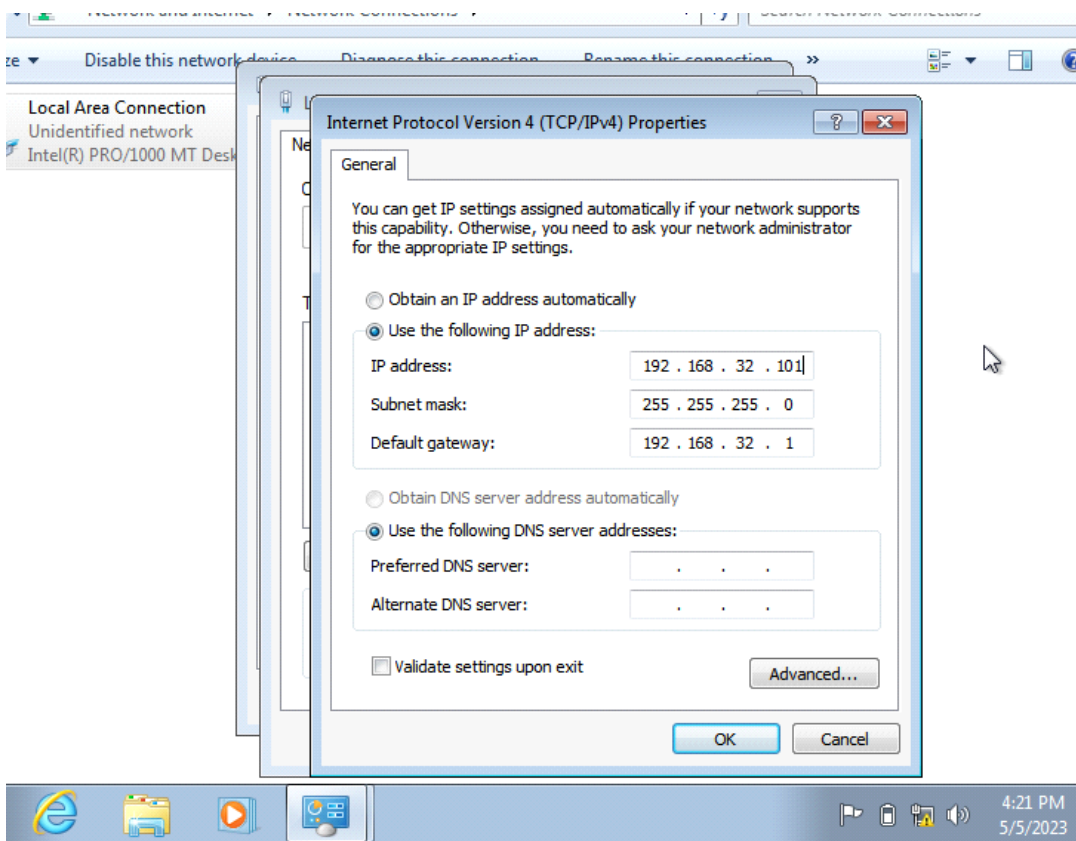


EPICODE ESERCIZIO

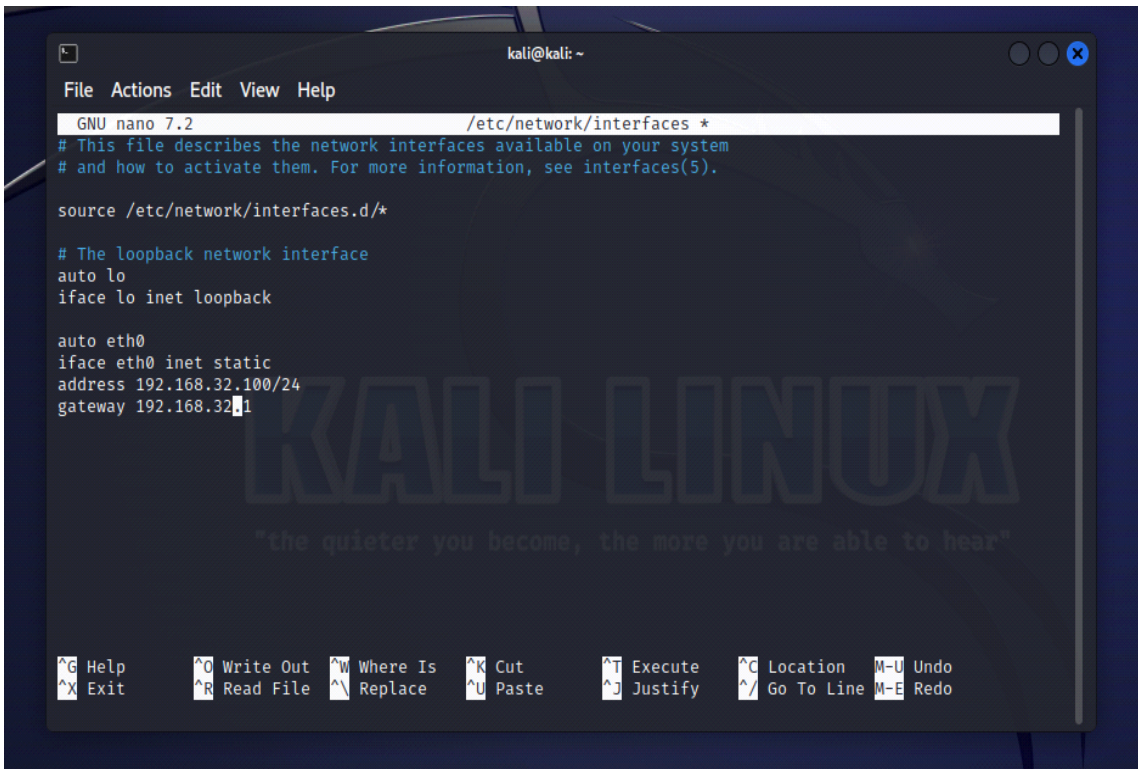
Simulazione in ambiente virtuale di un'architettura client-server in cui in client con ip 192.168.32.101 richiede tramite web browser una risorsa all'hostname epicode.internal che risponde all'indirizzo 192.168.32.100. Intercettando poi la comunicazione con wireshark evidenziando i MAC address di sorgente e destinazione e il contenuto della richiesta HTTPS e HTTP.

