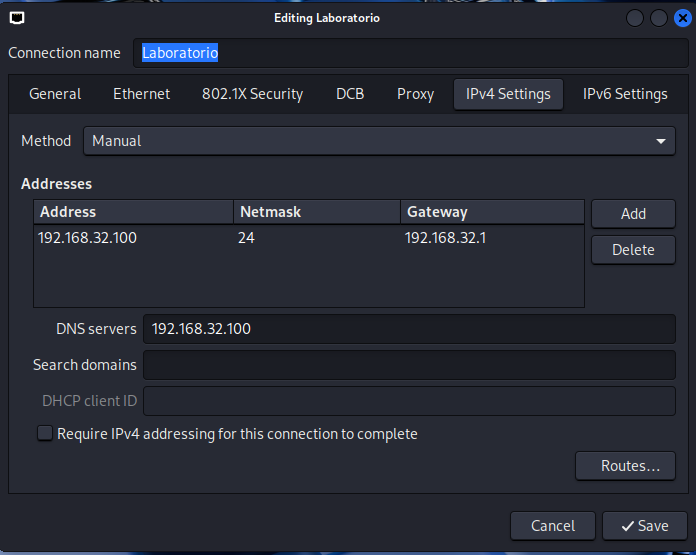
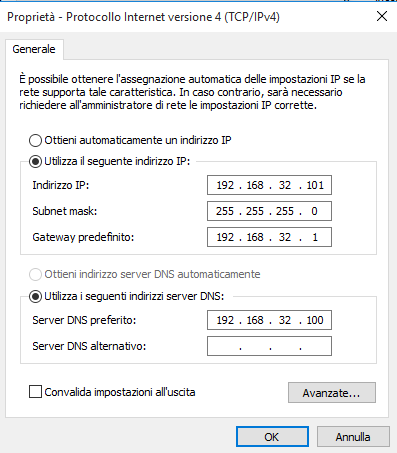
**Progetto di fine Modulo M1  
Antonio Piredda  
20.10.2025**

**INTRODUZIONE**

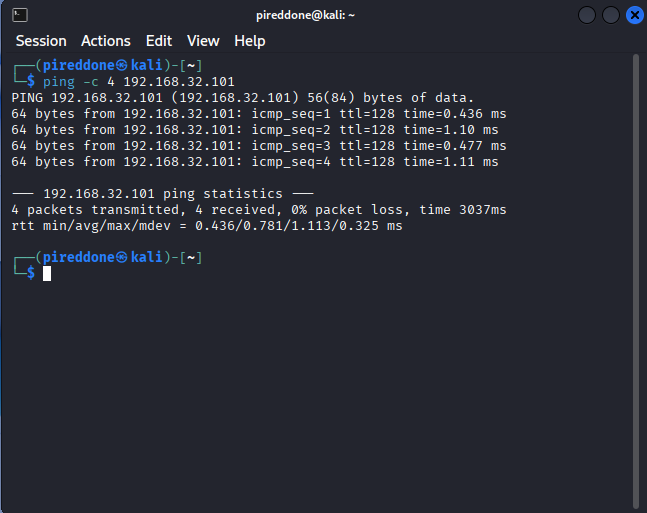
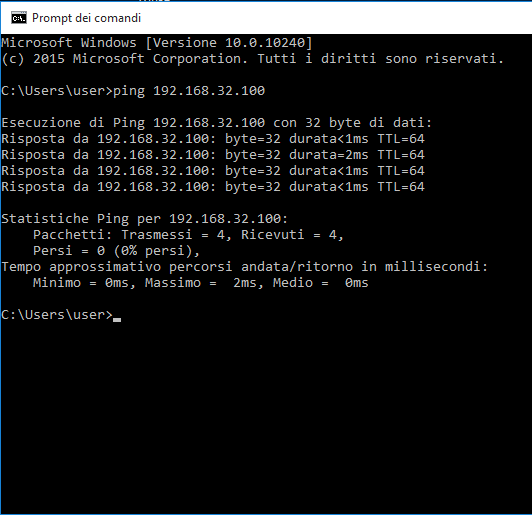
In questo progetto ci viene chiesto di simulare un’architettura client/server nella quale siano attivi sia il servizio HTTPS (in seguito solo HTTP) e il servizio DNS. Si richiede inoltre di intercettare il traffico per evidenziare la differenza tra il traffico HTTPS e HTTP.

