

Cyber security analyst

progetto finale di modulo 1

22/03/2025

Mattia Carlesso.

Requisiti e servizi:

- kali linux con IP 192.168.32.100
- windows 7 con IP 192.168.32.101
- Servizio HTTP ed HTTPS attivo.
- servizio DNS: attivo

Richiesta:

simulazione di un laboratorio virtuale, nella quale su kali attiviamo i servizi http https e DNS in modo tale che con wireshark, una volta che ci collegheremo con windows a questi servizi riusciremo a sniffare quanti più dati possibili.

indice:

- 1. Configurazione delle VM.***
- 2. Test di comunicazione.***
- 3. Avvio servizio DNS su Kali***
- 4. avvio servizio HTTP/HTTPS***
- 5. sniffing con wireshark***
- 6. analisi generale***

Step 1

configurazione delle VM

Kali

come già svolto precedentemente negli esercizi di pratica la configurazione degli IP si svolge, inserendo su Kali il comando:

```
sudo nano /etc/interface/interface.conf
```

```
File Actions Edit View Help
GNU nano 8.3 story file /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).
[sudo] password for kali:
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
```

	Protocol	Length	Info
5	NBNS	92	Name query NB WPAD<00>
0	DNS	84	Standard query 0x793b A www.update.microsoft.com
0	DNS	84	Standard query 0x793b A www.update.microsoft.com
0	DNS	84	Standard query 0x793b A www.update.microsoft.com
0	DNS	84	Standard query 0x793b A www.update.microsoft.com
0	DNS	84	Standard query 0x793b A www.update.microsoft.com
0	ARP	60	Who has 192.168.32.101 Tell 192.168.32.101
0	ARP	60	Who has 192.168.32.101 Tell 192.168.32.101
0	ARP	60	Who has 192.168.32.101 Tell 192.168.32.101
0	LLMNR	54	Standard query 0xa93c A wpad

^G Help

^X Exit

^O Write Out

^R Read File

[Read 14 lines]