PASSAGGI :

1 -Inserisco indirizzo ip su Windows 7



2-Inserisco indirizzo su Kali Linux e verifico



A screenshot of a terminal window in Kali Linux. The terminal shows the nano text editor editing the file `/etc/network/interfaces`. The file content is as follows:

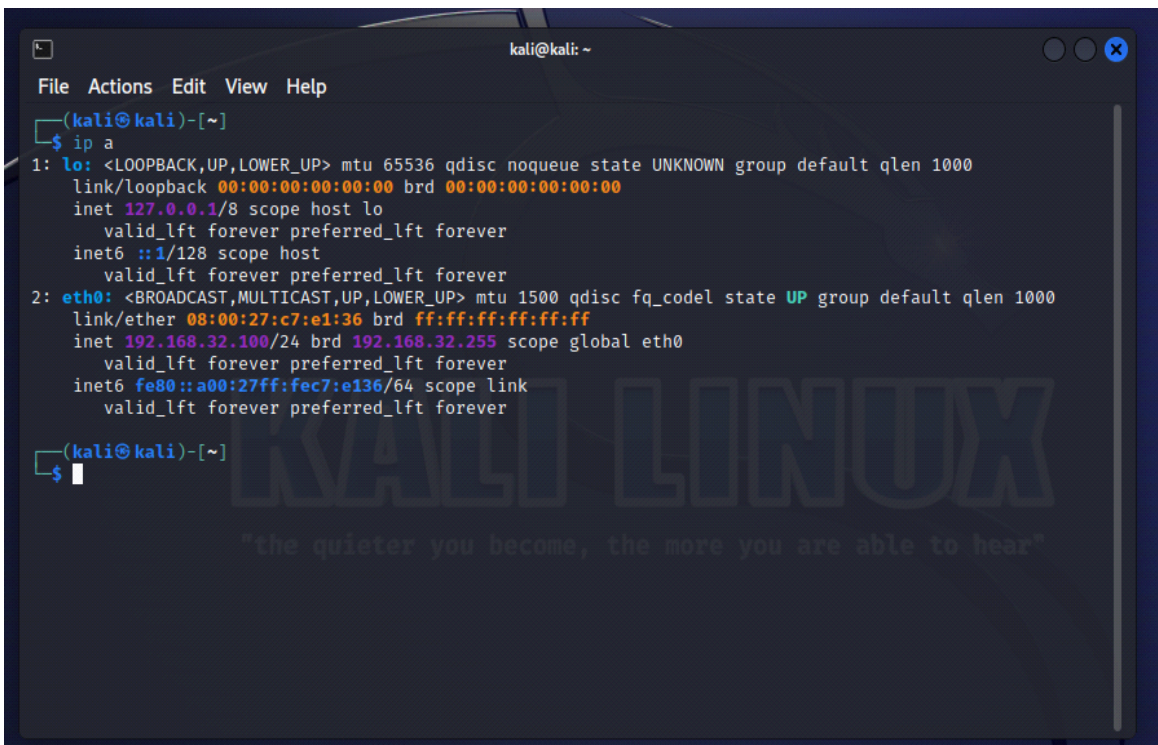
```
GNU nano 7.2 /etc/network/interfaces *
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
address 192.168.32.100/24
gateway 192.168.32.1
```

The terminal window has a menu bar with 'File', 'Actions', 'Edit', 'View', and 'Help'. At the bottom, there is a status bar with various keyboard shortcuts like `^G Help`, `^O Write Out`, etc.

