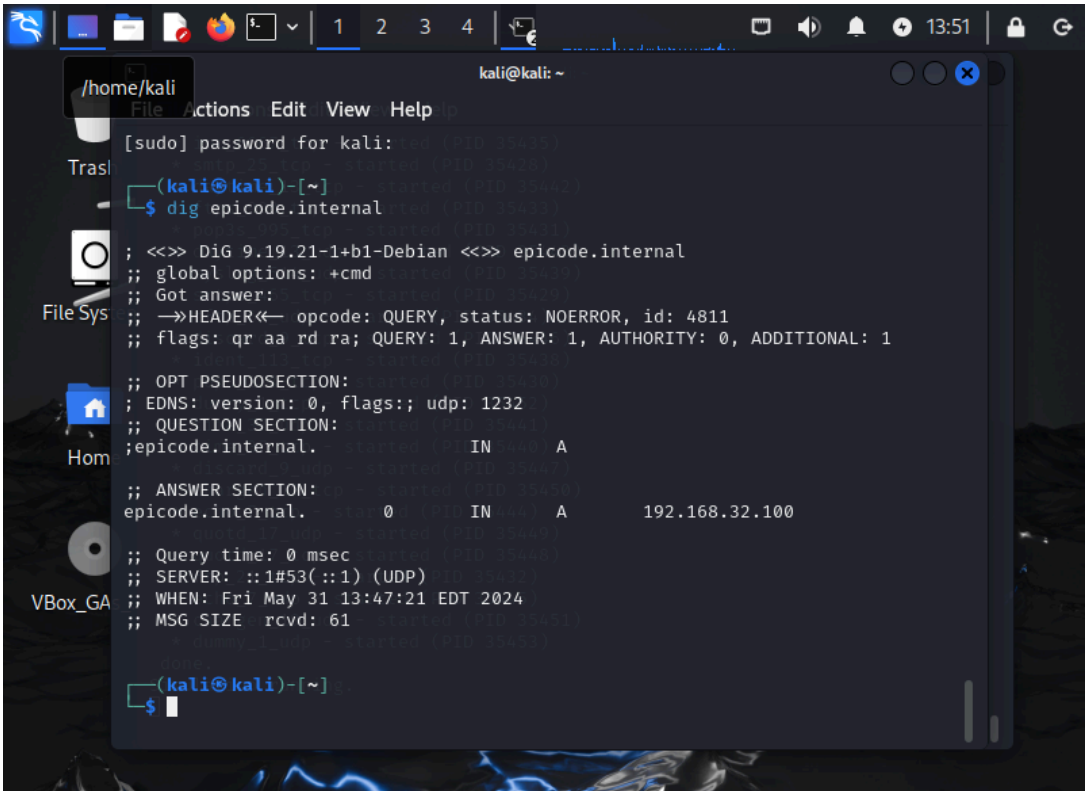
Setup dell'ambiente

- **Client (Windows 7): IP 192.168.32.101**
- **Server (Kali Linux): IP 192.168.32.100**
- **Hostname: epicode.internal**
- **Servizio DNS: attivo per risoluzione dei nomi di dominio**
- **Server HTTPS: attivo**
- **Server HTTP: attivo**





