

PROGETTO FINALE W4D4 EPICODE

INTRODUZIONE ALL' HACKING

PACE MASSIMILIANO 20 LUGLIO 2025

° PUNTO 1

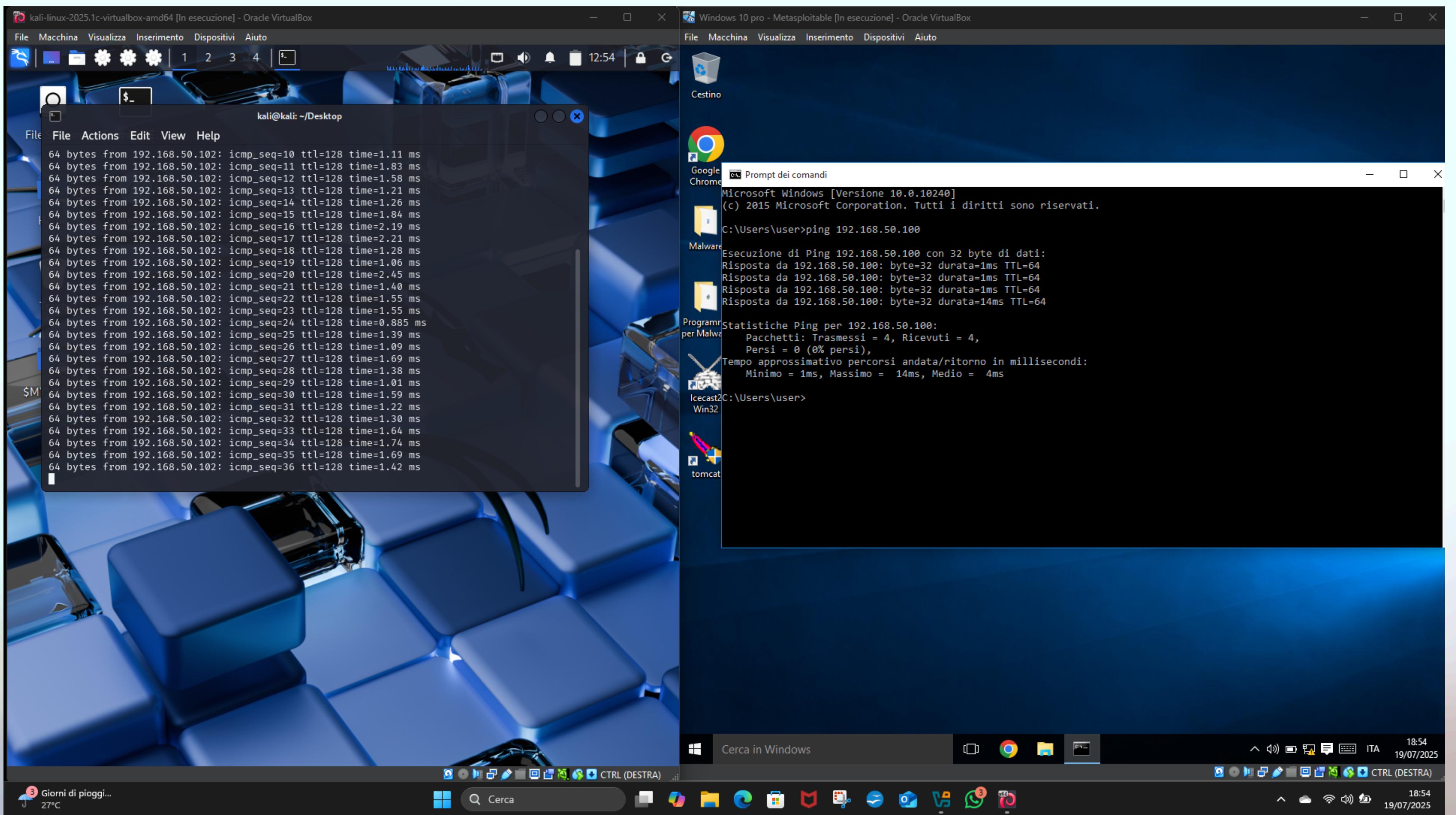
SIMULAZIONE IN AMBIENTE DI LABORATORIO VIRTUALE UN ARCHITETTURA CLIENT SERVER IN CUI UN CLIENT TRAMITE WEB BROWSER RICHIENDE UNA RISORSA ALL'HOSTNAME **EPICODE.INTERNAL**

° PUNTO 2

SI INTERCETTI LA COMUNICAZIONE TRAMITE WIRESHARK EVIDENZIANDO I MAC ADDRESS DI SORGENTE E DESTINAZIONE ED IL CONTENUTO DELL RICHIESTA HTTPS

° PUNTO 3

ESEGUIRE NUOVAMENTE L'ESERCIZIO, SOSTITUENDO IL SERVER HTTPS CON UN SERVER HTTP. INTERCETTARE NUOVAMENTE IL TRAFFICO, EVIDENZIANDO EVENTUALI DIFFERENZE TRA IL TRAFFICO APPENA CATTURATO IN HTTP E QUELLO PRECEDENTE IN HTTPS. SPIEGARE E MOTIVARE LE PRINCIPALI DIFFERENZE, SE PRESENTI



Fase_iniziale :