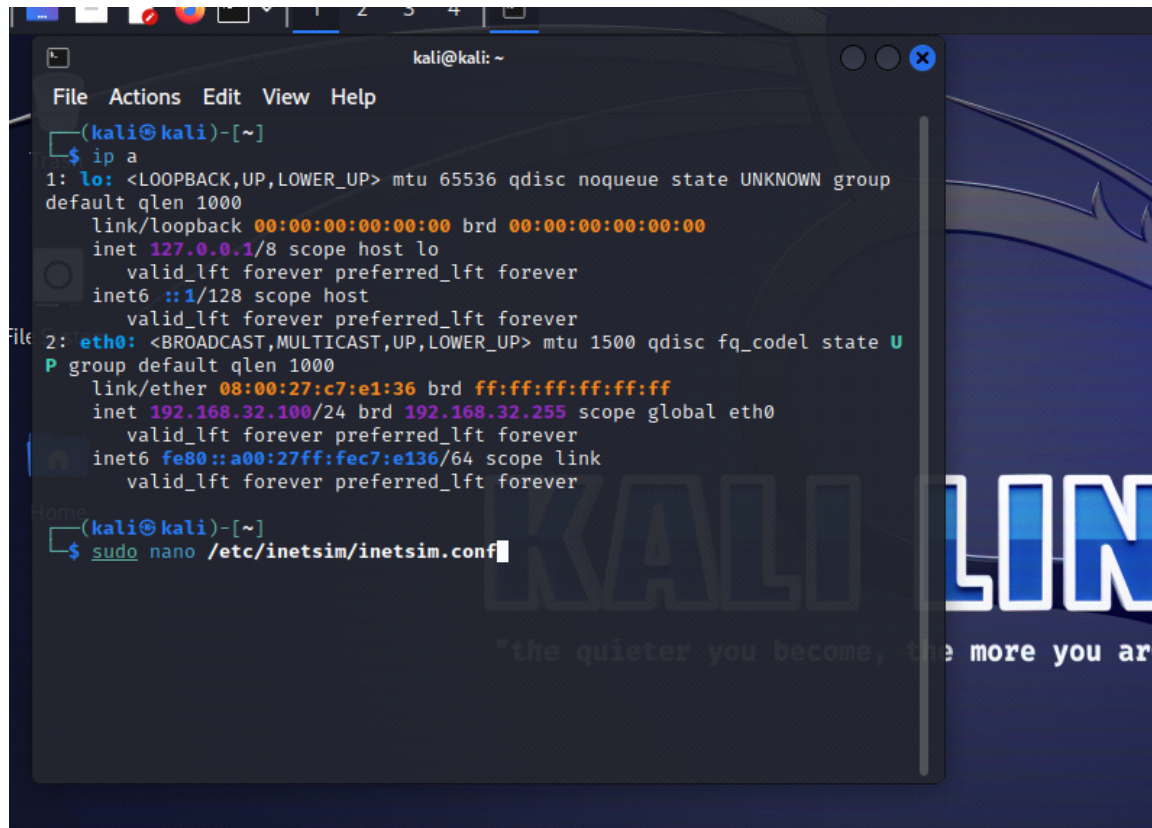


A screenshot of a terminal window in Kali Linux. The terminal shows the output of the `ip a` command. The output is as follows:

```
(kali@kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:c7:e1:36 brd ff:ff:ff:ff:ff:ff
    inet 192.168.32.100/24 brd 192.168.32.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fec7:e136/64 scope link
        valid_lft forever preferred_lft forever
```

The terminal window has a menu bar with 'File', 'Actions', 'Edit', 'View', and 'Help'. At the bottom, there is a status bar with various keyboard shortcuts like `^G Help`, `^O Write Out`, etc.

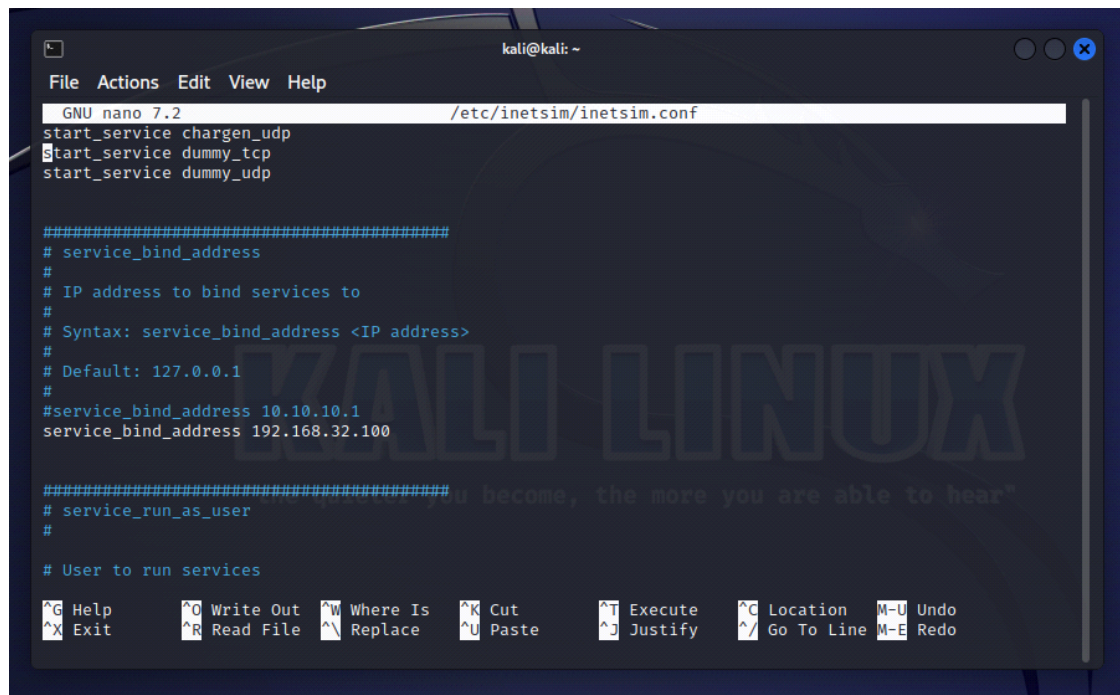
3 - Lancio il comando `sudo nano /etc/inetsim/inetsim.conf` per la configurazione del dns



A terminal window on a Kali Linux system. The prompt is `(kali@kali)-[~]`. The user runs `ip a`, displaying network interface details for `lo` and `eth0`. The `lo` interface has IP `127.0.0.1`. The `eth0` interface has IP `192.168.32.100`. After displaying the output, the user runs `sudo nano /etc/inetsim/inetsim.conf`, opening the configuration file in nano editor.

```
(kali@kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group
default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state U
P group default qlen 1000
    link/ether 08:00:27:c7:e1:36 brd ff:ff:ff:ff:ff:ff
    inet 192.168.32.100/24 brd 192.168.32.255 scope global eth0
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fec7:e136/64 scope link
        valid_lft forever preferred_lft forever
(kali@kali)-[~]
$ sudo nano /etc/inetsim/inetsim.conf
```

3 - Imposto il service bind address



The nano editor window showing the `/etc/inetsim/inetsim.conf` file. The file contains configuration for starting services and setting the bind address. The `service_bind_address` is set to `192.168.32.100`. The `service_run_as_user` is set to `#`. The nano editor interface includes a menu bar and a status bar at the bottom.

