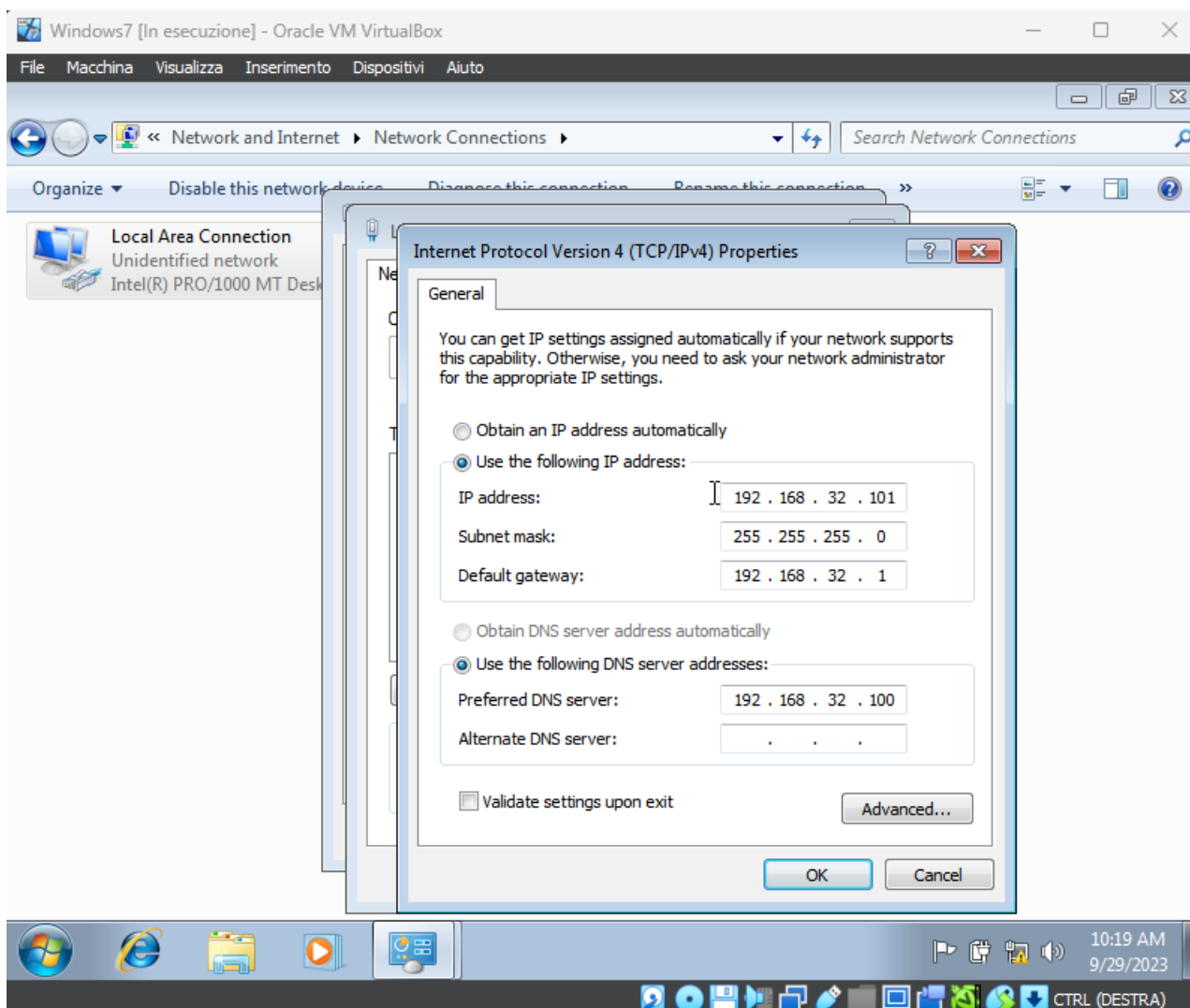
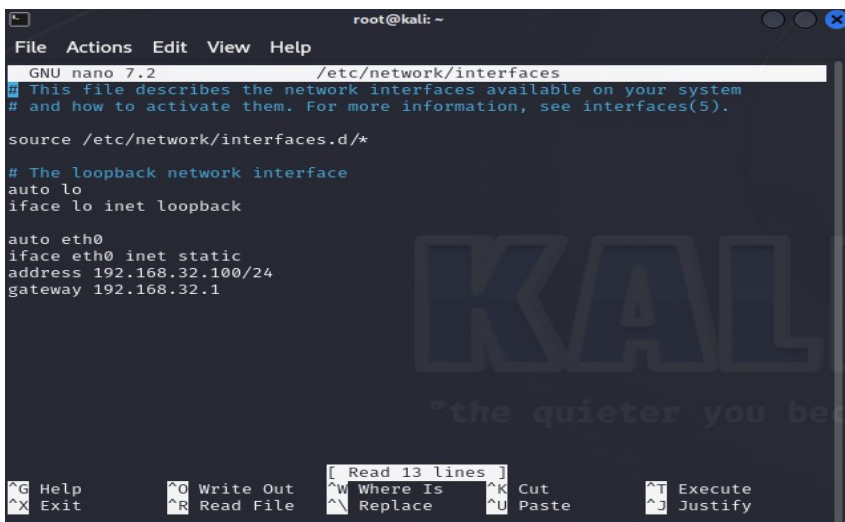


Progetto 29/09 - Luca Danelli

Imposto l'ip sulla macchina windows 7 configurando come server DNS l'ip che andrò ad assegnare alla macchina kali:



Imposto l'ip sulla macchina Kali con il comando **sudo nano /etc/network/interfaces**, andando quindi a modificare il file di configurazione di rete con l'editor nano, successivamente riavvio la macchina con il comando **sudo reboot** per applicare la configurazione:

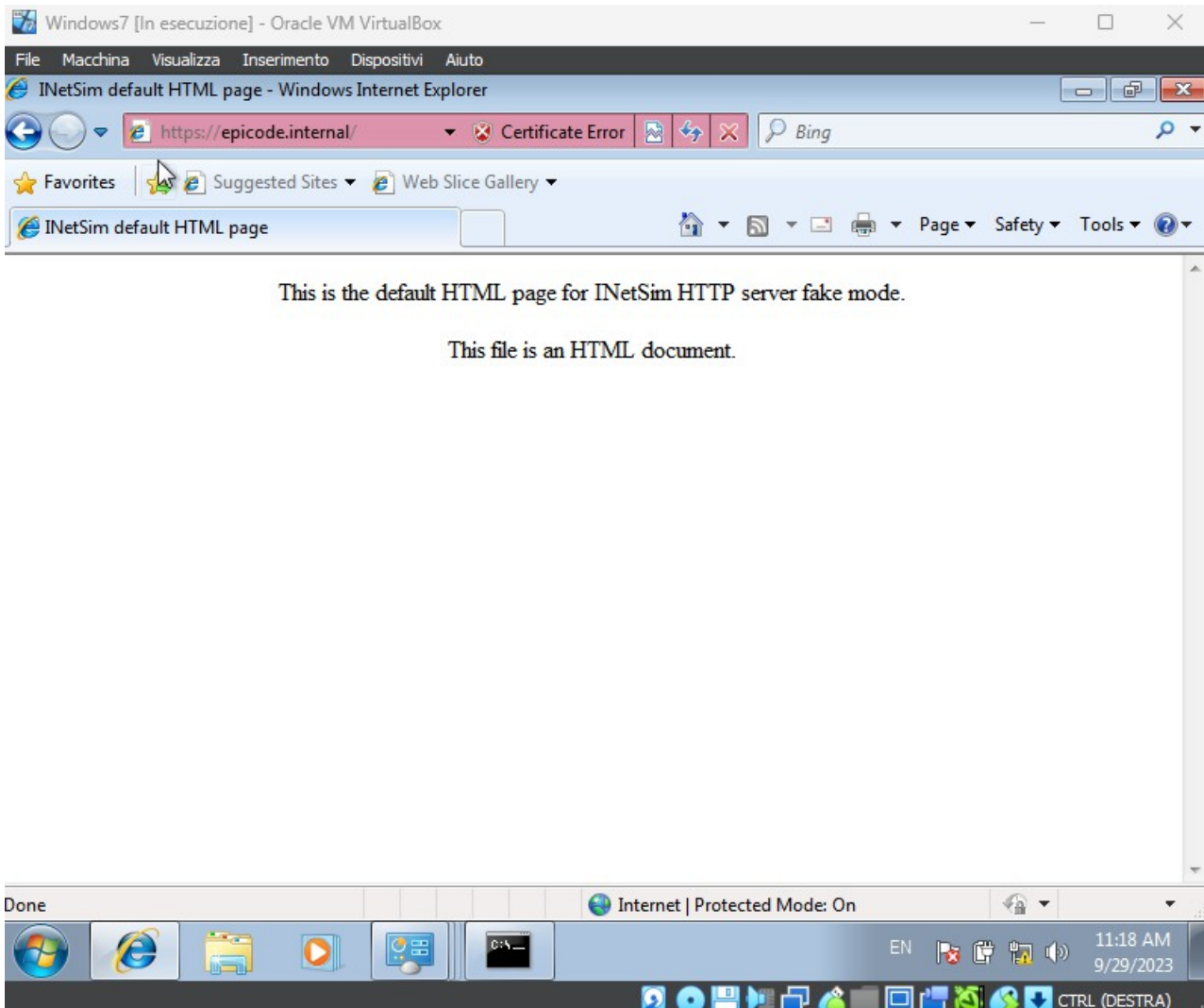


```
root@kali: ~  
File Actions Edit View Help  
(root@kali)-[~]  
# ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.32.100 netmask 255.255.255.0 broadcast 192.168.32.255  
    inet6 fe80::a00:27ff:feeb:7ef5 prefixlen 64 scopeid 0x20<link>  
    inet6 2001:b07:aac:7465:a00:27ff:feeb:7ef5 prefixlen 64 scopeid 0x0  
<global>  
    ether 08:00:27:cb:7e:f5 txqueuelen 1000 (Ethernet)  