Avvio di Kali linux e di Windows 10
Individuazione ip Kali

192.168.50.100

Individuazione ip windows

192.168.50.102

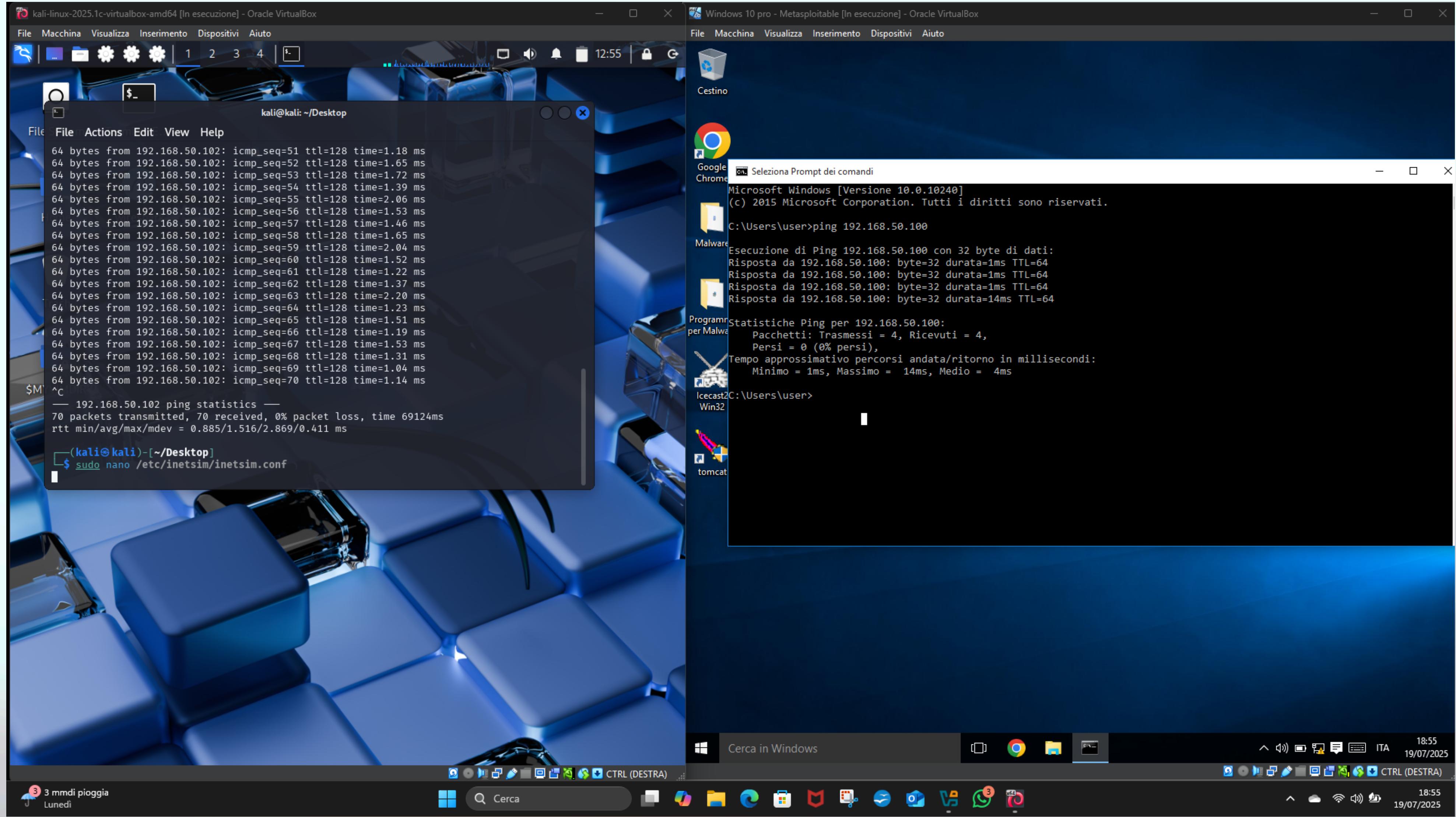
Avvio ping da Kali a windows

Eseguito

Avvio ping da Windows a Kali

Eseguito

Fase 2:
Preparazione e
configurazione di Inetsim
Inetsim e' un software che
simula servizi internet
come http , https , dns in
un ambiente di
laboratorio.
Per configurare Inetsim
utilizzeremo il comando
Sudo nano /etc/
inetsim.conf





Fase 3:

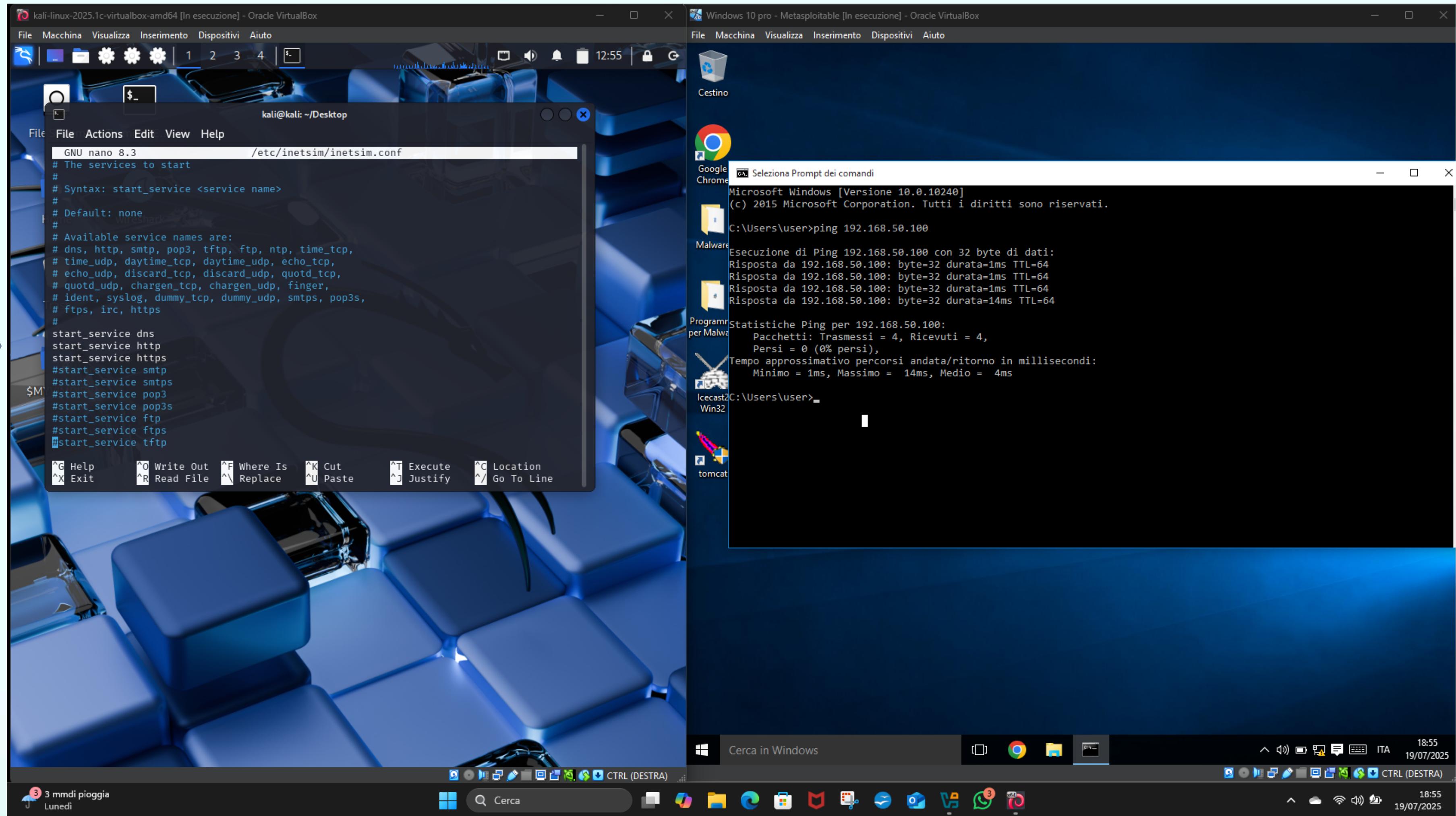
La configurazione di Inetsim
prevede l'attivazione del dns