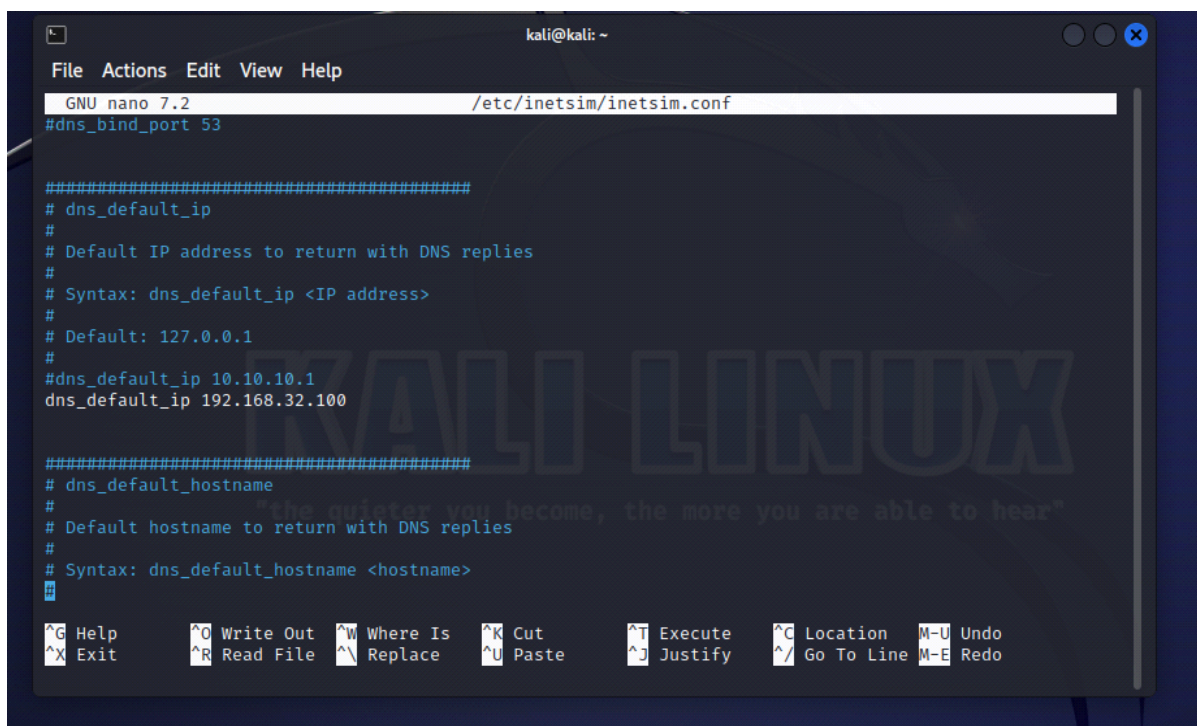
```
GNU nano 7.2 /etc/inetsim/inetsim.conf
start_service chargen_udp
start_service dummy_tcp
start_service dummy_udp

#####
# 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

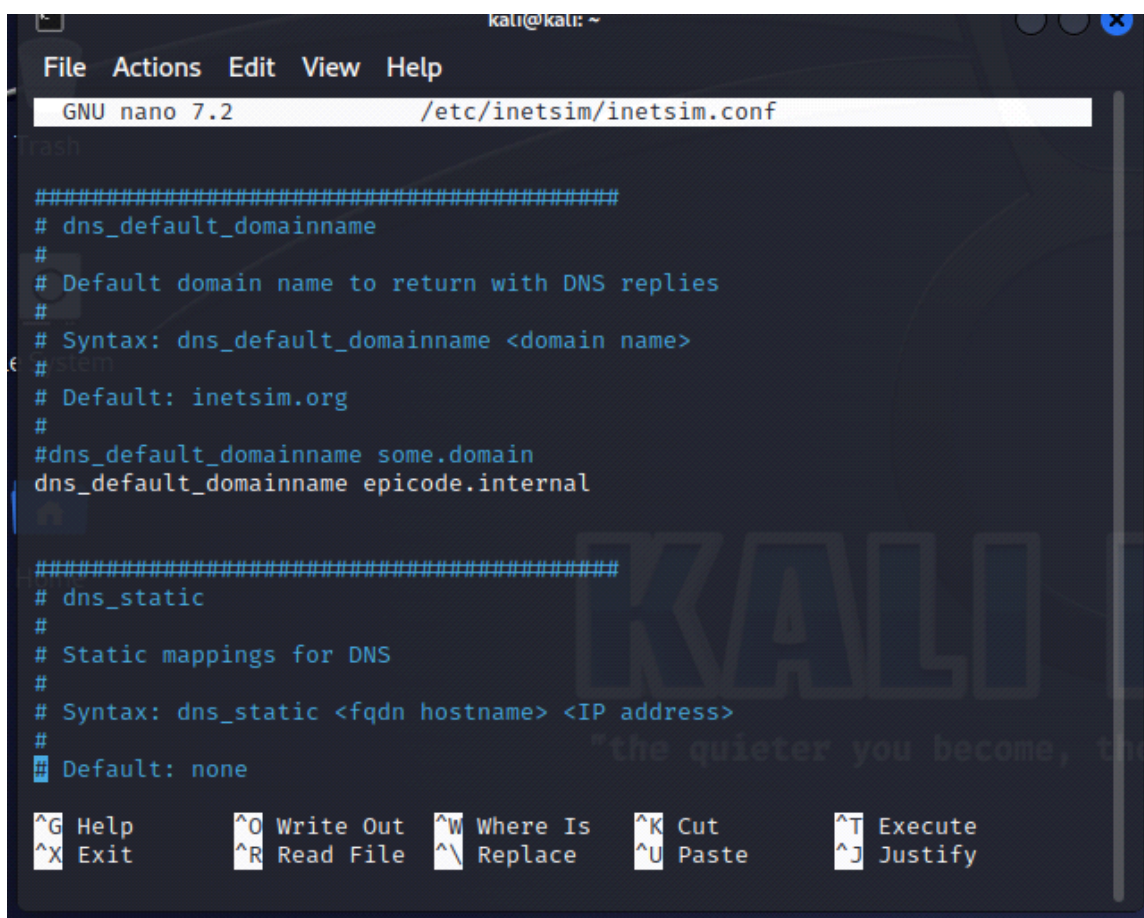
^G Help      ^O Write Out ^M Where Is  ^K Cut       ^T Execute  ^C Location  M-U Undo
^X Exit      ^R Read File ^N Replace  ^U Paste     ^J Justify  ^_ Go To Line M-E Redo
```