    RX packets 67 bytes 14341 (14.0 KiB)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 22 bytes 8379 (8.1 KiB)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536  
    inet 127.0.0.1 netmask 255.0.0.0  
    inet6 ::1 prefixlen 128 scopeid 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 4 bytes 240 (240.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 4 bytes 240 (240.0 B)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

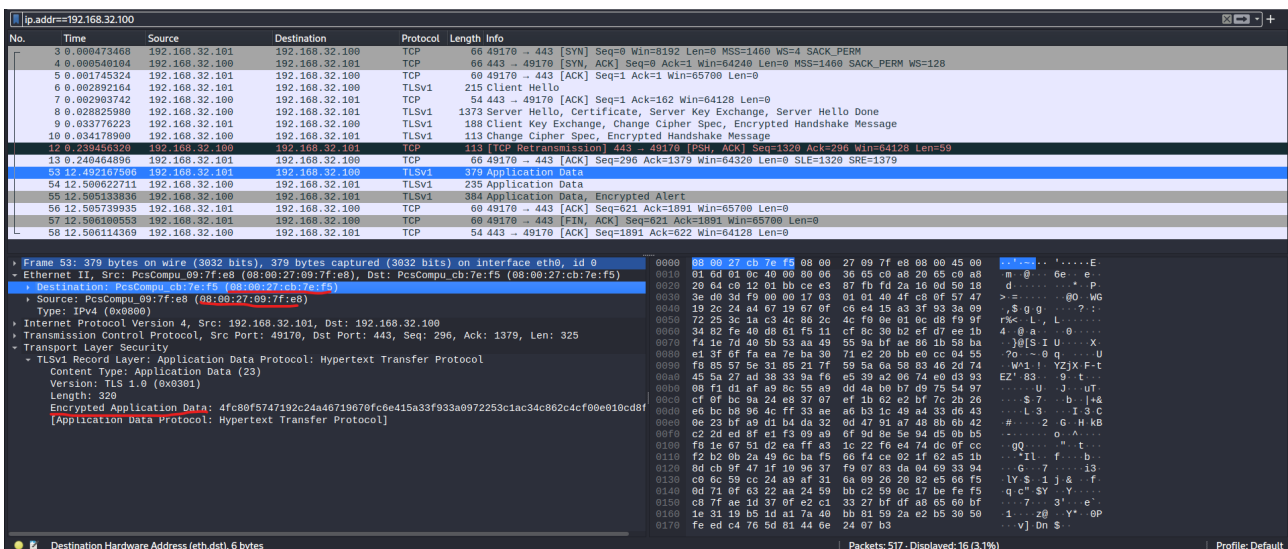
Con il comando **sudo nano /etc/inetsim/inetsim.conf** vado ad editare il file di configurazione di inetsim, andando ad attivare i servizi DNS e HTTPS come in figura e creando il record per associare epicode.internal all'indirizzo 192.168.32.100. Inoltre attivo i servizi sull'ip 192.168.32.100 modificando l'impostazione di default del bind address:

```
root@kali: ~  
File Actions Edit View Help  
GNU nano 7.2 /etc/inetsim/inetsim.conf *  
# 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  
#start_service tftp  
#start_service irc  
#start_service ntp  
#start_service finger  
