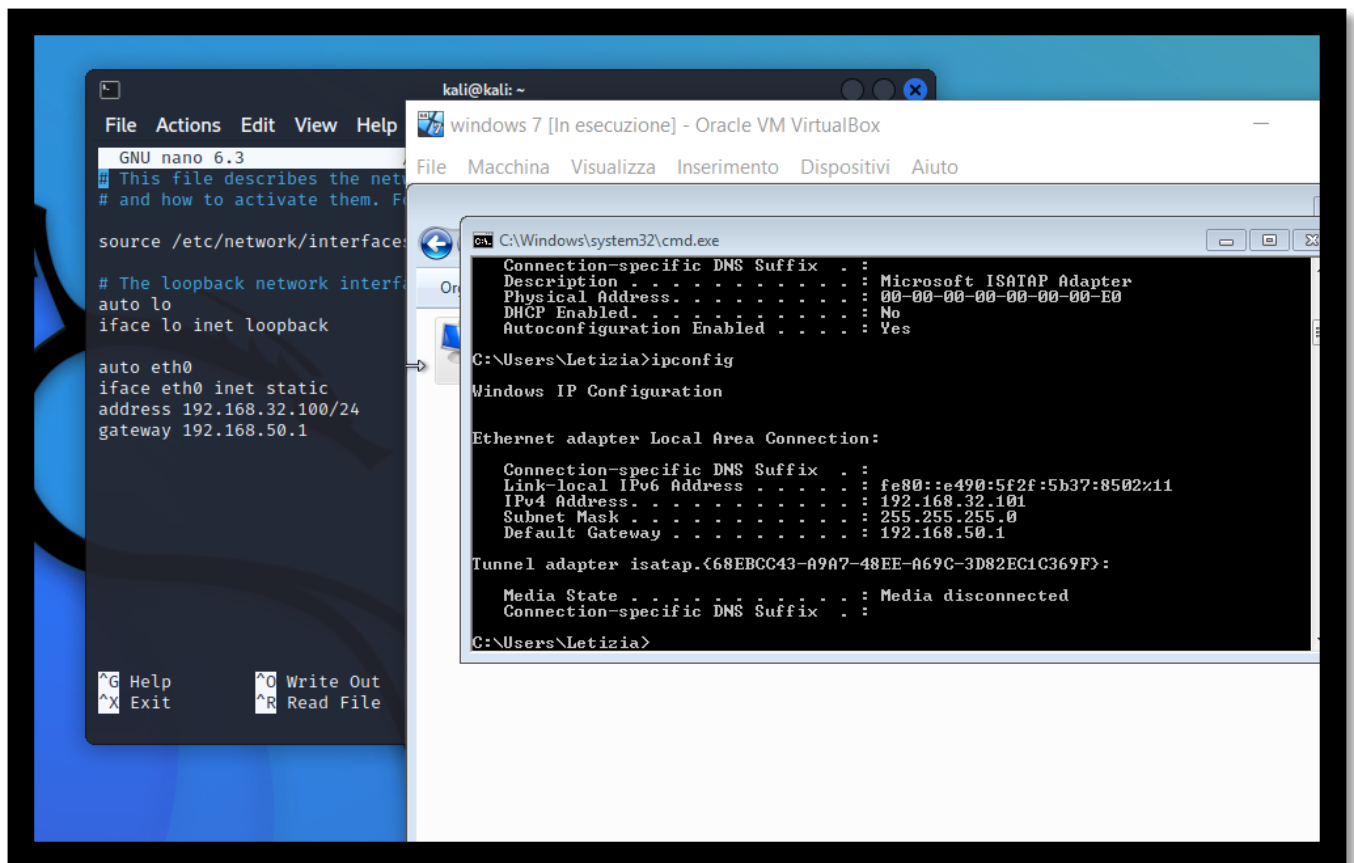


Report simulazione rete complessa

Per l'esercizio richiesto, iniziamo impostando in Kali e Windows 7 gli IP richiesti nella traccia:

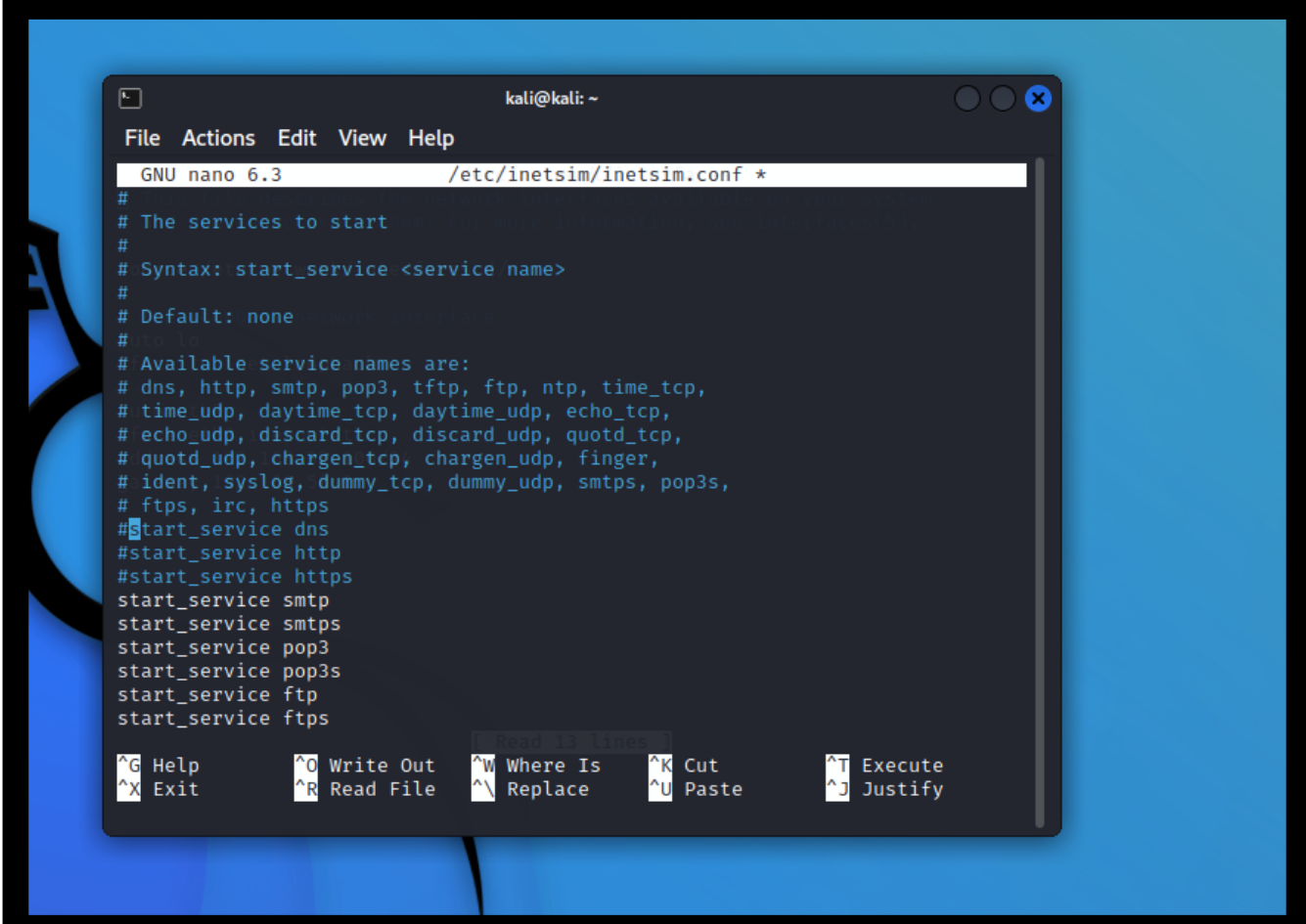
Kali Linux : IP 192.168.32.100

Windows : IP 192.168.32.101



(Nota: la schermata windows 7 non è quella del percorso per il cambio ip (internet protocol version 4 -TCP/IPv4- properties)ma solo a scopo illustrativo)

Fatto ciò, passeremo all'impostazione di inetSim, abilitando anche DNS service, http service e https service