4 - Imposto l'ip del dns



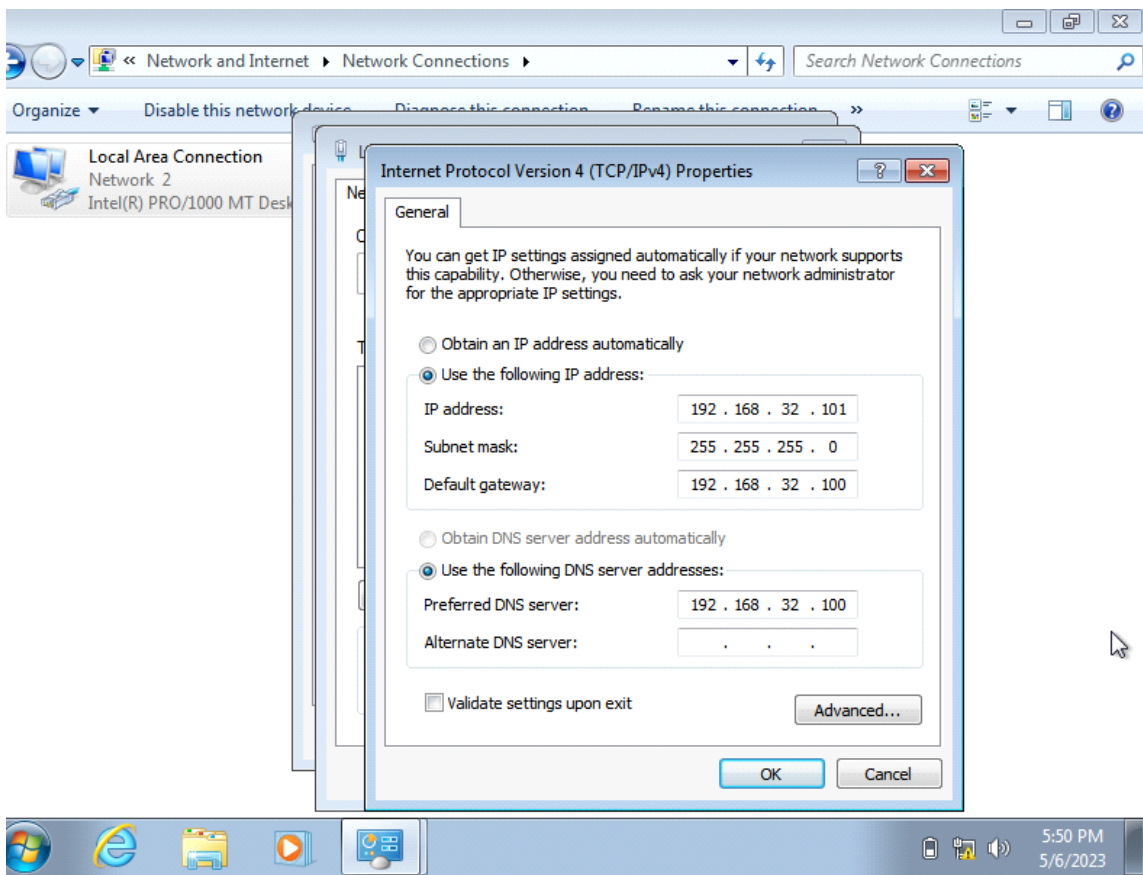
```
kali@kali: ~  
File Actions Edit View Help  
GNU nano 7.2 /etc/inetsim/inetsim.conf  
#dns_bind_port 53  
  
#####  
# dns_default_ip  
#  
# Default IP address to return with DNS replies  
#  
# Syntax: dns_default_ip <IP address>  
#  
# Default: 127.0.0.1  
#  
#dns_default_ip 10.10.10.1  
dns_default_ip 192.168.32.100  
  
#####  
# dns_default_hostname  
#  
# Default hostname to return with DNS replies  
#  
# Syntax: dns_default_hostname <hostname>  
#  
#####  
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute  
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify  
^C Location  M-U Undo  
^_ Go To Line M-E Redo
```

