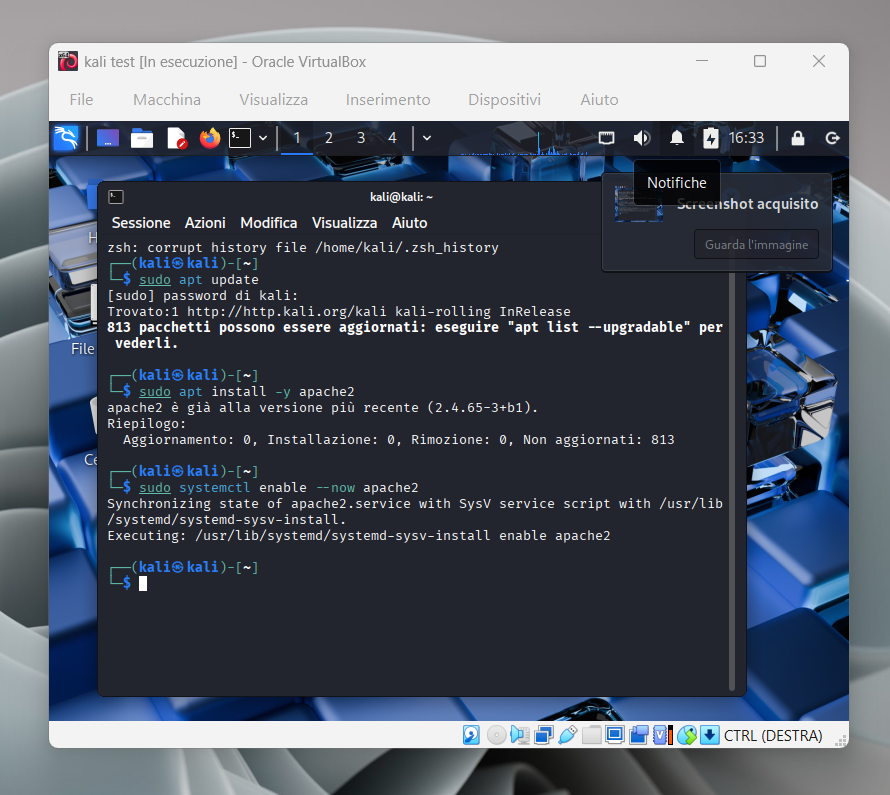
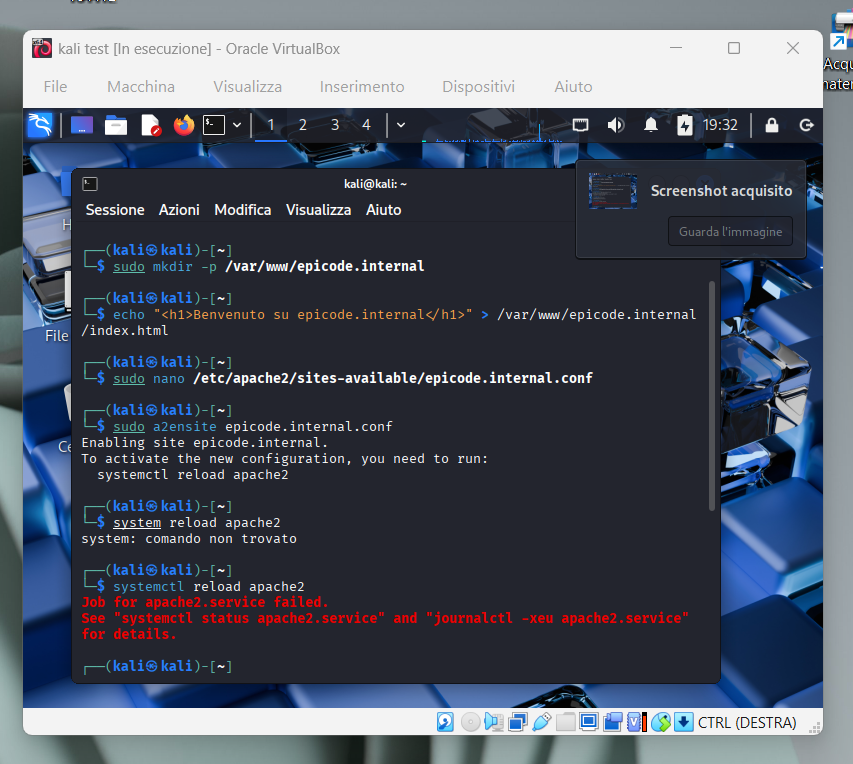
In questo report viene illustrata la simulazione di un’architettura client-server in un mbiente di laboratorio virtuale. L’obiettivo principale è configurare un client windows con indirizzo IP 192.168.1.102 che, tramite un web browser, richiede una risorsa all’hostname epicode.internal. Questa richiesta viene gestita da un server kali linux con indirizzo IP 192.168.1.100, che risponde fornendo la risorsa web richiesta.

