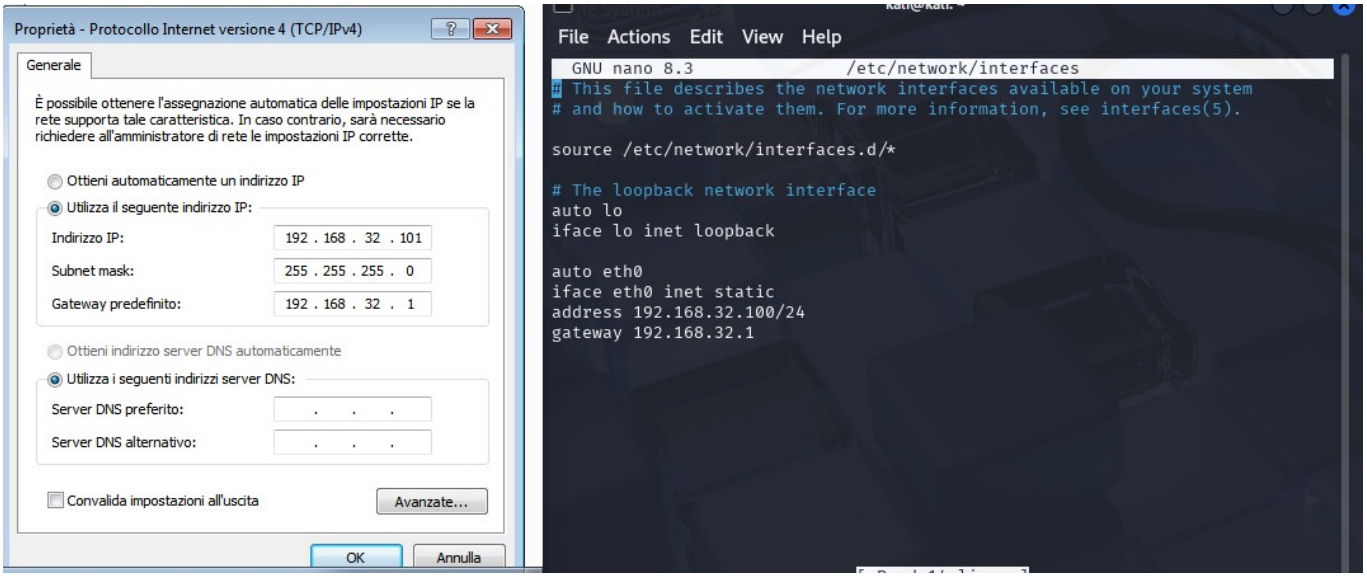
REPORT

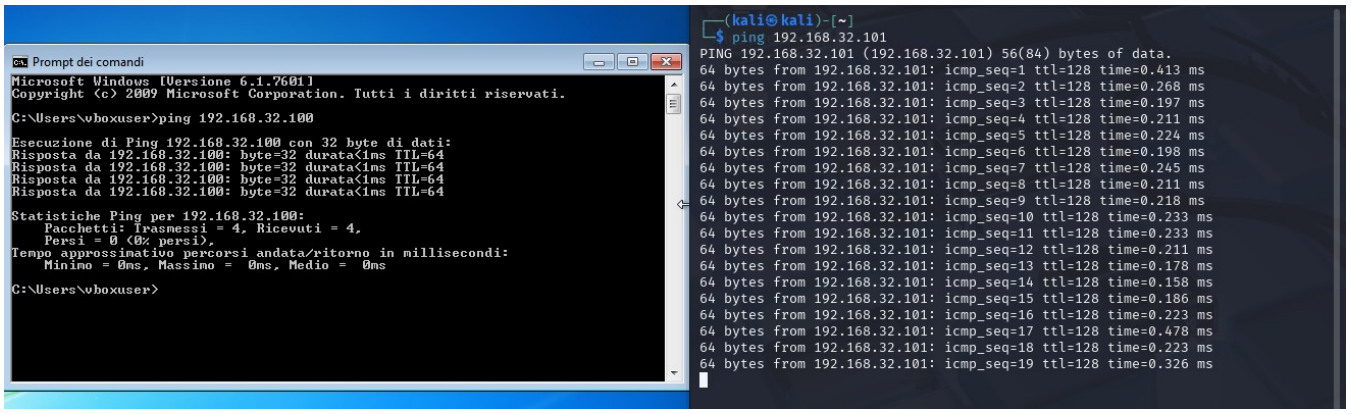
Per svolgere l’esercizio ho deciso di provare a fare qualcosa che mi impegnasse, non avendolo fatto mai, in modo da “fissare” bene i concetti per strutturare un server con certificati ecc.