5 - Imposto il domain name con epicode.internal

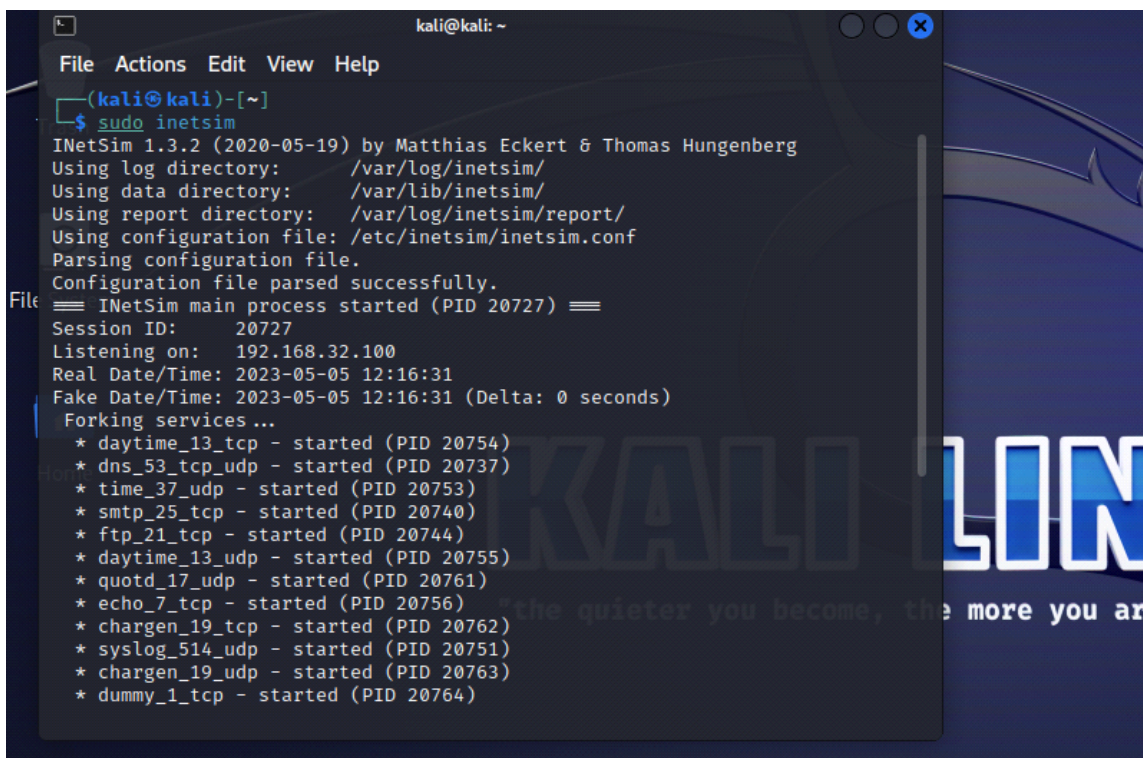


```
kali@kali: ~  
File Actions Edit View Help  
GNU nano 7.2 /etc/inetsim/inetsim.conf  
#dns_bind_port 53  
  
#####  
# dns_default_domainname  
#  
# Default domain name to return with DNS replies  
#  
# Syntax: dns_default_domainname <domain name>  
#  
# Default: inetsim.org  
#  
#dns_default_domainname some.domain  
dns_default_domainname epicode.internal  
  
#####  
# dns_static  
#  
# Static mappings for DNS  
#  
# Syntax: dns_static <fqdn hostname> <IP address>  
#  
# Default: none  
#  
#####  
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute  
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify  
^C Location  M-U Undo  
^_ Go To Line M-E Redo
```

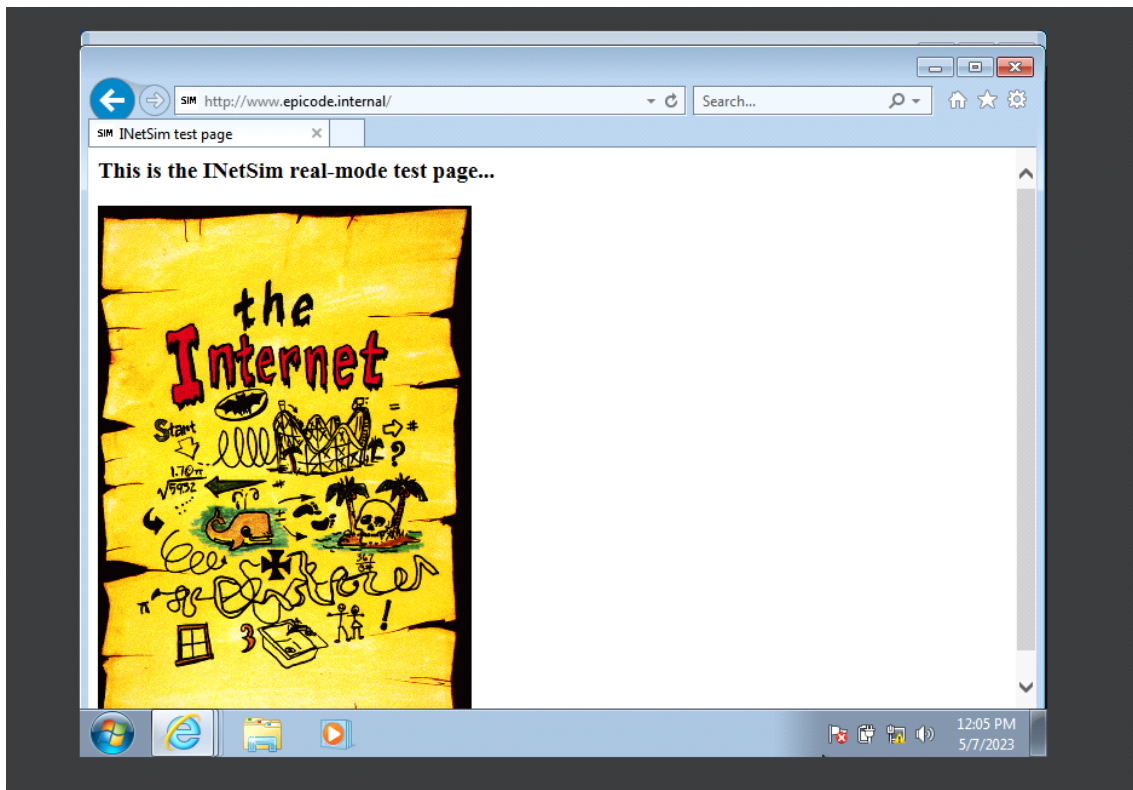
6 - Configuro in Windows 7 l'indirizzo che risponde al dns