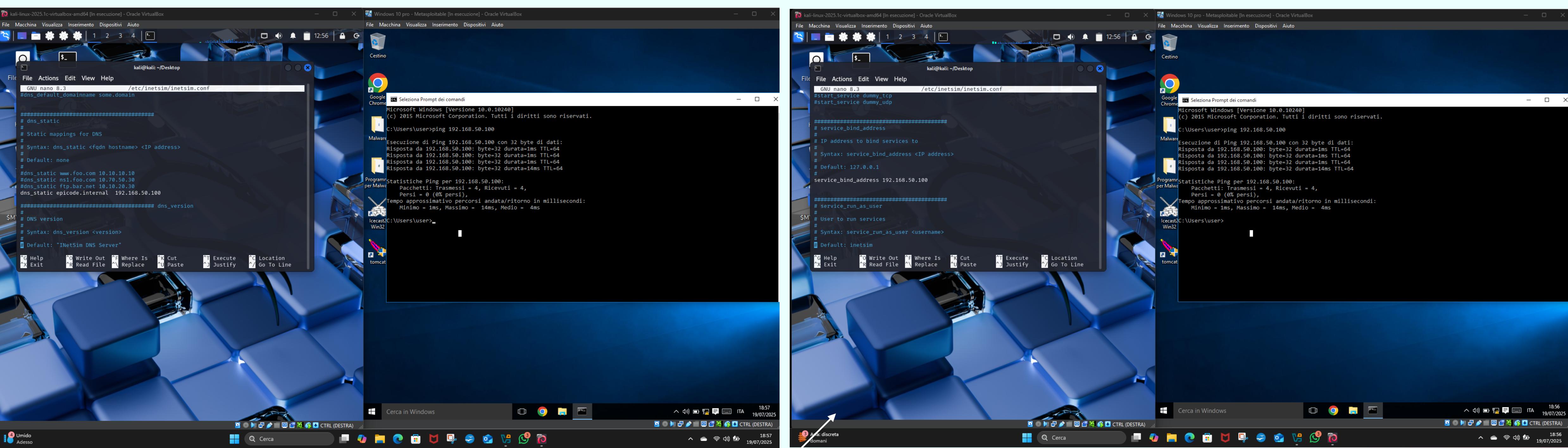
http e https

decommentandoli

Diciamo che i servizi
risultano "dormienti"

Per attivarli basta togliere il
cancelletto





Fase 5:

Abilitiamo il Dns inserendo l' hostname
In questo caso Epicode.internal più l'indirizzo ip
192.168.50.100

Fase 4:

Come precedentemente spiegato bisognerà attivare
anche il service bind associando l'indirizzo ip
192.168.50.100

Fase 6:
Dopo aver
configurato
Inetsim e abilitato
il dns , faremo
partire Inetsim
tramite il
comando
sudo Inetsim
Interessante
notare come già
all'avvio di Inetsim
ci indichi le porte
http e https

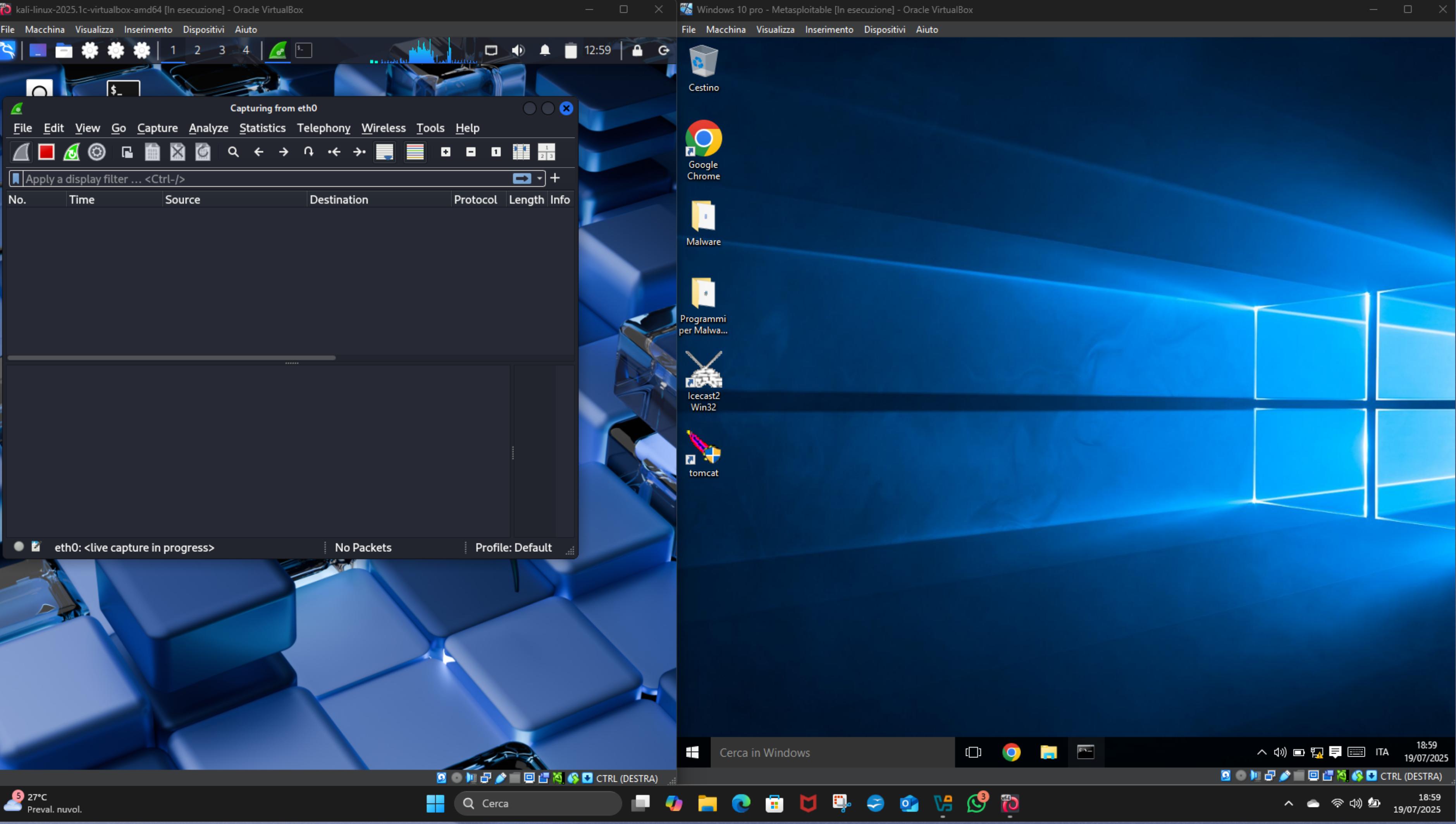
kali@kali: ~/Desktop

```
$ sudo nano /etc/inetsim/inetsim.conf
[sudo] password for kali:
(kali㉿kali)-[~/Desktop]
$ sudo inetsim
INetSim 1.3.2 (2020-05-19) by Matthias Eckert & Thomas Hungenberg
Using log directory: /var/log/inetsim/
Using data directory: /var/lib/inetsim/
Using report directory: /var/log/inetsim/report/
Using configuration file: /etc/inetsim/inetsim.conf
Parsing configuration file.
Configuration file parsed successfully.
== INetSim main process started (PID 4036) ==
Session ID: 4036
Listening on: 192.168.50.100
Real Date/Time: 2025-07-19 12:57:59
Fake Date/Time: 2025-07-19 12:57:59 (Delta: 0 seconds)
  Forking services ...
    * dns_53_tcp_udp - started (PID 4046)
  deprecated method; prefer start_server() at /usr/share/perl5/INetSim/DNS.pm line 69.
  Attempt to start Net::DNS::Nameserver in a subprocess at /usr/share/perl5/INetSim/DNS.pm line 69.
    * http_80_tcp - started (PID 4047)
    * https_443_tcp - started (PID 4048)
done.
Simulation running.
```