1. **Preparazione dell’ambiente di lavoro**

- Mi sono assicurato che Kali e Windows fossero sulla stessa rete e che la comunicazione tra loro fosse funzionante quindi ho fatto:

- configurazione ip su Kali tramite terminale
- configurazione su windows su scheda di rete
- eccezione su windows firewall per consentire la comunicazione da kali
- ping da windows a kali e viceversa

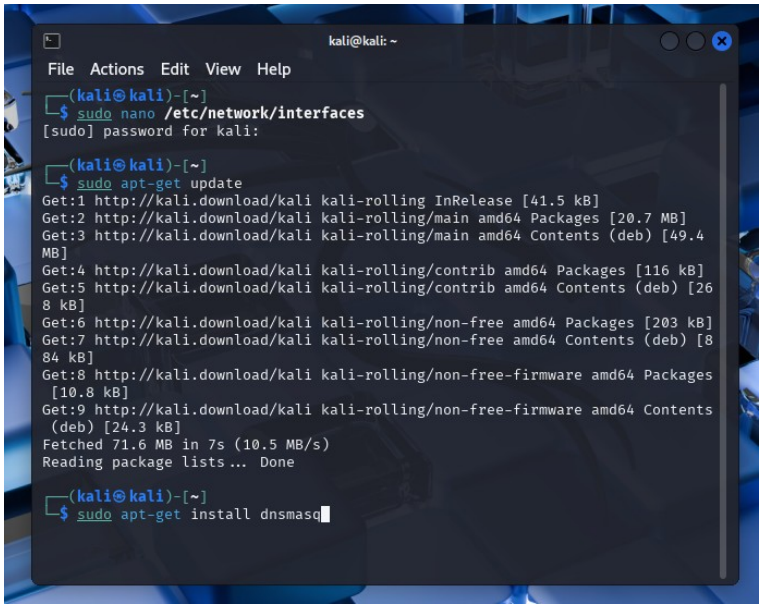


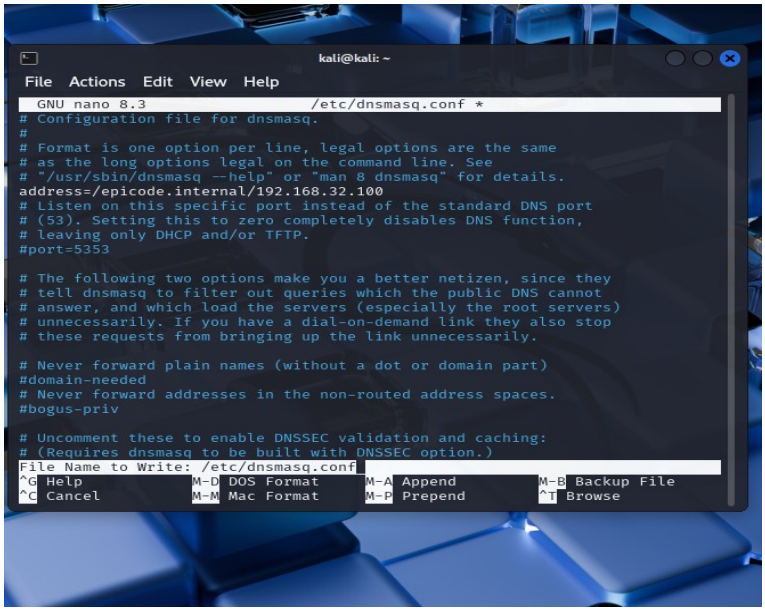


2. **Configurazione del server DNS (DNSMasq**)

- Ho aggiornato i pacchetti, installato DNSMasq e configurato in modo da poter risolvere “epicode.internal“

- **sudo apt-get update** (aggiornamento pacchetti di sistema)
- **sudo apt-get install dnsmasq** (installazione dns)
- **sudo nano /etc/dnsmasq.conf** (configurazione dns)