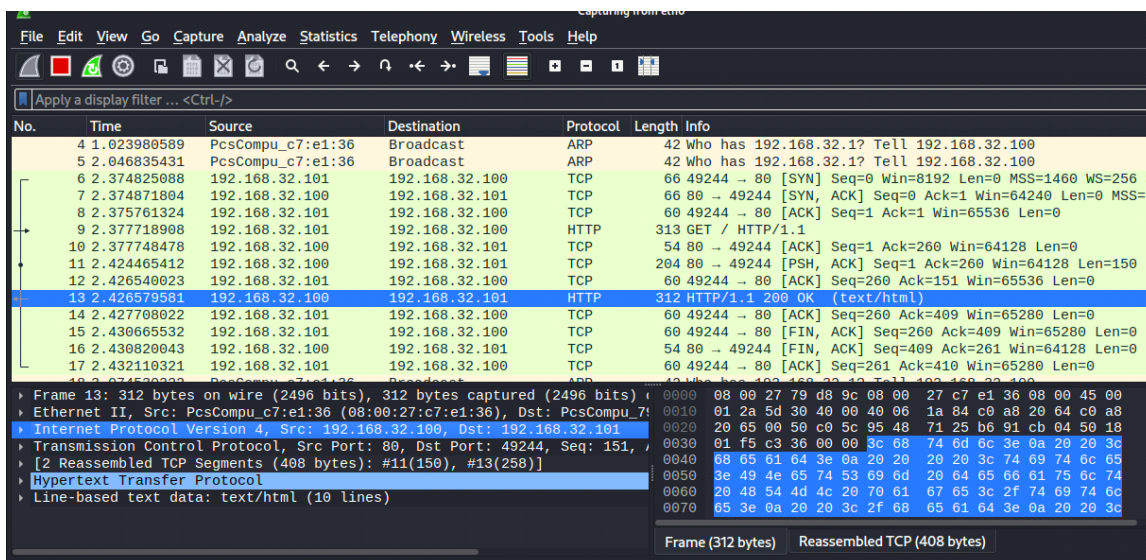
7 - Avvio il server con il comando : sudo intesim



8 - Apro internet explorer in windows 7 e lancio la richiesta con epicode.internal



9 - Lancio Wireshark in eth0 per l'intercettazione dei pacchetti



10 - Pacchetto http intercettato

```

> Frame 26: 312 bytes on wire (2496 bits), 312 bytes captured (2496 bits) on interface eth0, id 0
- Ethernet II, Src: PcsCompu_c7:e1:36 (08:00:27:c7:e1:36), Dst: PcsCompu_79:d8:9c (08:00:27:79:d8:9c)
  > Destination: PcsCompu_79:d8:9c (08:00:27:79:d8:9c)
  > Source: PcsCompu_c7:e1:36 (08:00:27:c7:e1:36)
  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: 49518, Seq: 151, Ack: 202, Len: 258
  Source Port: 80
  Destination Port: 49518
  [Stream index: 1]
  [Conversation completeness: Complete, WITH_DATA (31)]
  [TCP Segment Len: 258]
0000  08 00 27 79 d8 9c 08 00 27 c7 e1 36 08 00 45 00  ..y...6.E
0010  01 2a 19 84 40 00 40 06 5e 30 c0 a8 20 64 c0 a8  .*.@@.^0..d.
0020  20 65 00 50 c1 6e 6a 23 e9 2f 7b 4f 17 32 50 19  e.P nj# ./0.2P
0030  01 f5 c3 36 00 00 3c 68 74 6d 6c 3e 0a 20 20 3c  ..6..<html>.<
0040  68 65 61 64 3e 0a 20 20 20 20 3c 74 69 74 6c 65  head>.<title
0050  3e 49 4e 65 74 53 69 6d 20 64 65 66 61 75 6c 74  >INetSim default
0060  20 48 54 4d 4c 20 70 61 67 65 3c 2f 74 69 74 6c  HTML pa ge</titl
0070  65 3e 0a 20 20 3c 2f 68 65 61 64 3e 0a 20 20 3c  e>.</head>.<
0080  62 f6 64 79 3e 0a 20 20 20 20 3c 70 3e 3c 2f 70  body>.<p></p
0090  3e 0a 20 20 20 20 3c 70 20 61 6c 69 67 6e 3d 22  >.<p align="
00a0  63 65 6e 74 65 72 22 3e 54 68 69 73 20 69 73 20  center"> This is
00b0  74 68 65 20 64 65 66 61 75 6c 74 20 48 54 4d 4c  the defa ult HTML
00c0  20 70 61 67 65 20 66 6f 72 20 49 4e 65 74 53 69  page fo r INetSi

Frame (312 bytes)  Reassembled TCP (408 bytes)
```