C:\Users\user>ping 192.168.50.100

```
Esecuzione di Ping 192.168.50.100 con 32 byte di dati:
Risposta da 192.168.50.100: byte=32 durata=1ms TTL=64
Risposta da 192.168.50.100: byte=32 durata=1ms TTL=64
Risposta da 192.168.50.100: byte=32 durata=1ms TTL=64
Risposta da 192.168.50.100: byte=32 durata=14ms TTL=64

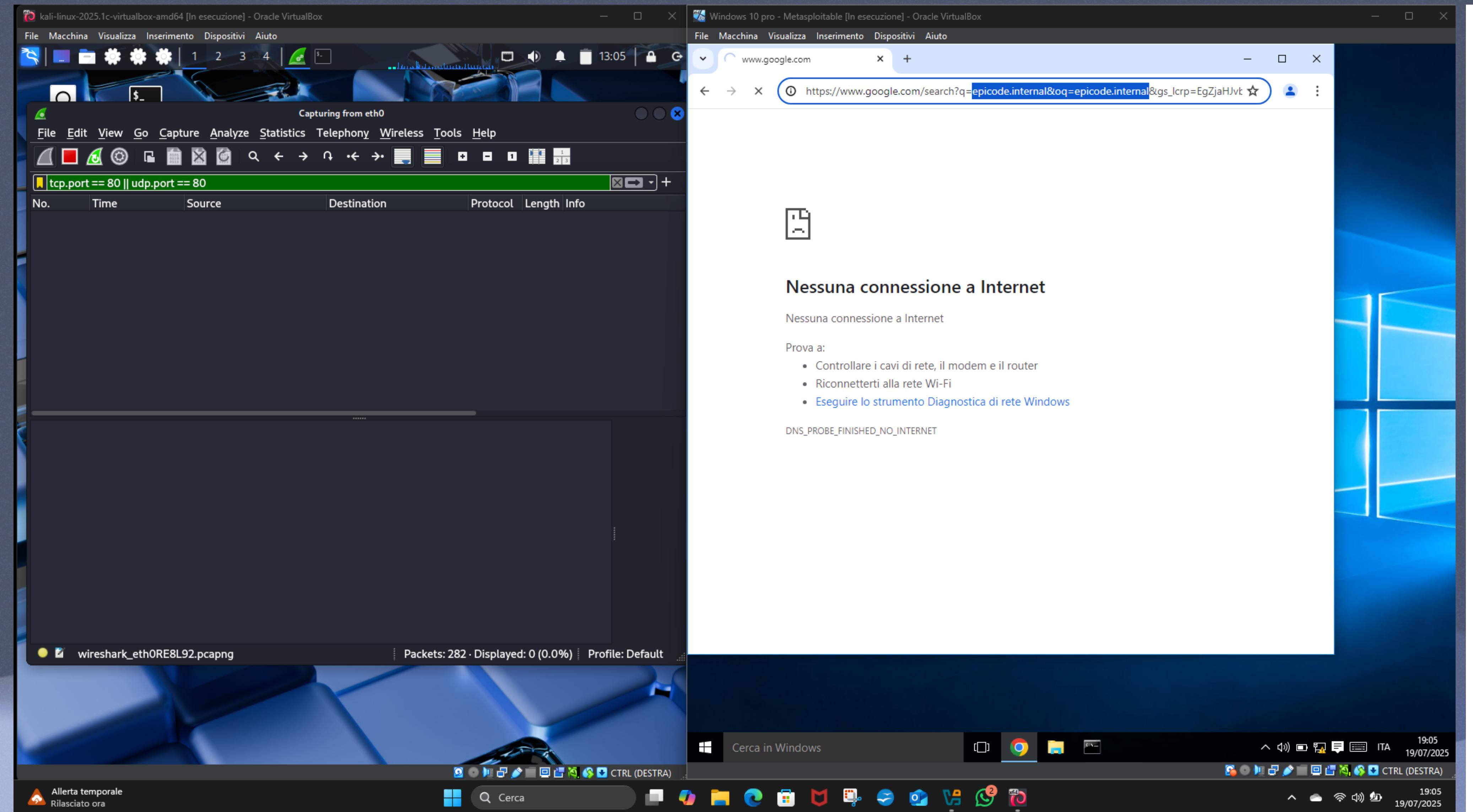
Statistiche Ping per 192.168.50.100:
  pacchetti: Trasmessi = 4, Ricevuti = 4,
  Persi = 0 (0% persi),
  Tempo approssimativo percorsi andata/ritorno in millisecondi:
    Minimo = 1ms, Massimo = 14ms, Medio = 4ms
```