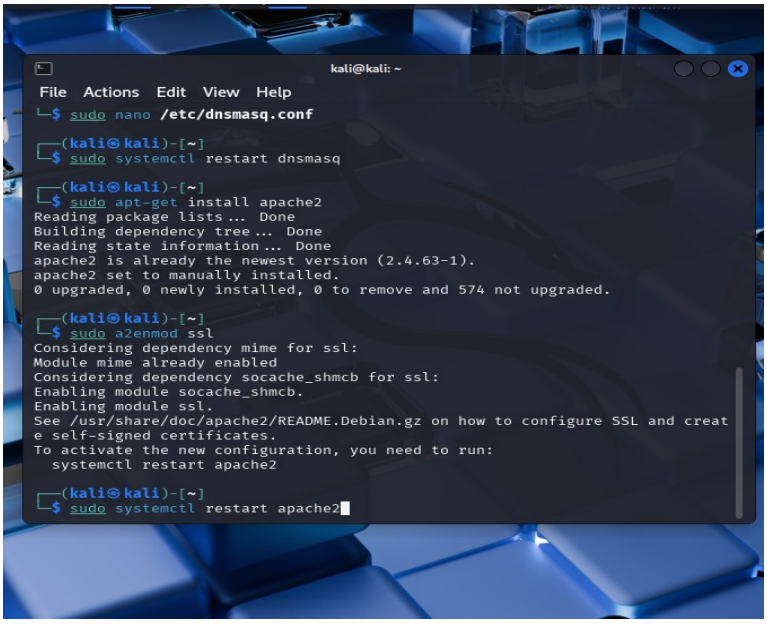


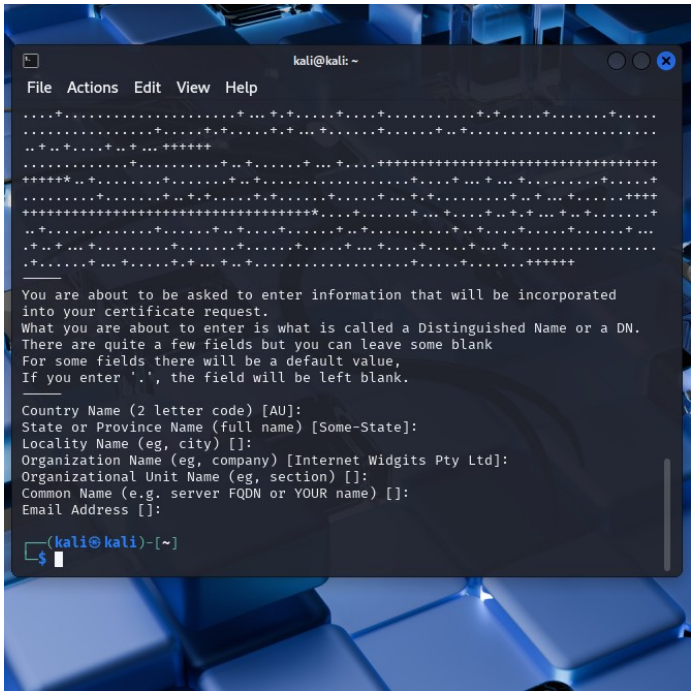
nota : dopo aver installato e configurato dnsmasq bisogna riavviare il servizio con : **sudo systemctl restart dnsmasq**

3. Configurazione server HTTPS (apache) e modulo SSL

- Dopo aver installato e configurato il dns si installa e configura apache, che serà il server che useremo per https e si abilita SSL (con certificato auto firmato) . La procedura per richiamare l’installazione e la configurazione è simile a quella gia vista nel dns

- **sudo apt-get install apache2** (installazione server apache)
- **sudo a2enmod ssl** (abilita il certificato SSL)
- **sudo nano /etc/apache2/sites-available/default-ssl.conf** (configurazione server https)





nota : per creare un certificato auto-firmato bisogna, dopo aver abilitato SSL , digitare :
sudo openssl req -x509 -notes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.ctr

- `-newkey rsa:4096` - Questo parametro ci permette di scegliere la dimensione in bit della nostra chiave RSA. Sebbene 2048 sia il valore di default, è sempre bene specificarla manualmente.
- `-x509` - Specifica che stiamo andando a creare un certificato self-signed.
- `-sha256` - Genera il certificato utilizzando un algoritmo SHA da 256 bit.
- `-days` - Specifica la durata del certificato in giorni.
- `-nodes` - Serve a generare un certificato che non richiede una password. Senza questo parametro, bisognerà inserire la password ogni volta che l'applicazione con cui si usa il certificato viene riavviata.
- `-out` - Indica il percorso dove andremo a salvare il certificato (il nome è a scelta personale).
- `-keyout` - Indica il percorso in cui si andrà a salvare la chiave del certificato (anche qui, il nome è a scelta).

4. Intercettazione del traffico HTTPS e HTTP

- Si passa a Windows , ma prima si attiva Wireshark avviandolo e selezionando l'interfaccia per la connessione (eth0). Su windows prima di accedere al browser e al server bisogna aggiungere una riga al file system32/... per risolvere epicode.internal. Poi si passa al browser e si lancia https://epicode.internal (io ho provato sia con https://192.168.32.100 che con https://epicode.internal). La procedura per HTTPS e HTTP è uguale, basta assicurarsi di disattivare il server HTTPS (**sudo a2disside default-ssl**) e riavviare apache (**sudo systemctl restart apache2**).