**PROCEDIMENTO**

Si procede ad impostare gli indirizzi assegnati alle macchine virtuali:  
- Kali 192.168.32.100  
- Win10 192.168.32.101  
Ho proceduto ad assegnare l’indirizzo della MV Kali (192.168.32.100) come indirizzo DNS per entrambe le macchine.



Una volta assegnati gli indirizzi alle macchine virtuali eseguo un controllo di PING per verificare che le macchine comunichino tra loro. Nella prima immagine il ping da Win10 a Kali, nella seconda immagine il ping da Kali a Win10.



Una volta eseguiti questi controlli di comunicazione ho proceduto alla configurazione dei servizi HTTPS e DNS, incontrando qualche difficoltà nella configurazione del DNS, poiché riscontravo dei problemi con la versione 1.50 di Perl già installata.  
Questo problema di versione creava una incompatibilità tra Perl e Inetsim che non faceva funzionare il DNS. Ho eseguito una ricerca ed ho scoperto che per un utilizzo senza problemi e senza conflitti tra Perl e Inetsim avrei dovuto scaricare la versione 1.37.  
Di seguito il comando eseguito (che ho ritrovato su un forum) per l’installazione della versione 1.37:  
***curl -O https://cpan.metacpan.org/authors/id/N/NL/NLNETLABS/Net-DNS-1.37.tar.gz***

***tar xzf Net-DNS-1.37.tar.gz***

***cd Net-DNS-1.37***

***perl Makefile.PL***

***make***

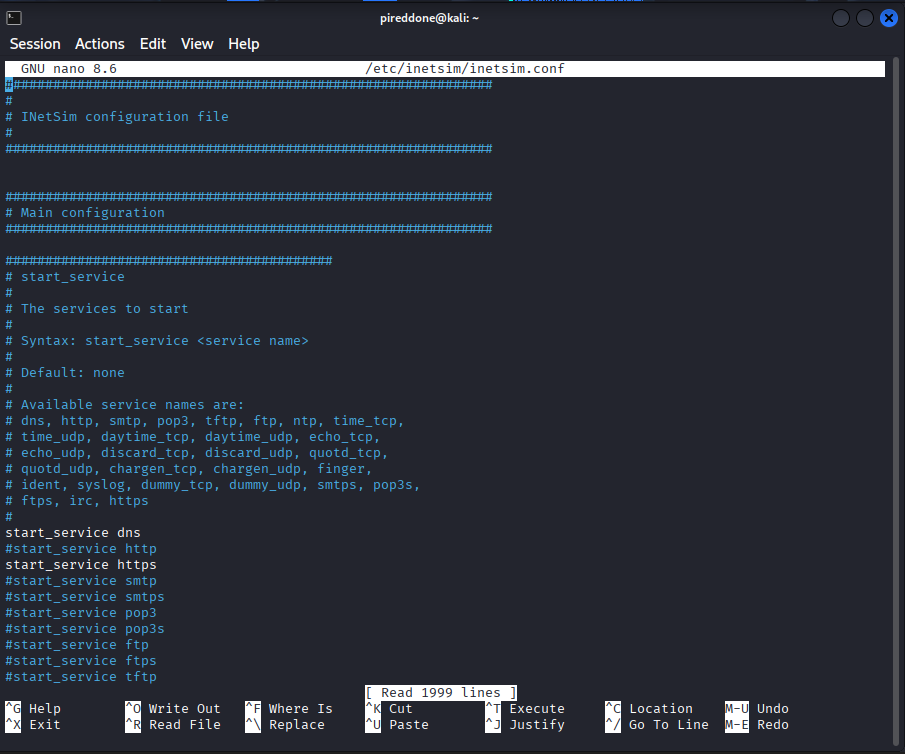
***make test***

***sudo make install***

Una volta eseguita questa modifica, i servizi hanno funzionato a dovere, di seguito illustro le fasi della configurazione dei servizi.

