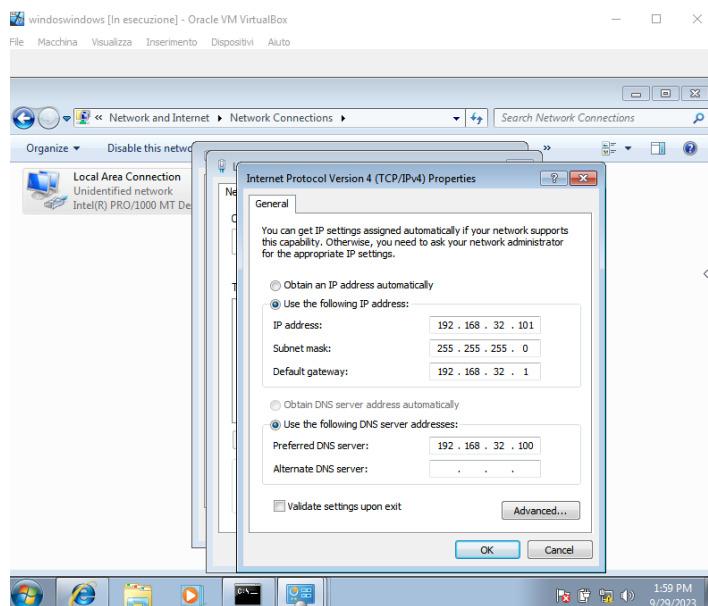
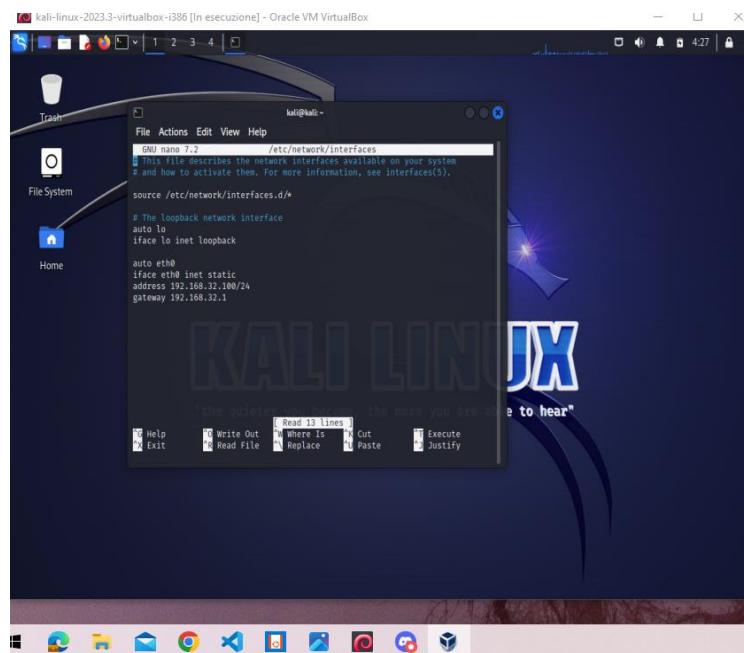


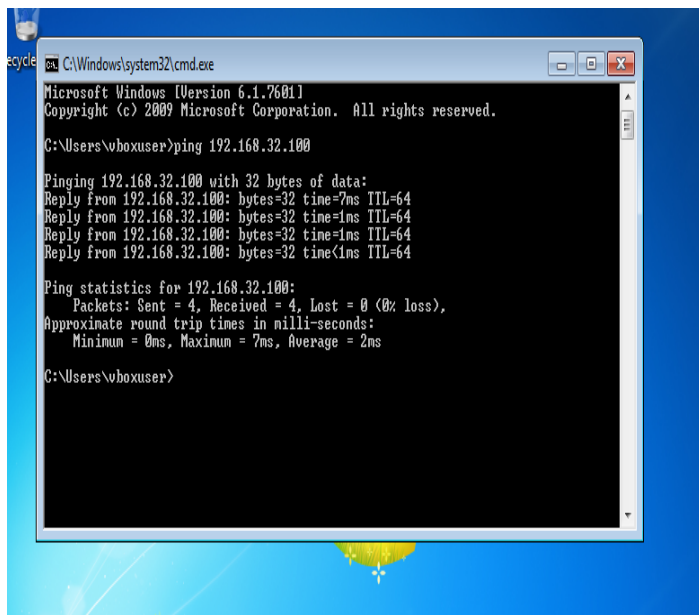
ESERCIZIO WEEK 1

1° FASE: settaggio indirizzi ip, sia su Kali Linux che su windows 7



Sappiamo che dopo dovremo usare Kali come client server e windows7 come client; perciò, impostiamo l'ip di Kali come DNS server in windows, andremo anche ad assicurarci che il firewall impostato di base su windows non ci dia noia.