^F Where Is

^_ Replace

^K Cut

^U Paste

^T Execute

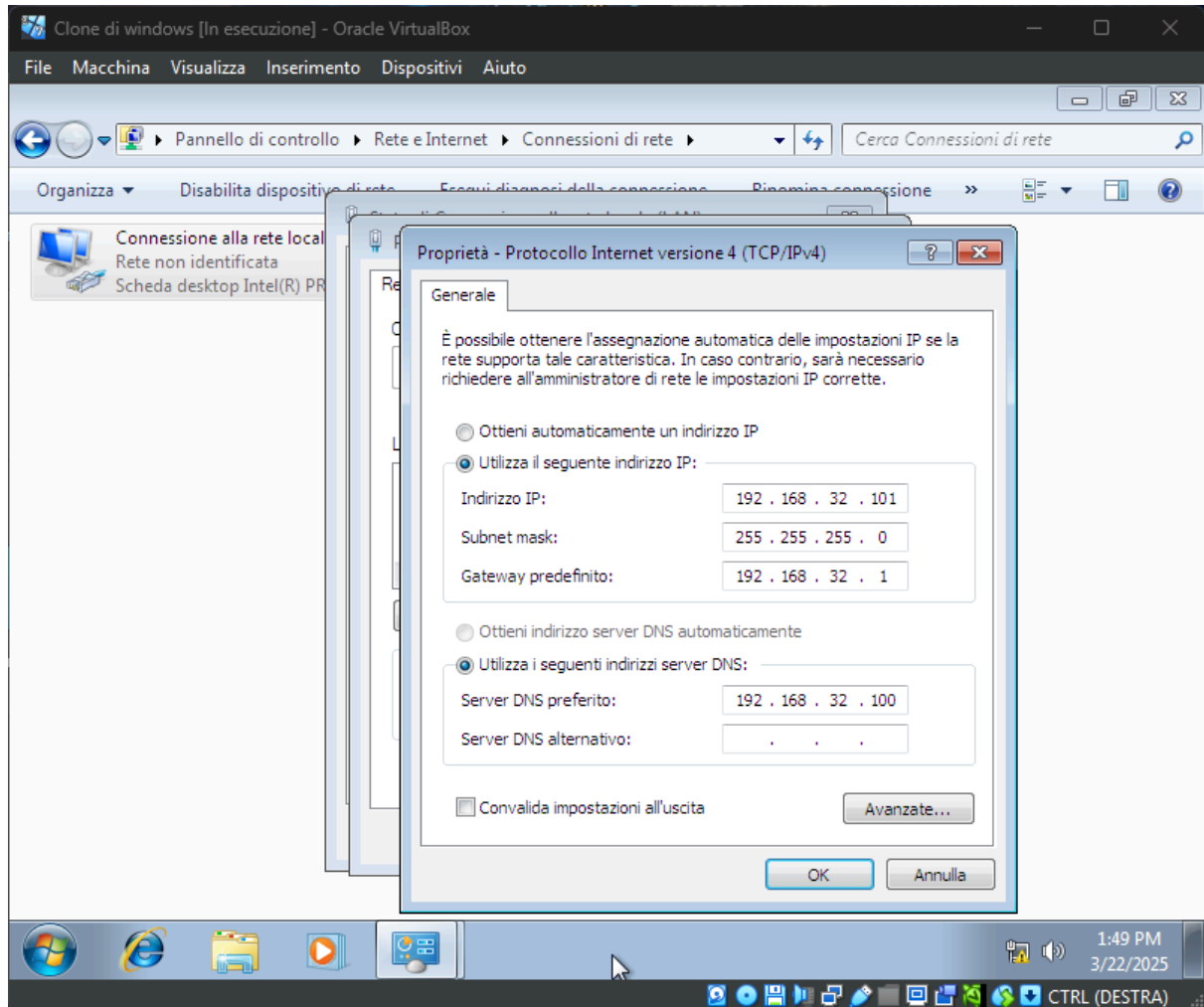
^J Justify

windows 7

per windows 7 la configurazione è differente basta seguire il path:

Rete e internet > Connessioni di rete >

per ritrovarci all'interno di un'interfaccia interagibile che ci permette le modifiche dell'indirizzo IP



STEP 2

Test di comunicazione delle VM

Una volta assegnati i parametri alle macchine andiamo ad effettuare dei test di comunicazione tra esse.

Da Kali a Windows

```
(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=1.43 ms  
64 bytes from 192.168.32.101: icmp_seq=2 ttl=128 time=0.691 ms  
64 bytes from 192.168.32.101: icmp_seq=3 ttl=128 time=0.668 ms  
^C  
— 192.168.32.101 ping statistics —  
3 packets transmitted, 3 received, 0% packet loss, time 2016ms  
rtt min/avg/max/mdev = 0.668/0.930/1.431/0.354 ms
```

Da Windows a Kali

```
C:\Users\vboxuser>ping 192.168.32.100  
Esecuzione di Ping 192.168.32.100 con 32 byte di dati:  
Risposta da 192.168.32.100: byte=32 durata<1ms TTL=64  
Risposta da 192.168.32.100: byte=32 durata<1ms TTL=64  
Risposta da 192.168.32.100: byte=32 durata<1ms TTL=64  
Risposta da 192.168.32.100: byte=32 durata<1ms TTL=64  
Statistiche Ping per 192.168.32.100:  
Pacchetti: Trasmessi = 4, Ricevuti = 4,  
Persi = 0 (0% persi),  
Tempo approssimativo percorsi andata/ritorno in millisecondi:  
Minimo = 0ms, Massimo = 0ms, Medio = 0ms  
C:\Users\vboxuser>_
```