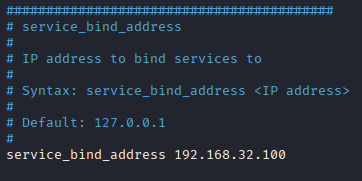
**Abilitazione di HTTPS e DNS***(accediamo al file di configurazione con sudo nano /etc/inetsim/inetsim.conf)*

In questa schermata andiamo ad eliminare il “#” che va a commentare le righe **start\_service dns** e **start\_service https** in modo che queste vengano lette ed eseguite.

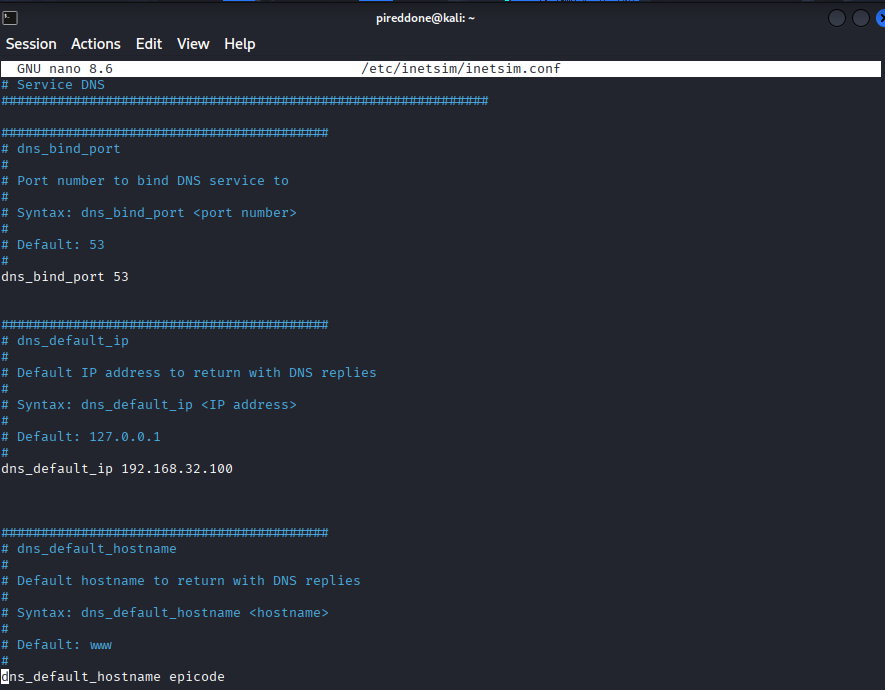


**Abilitazione e modifica dei parametri DNS**

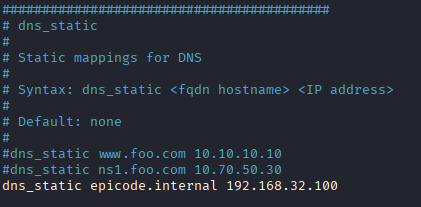
In service bind address ho impostato come IP di ascolto l’indirizzo della macchina Kali, poiché altrimenti Inetsim avrebbe impostato di default 127.0.0.1 rendendo impossibile lo svolgimento del progetto.

****

In questa schermata ho proceduto ad abilitare la porta 53 in ascolto, ho impostato come indirizzo di default del DNS l’indirizzo 192.168.32.100 ed ho assegnato il DNS Hostname “epicode”.