Per assicurarci che non ci siano problemi fra i dispositivi li facciamo 'pingare' fra di loro



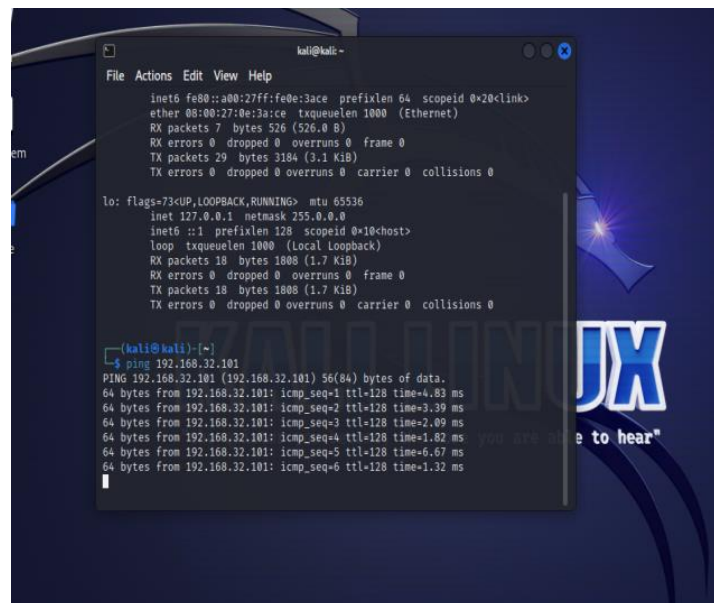
```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\ vboxuser>ping 192.168.32.100

Pinging 192.168.32.100 with 32 bytes of data:
Reply from 192.168.32.100: bytes=32 time=7ms TTL=64
Reply from 192.168.32.100: bytes=32 time=1ms TTL=64
Reply from 192.168.32.100: bytes=32 time=1ms TTL=64
Reply from 192.168.32.100: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.32.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 7ms, Average = 2ms

C:\Users\ vboxuser>
```



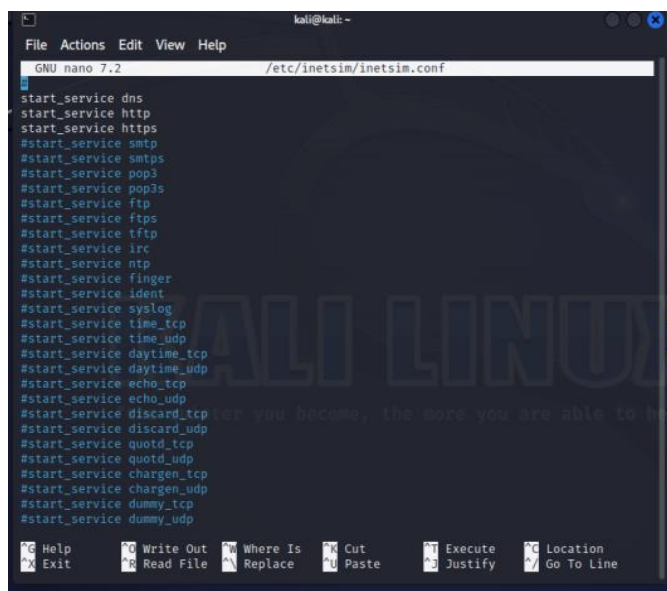
```
kali@kali:~$ ifconfig
eth0: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 18 bytes 1808 (1.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 18 bytes 1808 (1.7 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 0<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 18 bytes 1808 (1.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 18 bytes 1808 (1.7 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

kali@kali:~$ ping 192.168.32.101
PING 192.168.32.101 (192.168.32.101) 56(84) bytes of data.
64 bytes from 192.168.32.101: icmp_seq=1 ttl=128 time=4.83 ms
64 bytes from 192.168.32.101: icmp_seq=2 ttl=128 time=3.39 ms
64 bytes from 192.168.32.101: icmp_seq=3 ttl=128 time=2.09 ms
64 bytes from 192.168.32.101: icmp_seq=4 ttl=128 time=1.82 ms
64 bytes from 192.168.32.101: icmp_seq=5 ttl=128 time=6.67 ms
64 bytes from 192.168.32.101: icmp_seq=6 ttl=128 time=1.32 ms
```