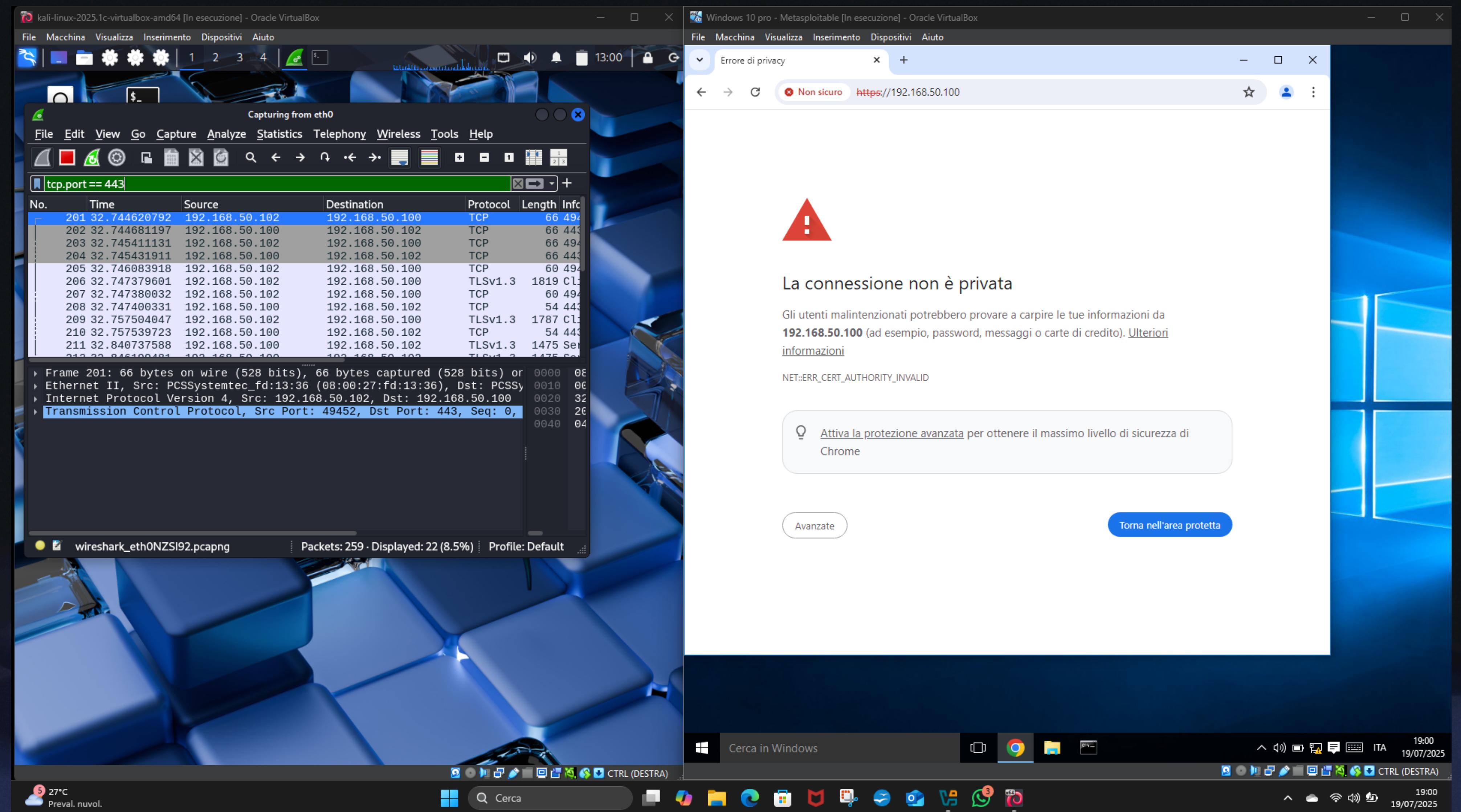
Fase 7: Avvio di Wireshark .

Questo software e' il cuore del nostro test, a lui viene dato il compito di analizzare il traffico di rete
E' un classico sniffer un analizzatore di pacchetti

Fase 8:
Avvio di Chrome e
apertura pagina
Epicode.internal
Purtroppo ho
riscontrato
l'impossibilità di poter
far funzionare il dns di
Inetsim
Sembra esserci un
qualche problema di
incompatibilità. Non
sono riuscito a trovare
molto a riguardo.
Quindi ci limiteremo ad
utilizzare direttamente il
nostro ip nella barra di
ricerca evitando di
epicode.internal