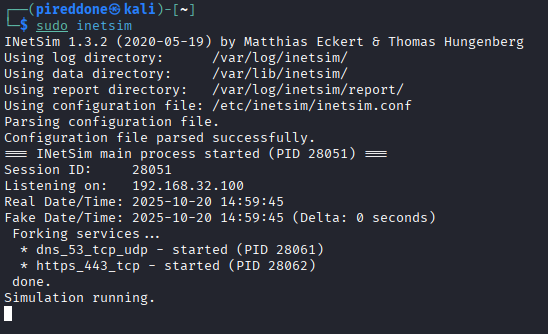
****

Ho proceduto in seguito ad associare l’indirizzo 192.168.32.100 a “epicode.internal” utilizzando la funzione “dns\_static” di inetsim.

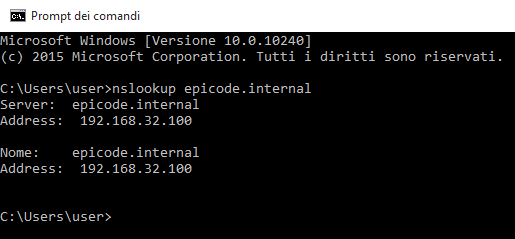
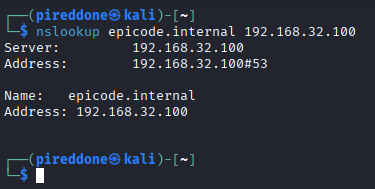


Nella parte finale della configurazione ho proceduto ad abilitare (rimuovendo il #) la porta 80 per il servizio HTTP e la porta 443 per il servizio HTTPS.

**Avvio del servizio INETSIM**

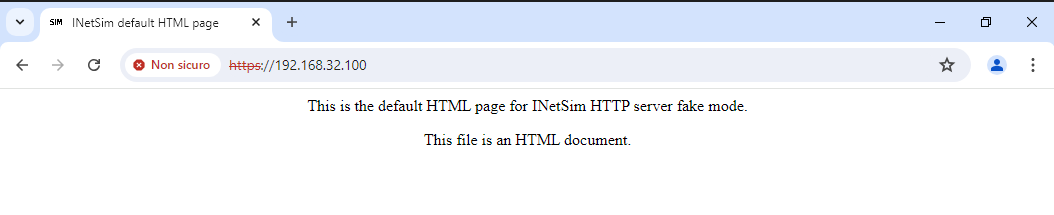
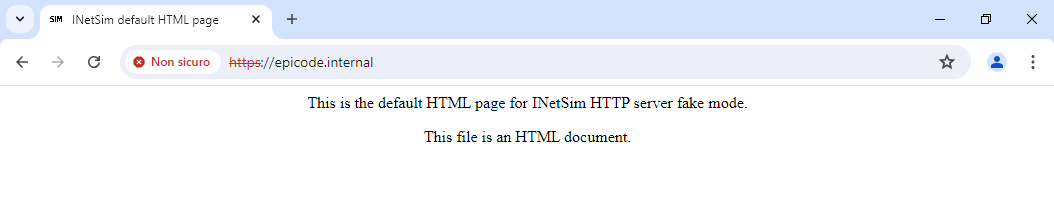
****

**Verifica del funzionamento - nslookup**Per verificare il corretto funzionamento del servizio DNS ho proceduto con un nslookup sia su Win10 che su Kali, con esito positivo.