#start_service ident  
#start_service syslog  
#start_service time_tcp  
#start_service time_udp  
#start_service daytime_tcp  
#start_service daytime_udp  
#start_service echo_tcp  
  
#####  
# dns_static  
#  
# Static mappings for DNS  
#  
# Syntax: dns_static <fqdn hostname> <IP address>  
#  
# 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  
  
#####  
# service_bind_address  
#  
# IP address to bind services to  
#  
# Syntax: service_bind_address <IP address>  
#  
# Default: 127.0.0.1  
#  
service_bind_address 192.168.32.100
```

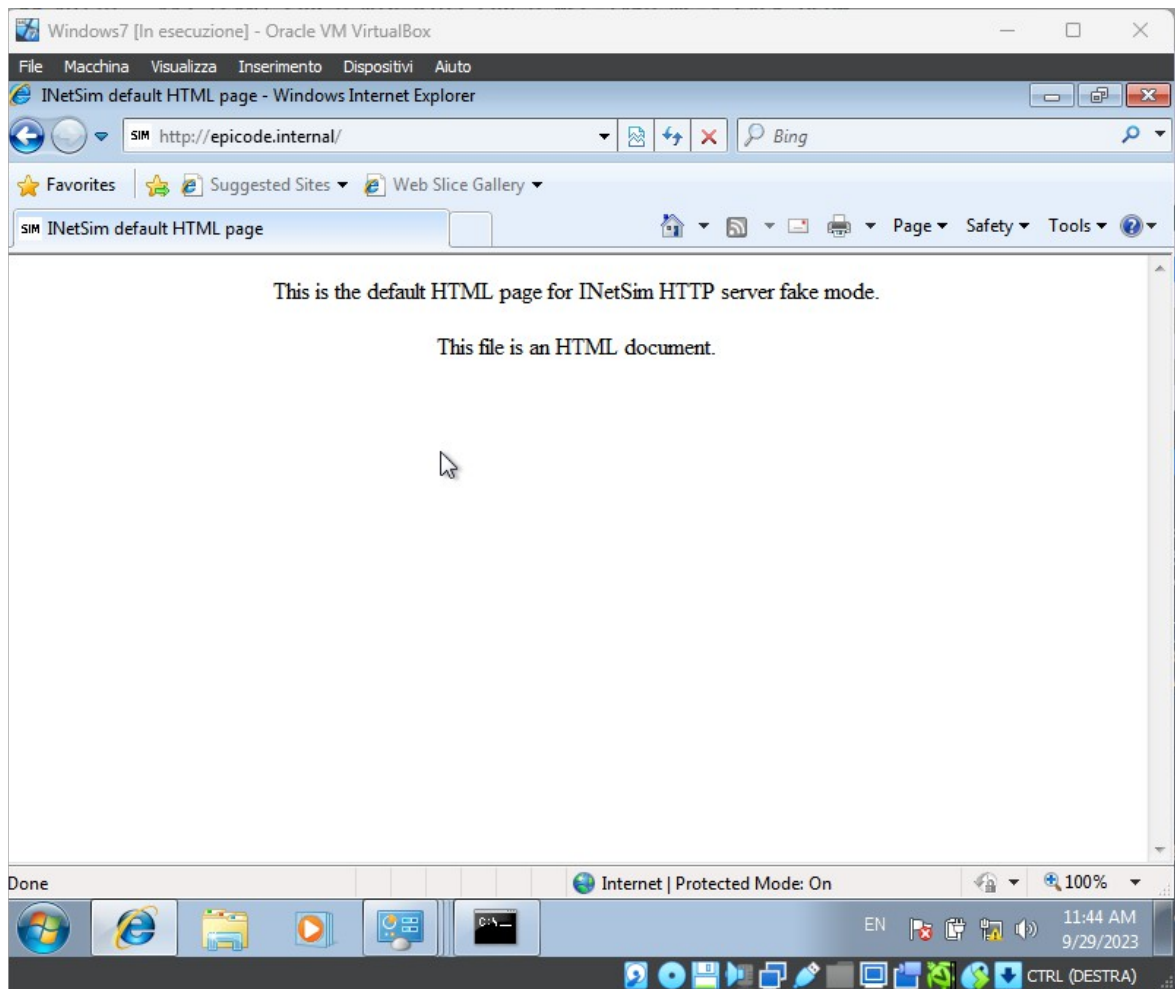
Successivamente da W7 punto da browser all'indirizzo epicode.internal. Accettando il rischio per la mancanza di certificato viene esposta questa pagina:



Di seguito la cattura con wireshark filtrata per ip 192.168.32.100, nell'immagine ho evidenziato in rosso i mac-address sorgente e destinazione di un pacchetto Application Data e la parte dove informa che il contenuto del payload è criptato:



Ripeto poi il procedimento visto sopra per attivare il servizio HTTP da inetsim andando a togliere il commento questa volta sul service_http su macchina Kali ed effettuo una richiesta HTTP da W7 verso Kali:



Traccio anche questa comunicazione con Wireshark da Kali. Noto che questa volta il testo html scambiato risulta visibile, evidenziato in rosso nell'immagine assieme ai mac-address. Questo perché il protocollo HTTP non si avvale di cifratura come l'HTTPS:

The image shows a Wireshark capture of network traffic on the interface eth0. The packet list on the left shows several TCP and HTTP packets. The selected packet is a GET request (No. 23) from 192.168.32.100 to 192.168.32.101. The packet details pane shows the structure of the HTTP request, including the GET method, host, and user-agent. The packet bytes pane shows the raw data, with the HTML response body highlighted in red. The response body contains the text: "This is the default HTML page for InetSim HTTP server fake mode." and "This file is an HTML document."

Frame 23: 312 bytes on wire (2496 bits), 312 bytes captured (2496 bits) on interface eth0, id 0

Ethernet II, Src: PcsCompu_cb7e:7f:5 (08:00:27:cb:7e:7f), Dst: PcsCompu_09:7f:e8 (08:00:27:09:7f:e8)

Destination: PcsCompu_09:7f:e8 (08:00:27:09:7f:e8)

Source: PcsCompu_cb7e:7f:5 (08:00:27:cb:7e:7f)

Type: IPv4 (0x0800)

Internet Protocol Version 4, Src: 192.168.32.100, Dst: 192.168.32.101

Transmission Control Protocol, Src Port: 80, Dst Port: 49174, Seq: 151, Ack: 287, Len: 258

[2 Reassembled TCP Segments (408 bytes): #22(150), #23(258)]

Hypertext Transfer Protocol

Line-based text data: text/html (10 lines)

<html>\n

<head>\n

<title>InetSim default HTML page</title>\n

</head>\n

<body>\n

<p></p>\n

<p align="center">This is the default HTML page for InetSim HTTP server fake mode.</p>\n

<p align="center">This file is an HTML document.</p>\n

</body>\n

</html>\n

Frame (312 bytes) Reassembled TCP (408 bytes)

Packets: 125 - Displayed: 10 (8.0%)

Profile: Default