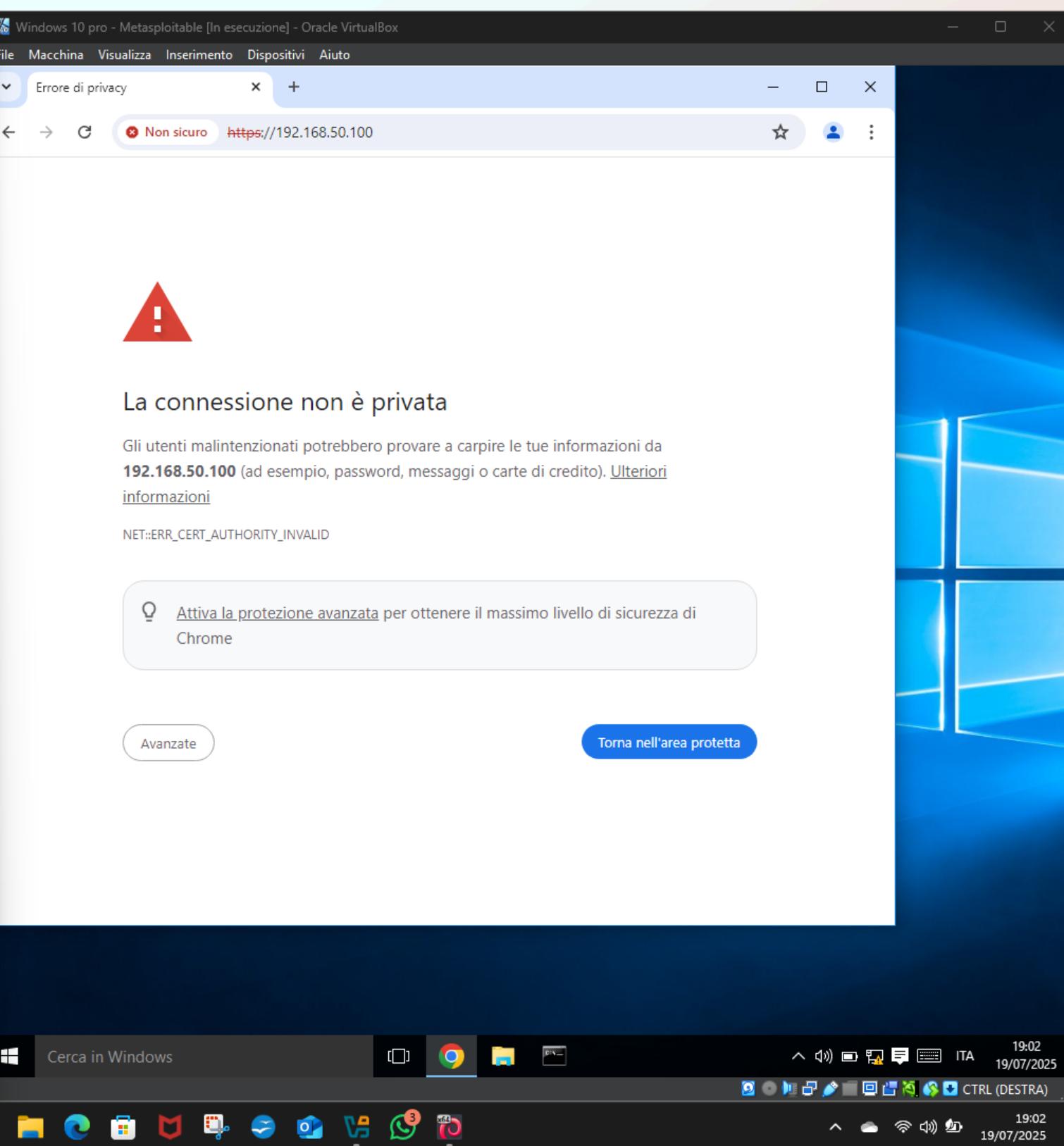
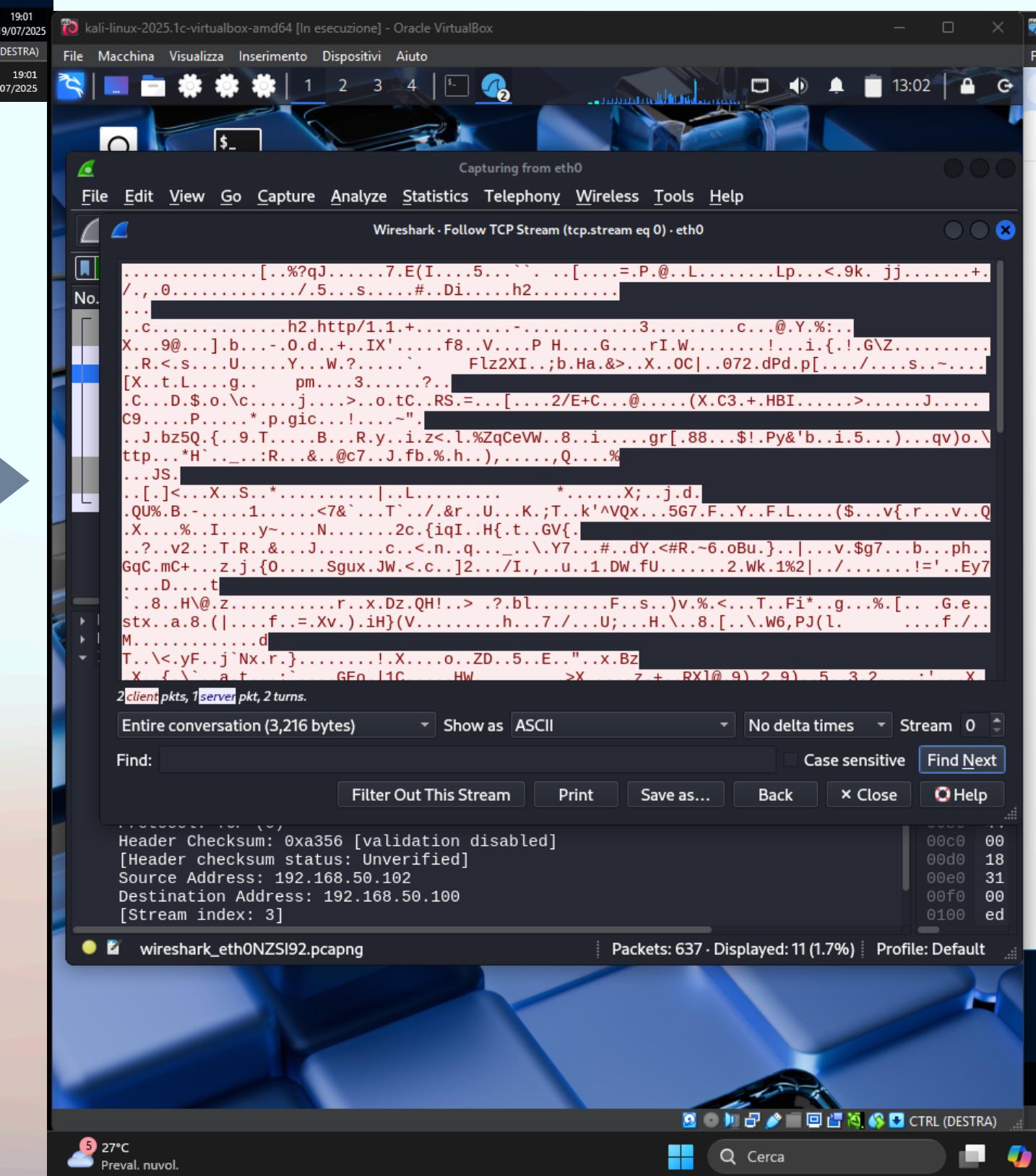
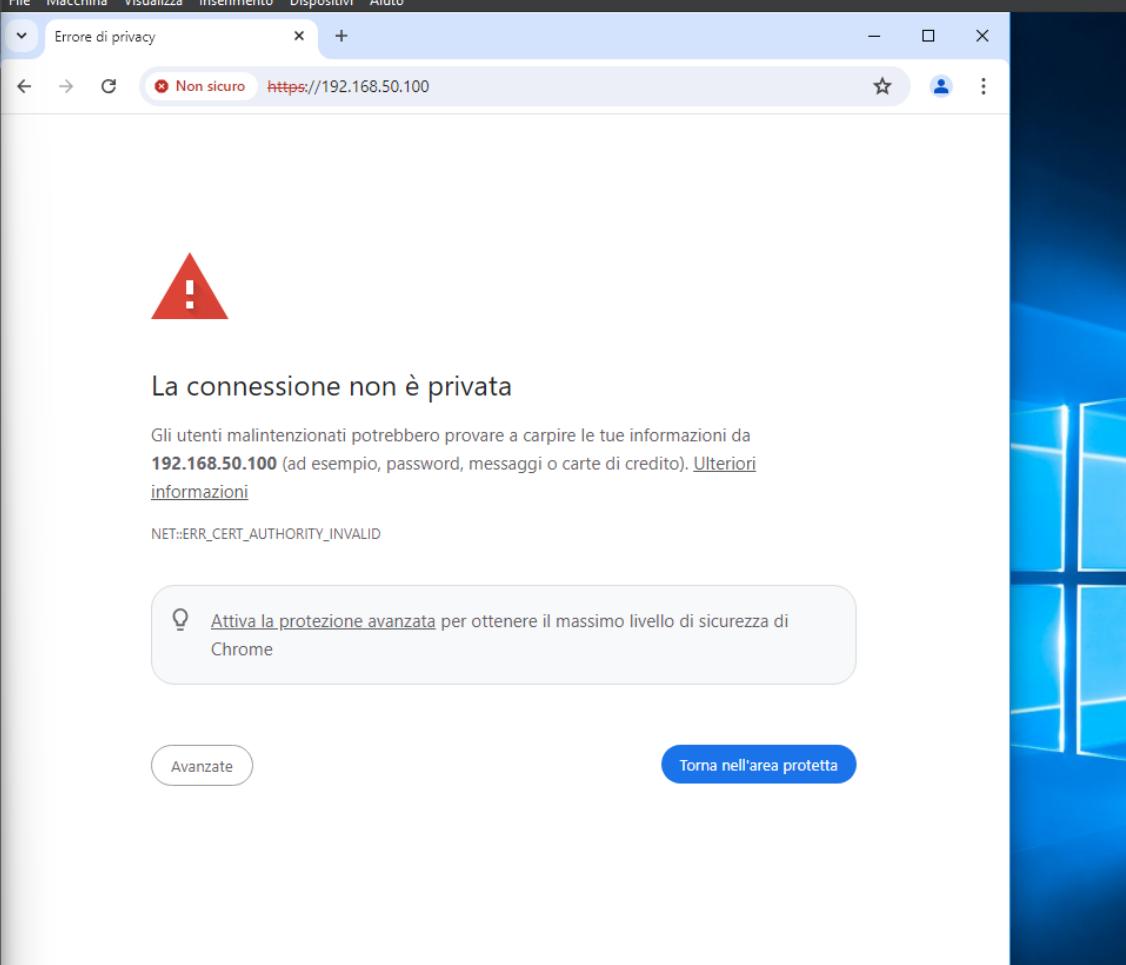
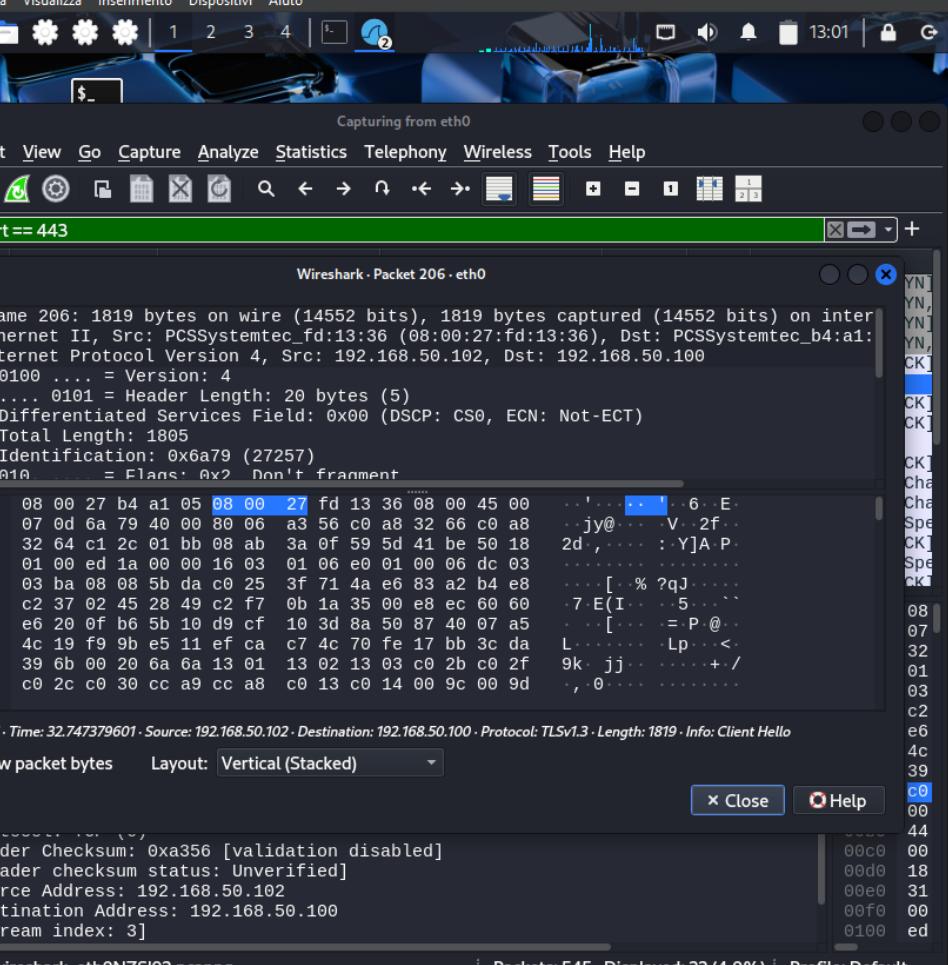
Fase 9:
Procediamo con
l'inserimento di
<https://192.168.50.100>
Per visualizzare il
traffico che ci interessa
Selezioniamo la porta
tcp.port ==443
Https essendo un
protocollo di
comunicazione sicura
in Wireshark
appariranno pacchetti
che se andati a
visionare risulteranno
protetti da crittografia