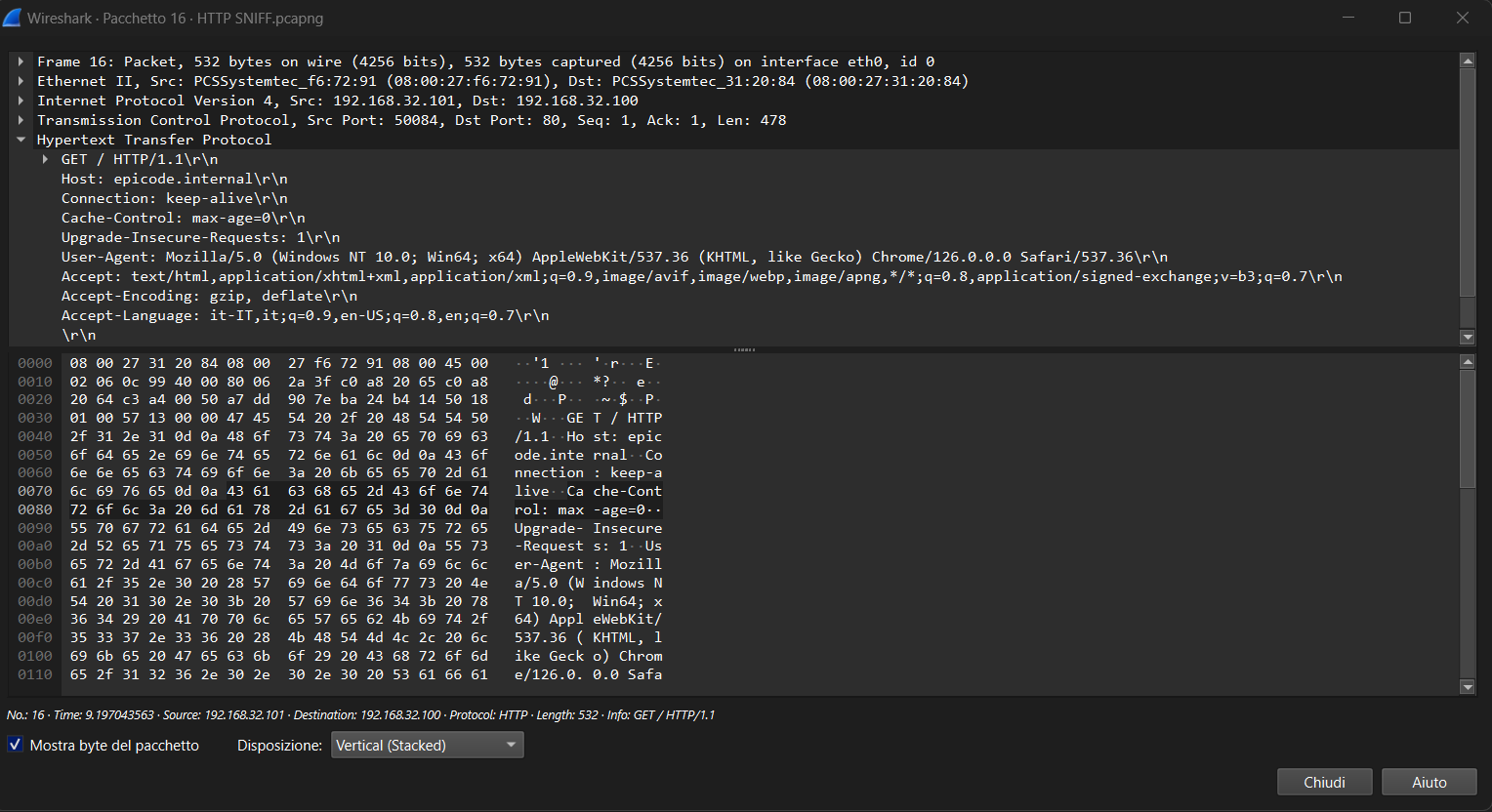
**Verifica del funzionamento – Chrome**

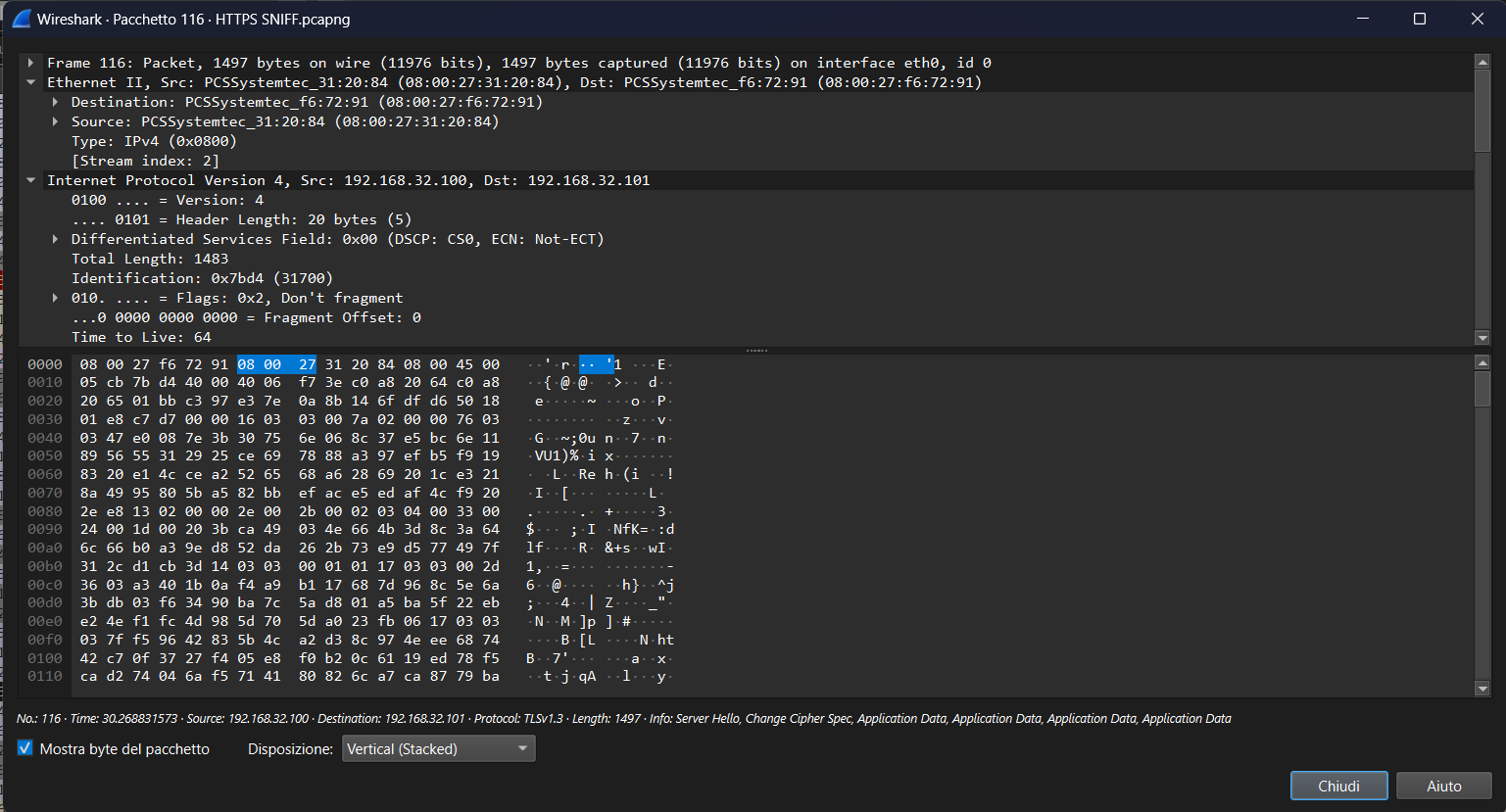
Per verificare che da Win10 riusciamo a raggiungere il servizio HTTPS attivo, ho provato ad utilizzare Chrome per raggiungere sia il 192.168.32.100 che “epicode.internal”, in entrambe i casi con esito positivo, raggiungendo questa schermata di seguito.

**Analisi di Wireshark e conclusione**

Nella schermata di Wireshark, con il servizio **HTTP** attivo su Inetsim, tutto rimane in chiaro, poiché si nota il metodo (GET), l’URL richiesto, l’header e il contenuto.



Nella schermata di Wireshark, con il servizio **HTTPS** attivo su Inetsim invece tutto è cifrato con TLS, il che dimostra l’efficacia del servizio. 

In entrambe i casi, gli indirizzi MAC invece restano visibili, essendo al livello 2 della pila ISO/OSI.