Una volta controllato che le macchine virtuali riescono a comunicare fra loro possiamo alla seconda fase.

2°FASE: Selezione delle impostazioni corrette su inetsim (ricordandoci che inetsim è un simulatore di servizi di rete) e noi vogliamo utilizzare i servizi DNS prima con HTTPS e poi HTTP, la vm Kali farà da router quindi nelle impostazioni andrà messo l'IP di Kali.



```
kali@kali:~$ nano /etc/inetsim/inetsim.conf
GNU nano 7.2 /etc/inetsim/inetsim.conf

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
#start_service echo_udp
#start_service discard_tcp
#start_service discard_udp
#start_service quotd_tcp
#start_service quotd_udp
#start_service chargen_tcp
#start_service chargen_udp
#start_service dummy_tcp
#start_service dummy_udp
```

← I tre servizi che andiamo ad adoperare

```
kali@kali: ~  
File Actions Edit View Help  
GNU nano 7.2 /etc/inetsim/inetsim.conf  
#####  
# dns_bind_port  
#  
# Port number to bind DNS service to  
#  
# Syntax: dns_bind_port <port number>  
#  
# Default: 53  
#  
#dns_bind_port 53  
  
#####  
# dns_default_ip  
#  
# Default IP address to return with DNS replies  
#  
# Syntax: dns_default_ip <IP address>  
#  
# Default: 192.168.32.100  
#  
dns_default_ip 192.168.32.100  
  
#####  
# dns_default_hostname  
#  
# Default hostname to return with DNS replies  
#  
#####  
Help Write Out Where Is Cut Execute Location  
Exit Read File Replace Paste Justify Go To Line
```

← uno dei vari settaggi di DNS service che andremo a impostare (Togliendo # per inserire il comando e modificano in questo caso con l'ip Kali)