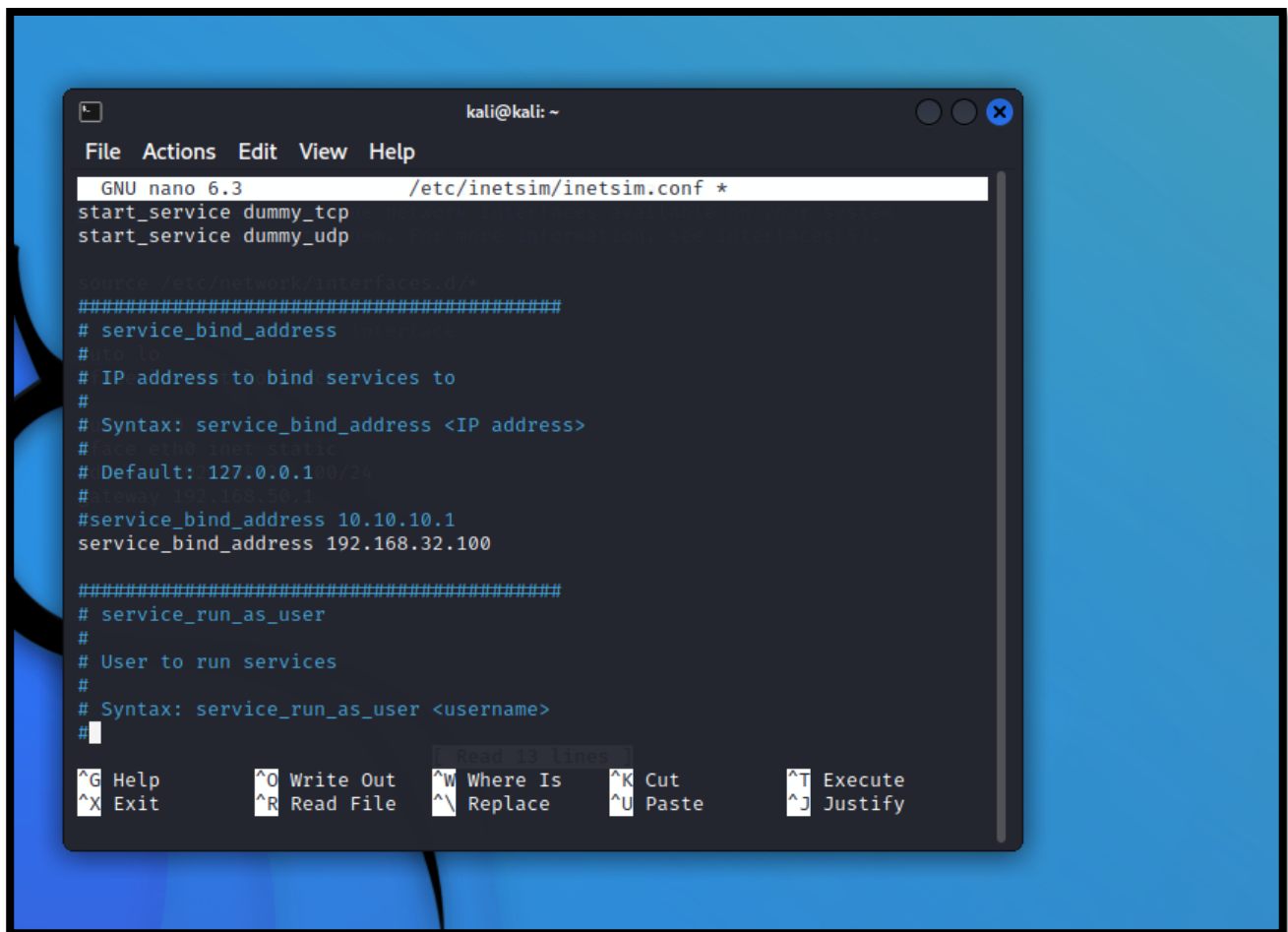


The image shows a terminal window titled 'kali@kali: ~' with a menu bar (File, Actions, Edit, View, Help). The window displays the nano 6.3 editor editing the file '/etc/inetSim/inetSim.conf'. The content of the file is as follows:

```
#
# The services to start
#
# Syntax: start_service <service name>
#
# Default: none
#
# Available service names are:
# dns, http, smtp, pop3, tftp, ftp, ntp, time_tcp,
# time_udp, daytime_tcp, daytime_udp, echo_tcp,
# echo_udp, discard_tcp, discard_udp, quotd_tcp,
# quotd_udp, chargen_tcp, chargen_udp, finger,
# ident, syslog, dummy_tcp, dummy_udp, smtps, pop3s,
# ftps, irc, https
#start_service dns
#start_service http
#start_service https
start_service smtp
start_service smtps
start_service pop3
start_service pop3s
start_service ftp
start_service ftps
```

At the bottom of the terminal, there is a status bar with the following keyboard shortcuts: ^G Help, ^O Write Out, ^W Where Is, ^K Cut, ^T Execute, ^X Exit, ^R Read File, ^\ Replace, ^U Paste, and ^J Justify. A message 'Read 13 lines' is also visible above the shortcuts.