Fase 10: Analizziamo i pacchetti ricevuti

Cio' che si potra' notare nel seguire pacchetti specifici e' che tutti i dati sono a primo impatto incomprensibili

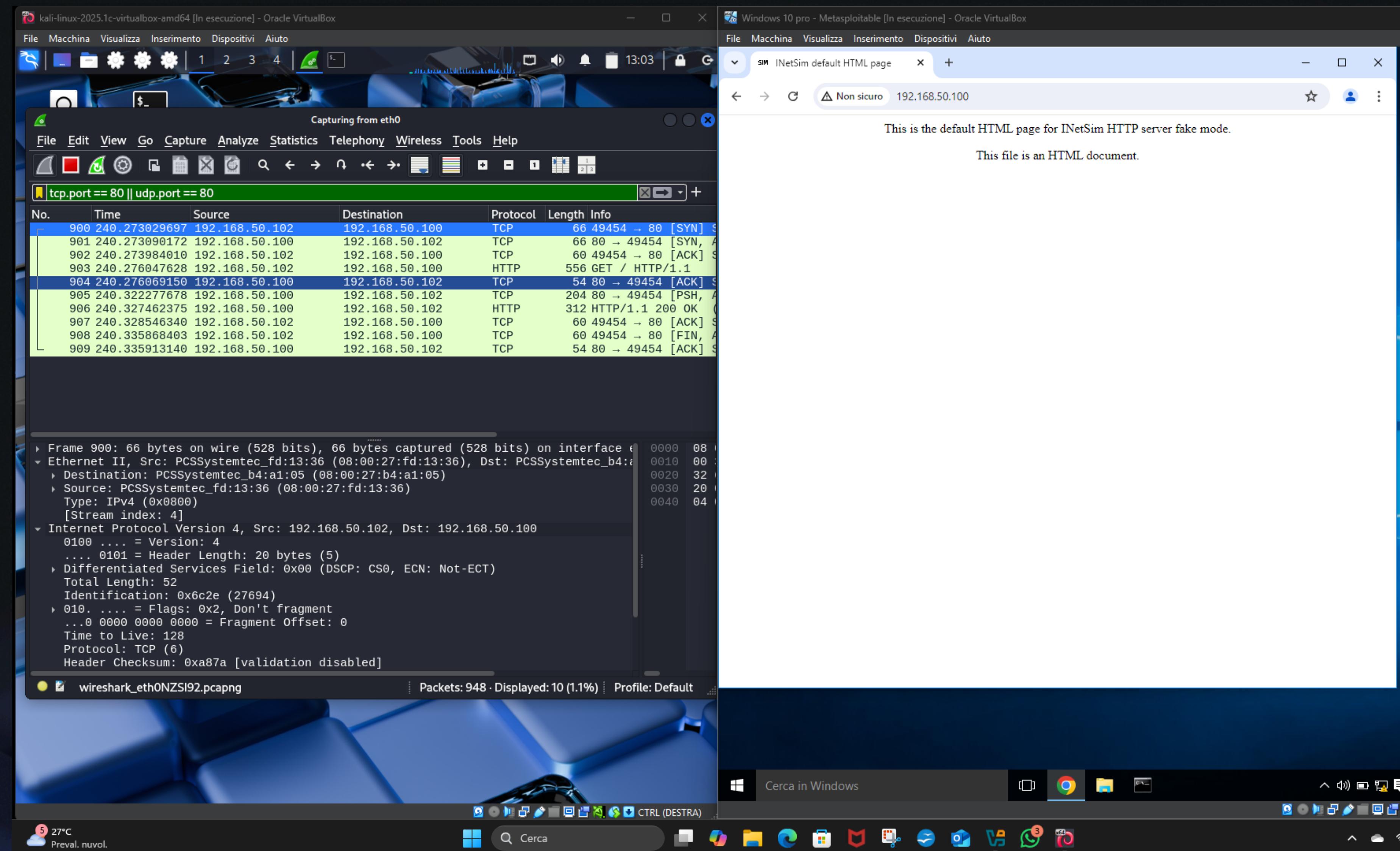
Specialmente se analizzati in parallelo ai pacchetti http

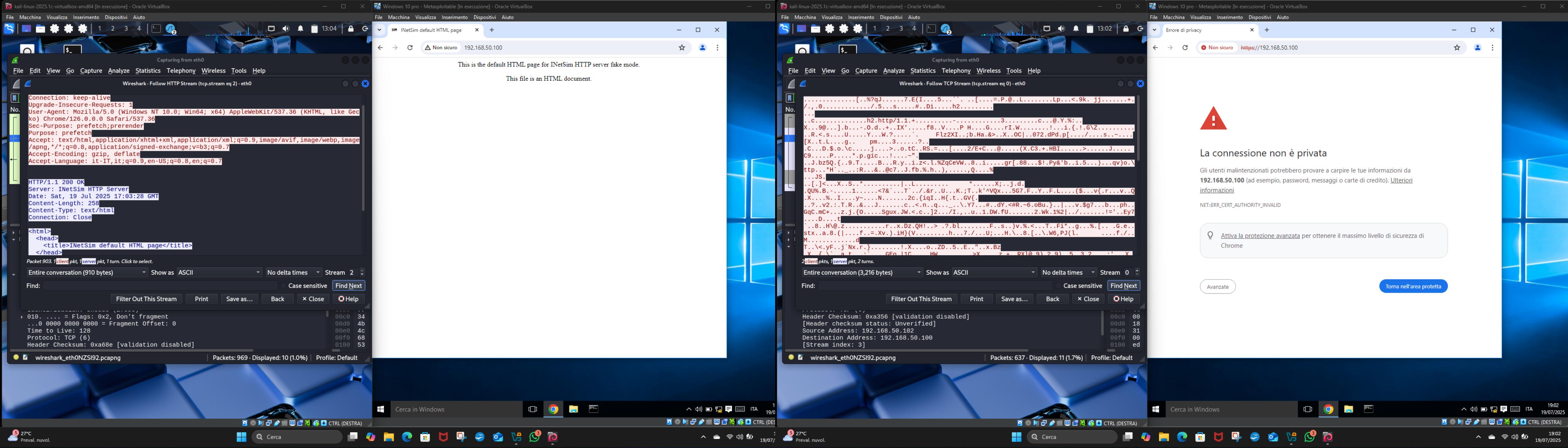


Fase 11

Analizzando invece
http://192.168.50.100

Noteremo che a
differenza del
precedente https
Qui nella colonna
protocolli apparira' http e
non tls





Fase 12:

Qui sopra vengono messe a confronto le due schermate come

paragone di HTTP e HTTPS

Cio' che subito si puo' notare e' come lavorano i due protocolli .

In Http tutto e' in chiaro .

Grazie a Wireshark saremo in grado di visualizzare pacchetti http come richieste , risposte , payload , indirizzo ip mittente destinatario ma solo con protocollo HTTP

CONCLUSIONI

WIRESHARK E' UN SOFTWARE MOLTO INTERESSANTE , MA ANCHE MOLTO LIMITATO.
AD OGGI HTTP VIENE ANCORA UTILIZZATO MA PER LA TRASMISSIONE DI DATI NON SENSIBILI PER ESEMPIO
RETI INTERNE DOVE LA SICUREZZA NON E' UNA PRIORITA'.
HTTPS ORAMAI E' DIVENTATO LO STANDARD PER LA MAGGIOR PARTE DEI SITI INTERNET E FORTEMENTE
RACCOMANDATO , TANTO CHE GOOGLE E ALTRI MOTORI DI RICERCA CONSIDERANO L'UTILIZZO DEL
PROTOCOLLO UN FATTORE DI RANKING.
L'ESERCIZIO MIRA A FARTI VEDERE QUALI SONO LE DIFFERENZE FRA 2 PROTOCOLLI E COME SIA SEMPLICE
CARPIRE INFORMAZIONI (PACCHETTI)DA UN PROTOCOLLO NON SICURO