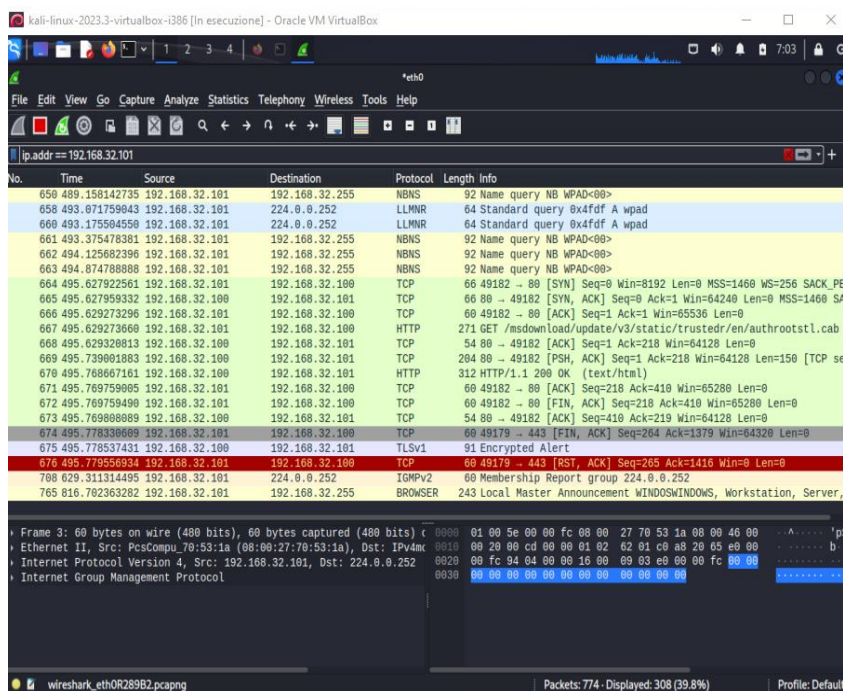
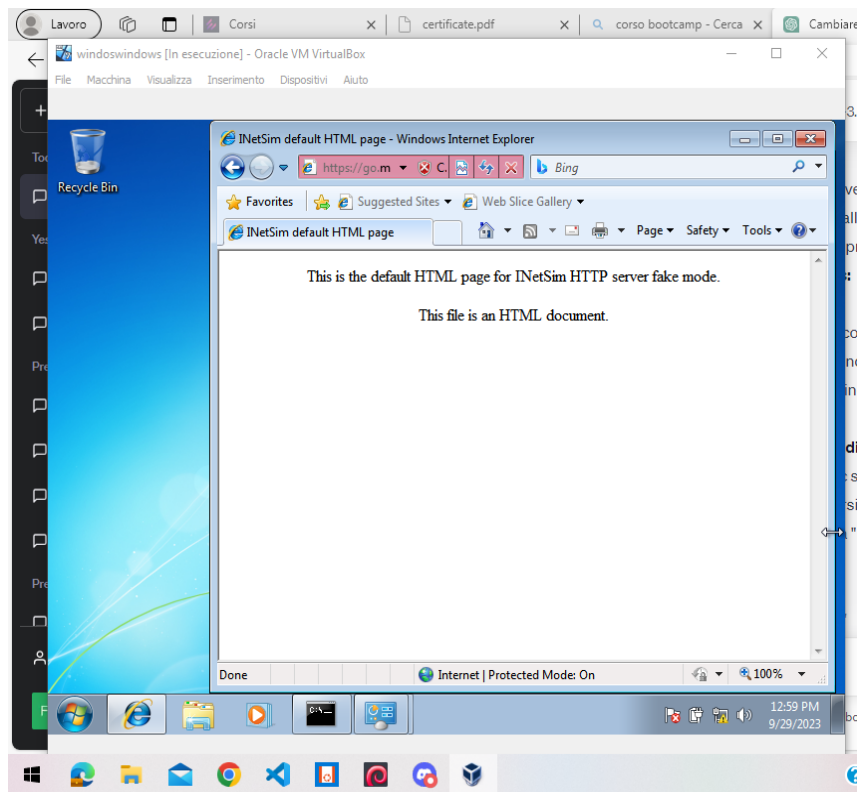
Attivazione di Inetsim

```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ sudo nano /etc/inetsim/inetsim.conf  
[sudo] password for kali:  
  
(kali@kali)-[~]  
$ 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.  
Warning: Unknown option 'Service' in configuration file '/etc/inetsim/inetsim.conf' line 183  
Warning: Unknown option 'Service' in configuration file '/etc/inetsim/inetsim.conf' line 368  
Configuration file parsed successfully.  
== INetSim main process started (PID 6986) ==  
Session ID: 6986  
Listening on: 192.168.32.100  
Real Date/Time: 2023-09-29 08:51:25  
Fake Date/Time: 2023-09-29 08:51:25 (Delta: 0 seconds)  
Forking services...  
* dns_53_tcp_udp - started (PID 7001)  
print() on closed filehandle MLOG at /usr/share/perl5/Net/DNS/Nameserver.pm line 399.  
print() on closed filehandle MLOG at /usr/share/perl5/Net/DNS/Nameserver.pm line 399.  
* http_80_tcp - started (PID 7002)  
* https_443_tcp - started (PID 7003)  
done.  
Simulation running.  
█
```

3° FASE: sniffing

Andiamo sul browser della macchina con windows 7 e scriviamo nella barra di ricerca il dominio che abbiamo impostato nelle opzioni di DNS su inetsim: “epicode. internal”, poi utilizziamo wireshark sulla macchina Kali per “sniffare” il traffico della rete, usiamo il protocollo HTTPS prima e http poi, questo passaggio ci permette di vedere chiaramente la differenza che fra questi due protocolli, specialmente per quanto riguarda la crittazione di HTTPS

HTTPS



HTTP