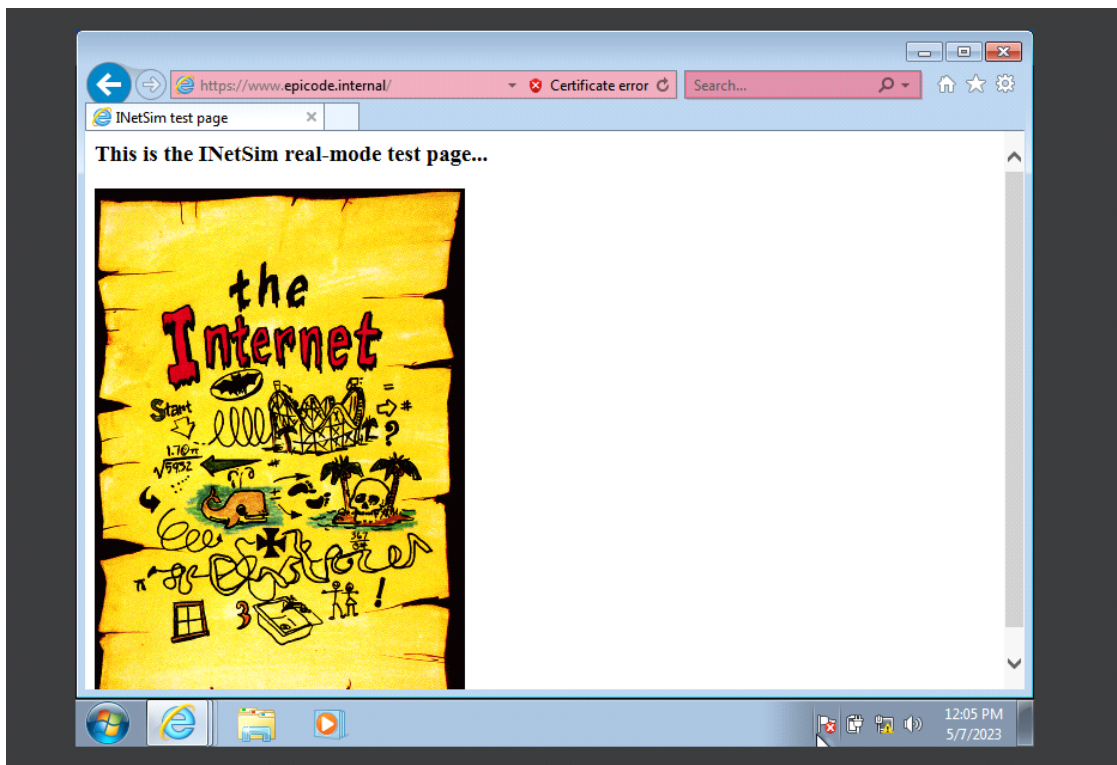
11 - Intercettiamo il pacchetto https abilitando il server https con start_service https

```

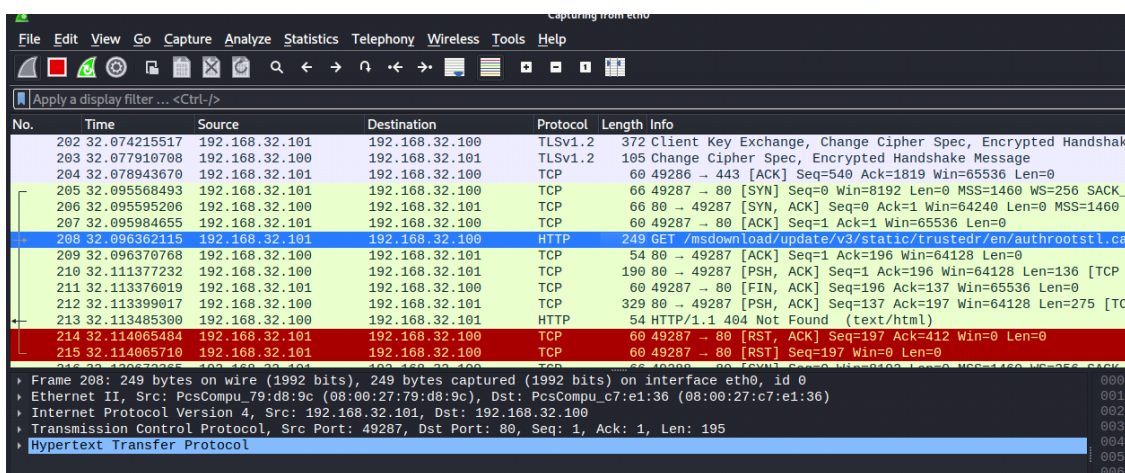
kali@kali: ~
File Actions Edit View Help
GNU nano 7.2 /etc/inetsim/inetsim.conf
# echo_udp, discard_tcp, discard_udp, quotd_tcp,
# quotd_udp, chargen_tcp, chargen_udp, finger,
# ident, syslog, dummy_tcp, dummy_udp, smtps, pop3s,
# ftps, irc, https
#
start_service dns
#start_service http
start_service https
start_service smtp
start_service smtps
start_service pop3
start_service pop3s
start_service ftp
start_service ftps
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

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

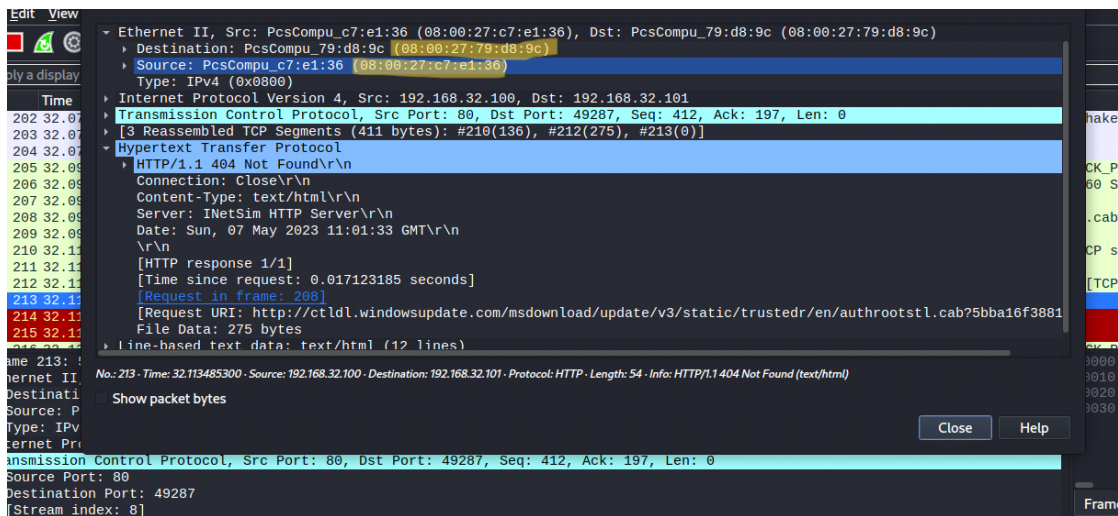
12 - Apro internet explorer in windows 7 e lancio la richiesta con epicode.internal



13 - Lancio Wireshark in eth0 per l'intercettazione dei pacchetti



14 - Pacchetto https intercettato



15 - **Conclusione** : intercettando tutti e due i pacchetti possiamo dedurre che mentre nel pacchetto http possiamo vedere tutte le informazioni , nel pacchetto https no perchè criptato.