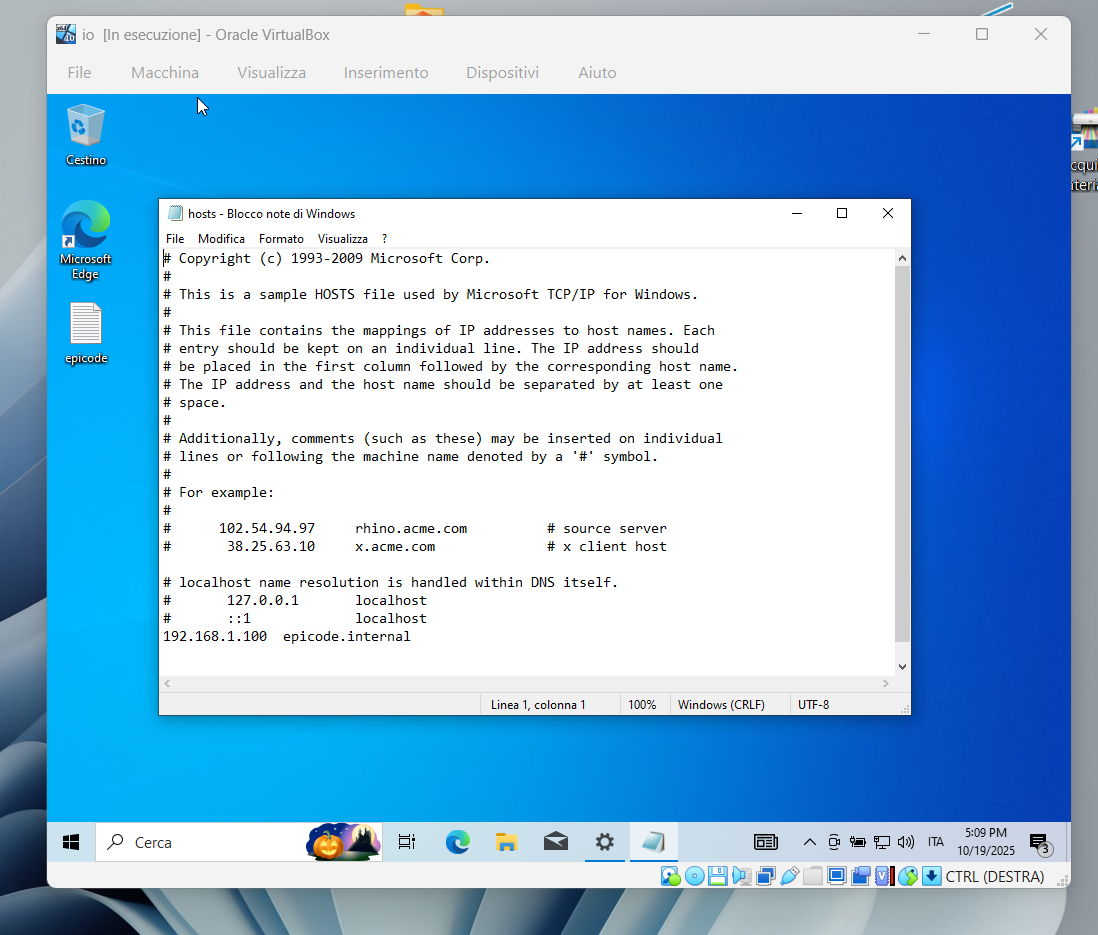
Inizieremo la simulazione aprendo kali linux e attraverso l’ “emulatore terminale” andremo a installare e configurare apache (software opern-source e multipiattaforma che funge da web server, gestendo le richieste tra server e browser per pubblicare siti web)



Come possiamo vedere nell’immagine sopra aprendo l’emulatore terminale andiamo a digitare il comando, sudo apt update, per apportare modifiche al sistema kali dopo fatto ciò per installare apache andremo a digitare, sudo apt install -y apache2, in questo caso è già installato quindi possiamo passare direttamente ad abilitarlo digitando, sudo systemctl enable -- now apache2.

Dopo di che come mostrato nell’immagine sotto andremo ad usare il codice sudo mkdir -p /var/www/epicode.internal che ci servirà per creare la cartella in apache del nostro sito browser



Per ultimare il processo su kali linux dovremmo digitare l’ultimo comando sudo a2ensite epicode.internal.conf , adesso il nostro sito browser è pronto, dobbiamo solo apportare qualche modifica all’altra VM. Su windows 10 andremo ad eseguire il blocco note come amministratore, cliccheremo su file e cercheremo C:\windows\system32\drivers\etc\hosts ed andremo ad aggiungere in fondo la riga l’ip del nostro kali in questo 192.168.1.100 e affianco epicode.internal.

Adesso non ci resta che aprire il nostro sistema di navigazione su windows VM e cercare il sito browser appena creato <http://epicode.internal>

Immagine che contiene testo, elettronica, schermata, software

Il contenuto generato dall'IA potrebbe non essere corretto.

Come possiamo vedere funziona!!

Se lanciamo il nostro sito in http, su wireshark potremmo notare che i pacchetti di dati sono in chiaro e leggibili mentre se lo lanciassimo in https sarebbero crittografati e illegibili, perché è una versione piu’ sicura per navigare e quindi fa in modo che i dati non vengono mostrati.