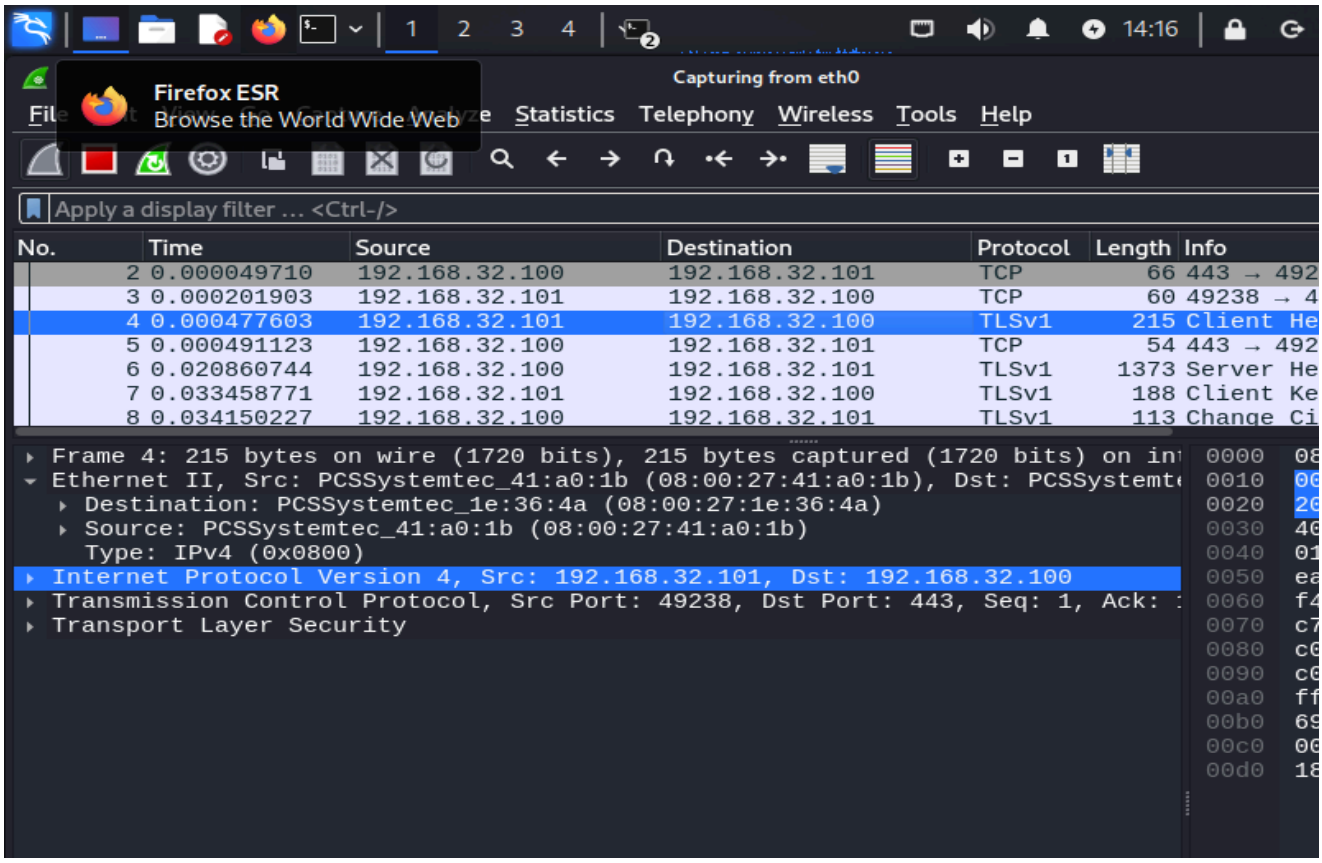
L'esercizio ha permesso di simulare un'architettura client-server in un ambiente di laboratorio virtuale, utilizzando un client con Windows 7 e un server con Kali Linux. Sono state effettuate richieste a un server utilizzando sia HTTPS che HTTP, con l'obiettivo di intercettare e analizzare il traffico di rete tramite Wireshark.
Linux è stato configurato come DNS tramite **dnsmasq**.