Tramite la configurazione del parametro Bind address, andremo a legare l'indirizzo ip del dns a quello di Kali.



```
kali@kali: ~  
File Actions Edit View Help  
GNU nano 6.3 /etc/inetsim/inetsim.conf *  
start_service dummy_tcp  
start_service dummy_udp  
  
source /etc/network/interfaces.d/*  
#####  
# service_bind_address  
# IP address to bind services to  
#  
# Syntax: service_bind_address <IP address>  
# Default: 127.0.0.1  
# service_bind_address 10.10.10.1  
service_bind_address 192.168.32.100  
  
#####  
# service_run_as_user  
#  
# User to run services  
#  
# Syntax: service_run_as_user <username>  
#
```

Impostiamo quindi come Domain name system <<epicode.internal>>

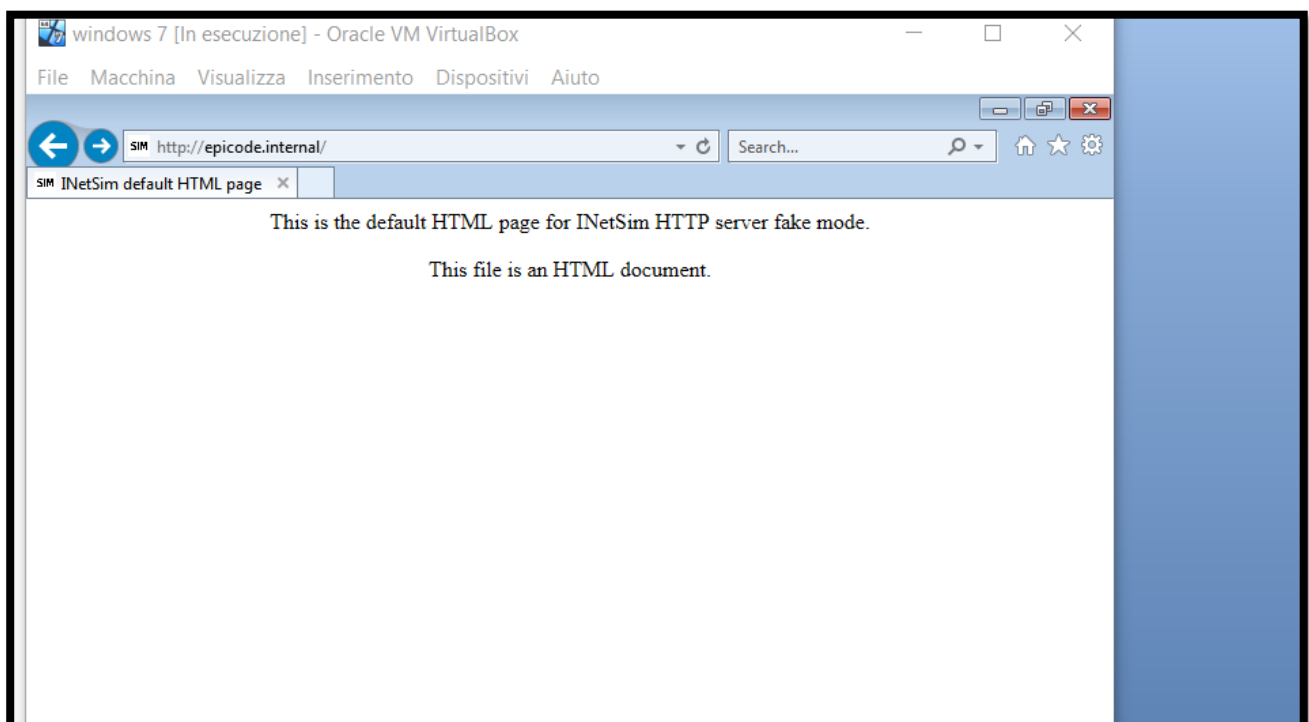
```

kali@kali: ~
File Actions Edit View Help
GNU nano 6.3 /etc/inetsim/inetsim.conf *
#
# Default: none
#
#dns_static www.foo.com 10.10.10.10
#dns_static ns1.foo.com 10.70.50.30
#dns_static ftp.bar.net 10.10.20.30
#dns_static epicode.internal 192.168.32.100
#face ip inet loopback
#####
# dns_version
#face eth0 inet static
# DNS version 192.168.32.100/24
# gateway 192.168.50.1
# Syntax: dns_version <version>
#
# Default: "INetSim DNS Server"
#
#dns_version "9.2.4"

#####
# Service HTTP
#####
Read 13 lines
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute
^X Exit      ^R Read File ^_ Replace   ^U Paste     ^J Justify

```

Verificando l'avvenuta connessione otterremo una pagina html in server fake mode grazie ad inetSIM



Una volta aperto il tool Wireshark saremo in grado di catturare i pacchetti intercorsi tramite http, in chiaro, come nel precedente esercizio.

Ma come potremo notare dal passaggio da HTTP a HTTPS, nel secondo caso i dati non saranno più trasmessi tramite canale non protetto (ed in chiaro), bensì criptati tramite chiavi simmetriche in TLS handshake.