Risultati

HTTPS

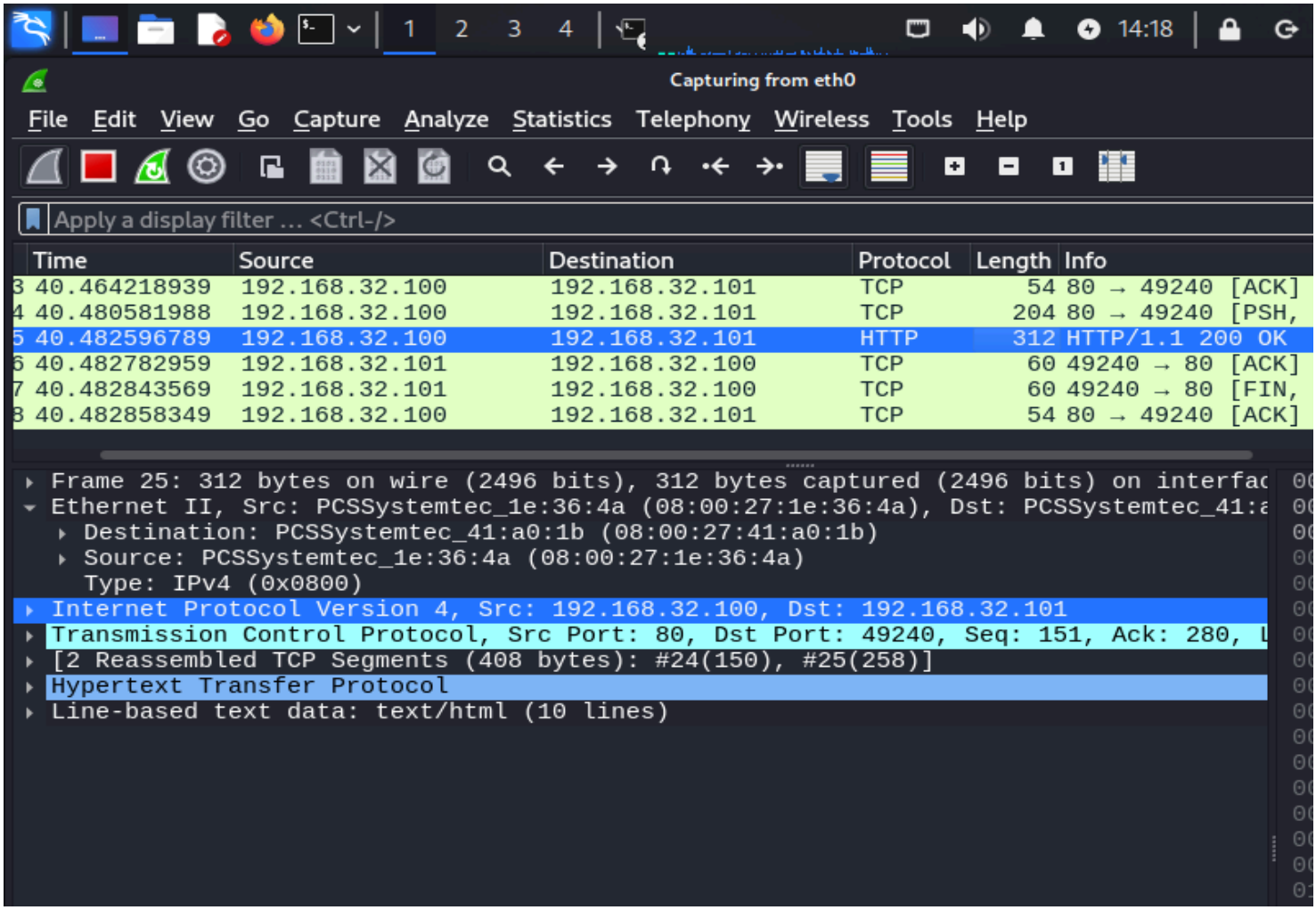
- **Intercettazione**: Utilizzando Wireshark, è stato catturato il traffico generato dalla richiesta HTTPS. Abbiamo identificato correttamente i MAC address di sorgente e destinazione, che corrispondono alle schede di rete del client (Windows 7) e del server (Kali Linux).
- **Contenuto della Richiesta**: Il contenuto della richiesta HTTPS è stato crittografato, rendendolo non leggibile direttamente tramite Wireshark. Solo le fasi iniziali del handshake TLS erano visibili. Questo processo di handshake è fondamentale per stabilire una connessione sicura, negoziando chiavi di crittografia tra il client e il server.



(HTTPS in Wireshark)

HTTP

- **Intercettazione**: Anche il traffico HTTP è stato catturato con Wireshark, e i MAC address di sorgente e destinazione sono stati nuovamente identificati correttamente.
- **Contenuto della Richiesta**: Il contenuto della richiesta HTTP è stato visibile in chiaro. È stato possibile leggere completamente gli header HTTP e il corpo del messaggio, evidenziando la mancanza di crittografia nel protocollo HTTP.



(HTTP in Wireshark)

Differenze Osservate

- **Sicurezza**: La principale differenza tra HTTPS e HTTP è la presenza di crittografia. HTTPS utilizza il protocollo SSL/TLS per proteggere i dati trasmessi, garantendo riservatezza e integrità. Questo rende le comunicazioni HTTPS sicure contro intercettazioni e manomissioni. Al contrario, HTTP non offre alcuna protezione, esponendo i dati trasmessi a potenziali attacchi.
- **Visibilità del Contenuto**: Nel traffico HTTPS, il contenuto della richiesta e della risposta è crittografato e quindi illeggibile senza le chiavi di decrittazione appropriate. Solo le informazioni relative al handshake sono visibili. Nel traffico HTTP, tutto il contenuto è visibile in chiaro, rendendo facile l'intercettazione e l'accesso non autorizzato alle informazioni trasmesse.