Le macchine tra loro comunicano, quindi possiamo procedere allo step numero 3

STEP 3

Avvio servizio DNS su Kali

questo step ha molteplici modi per essere effettuato, io dopo svariate ricerche e tentativi ho deciso di affidarmi a [DNSCHEF](#), un tool molto semplice e intuitivo, già presente in Kali cercando in rete ho trovato il comando per attivare dnscchef e impostarlo .

```
dnscchef --fakedomains epicode.internal --fakeip 192.168.32.100 --nameservers 192.168.32.100 --interface 192.168.32.100
```

```
(kali@kali)-[~]
$ dnscchef --fakedomains epicode.internal --fakeip 192.168.32.100 --nameservers 192.168.32.100 --interface 192.168.32.100
/usr/bin/dnscchef:453: SyntaxWarning: invalid escape sequence '\/'
header += " / _ ` | ' _ V _ | / _ | ' _ \ / _ \ _ | \n"
/usr/bin/dnscchef:454: SyntaxWarning: invalid escape sequence '\_'
header += " | ( _ | | | | \ _ \ ( _ | | | | _ / | _ \n"
/usr/bin/dnscchef:455: SyntaxWarning: invalid escape sequence '\_'
header += " \_, _ | | | _ | \ _ | | | _ | \ _ | | \n"

  _ | version 0.4 | _ | / _ |
 _ | _ | _ | _ | _ | _ | _ | _ |
 _ | ( _ | | | | \ _ \ ( _ | | | | _ / | _ \n"
 _ | \_, _ | | | _ | \ _ | | | _ | \ _ | | \n"
                                     iphelix@thesprawl.org

(09:58:04) [*] DNSChef started on interface: 192.168.32.100
(09:58:04) [*] Using the following nameservers: 192.168.32.100
(09:58:04) [*] Cooking A replies to point to 192.168.32.100 matching: epicode.internal
(09:58:11) [*] 192.168.32.101: cooking the response of type 'A' for epicode.internal to 192.168.32.100
(09:58:11) [*] 192.168.32.101: cooking the response of type 'A' for epicode.internal to 192.168.32.100
```

con questa schermata il servizio dns è attivo e funzionante.

Step 4

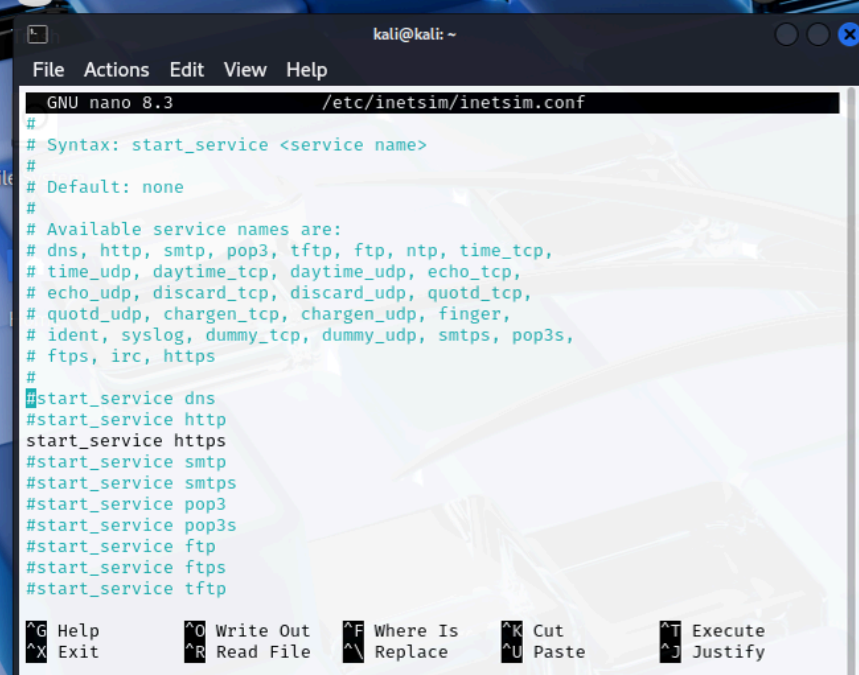
Attivazione HTTP e HTTPS su Kali

Per attivare i servizi si utilizza [inetsim](#) come già svolto durante le lezioni.

Il comando per effettuare modifiche è:

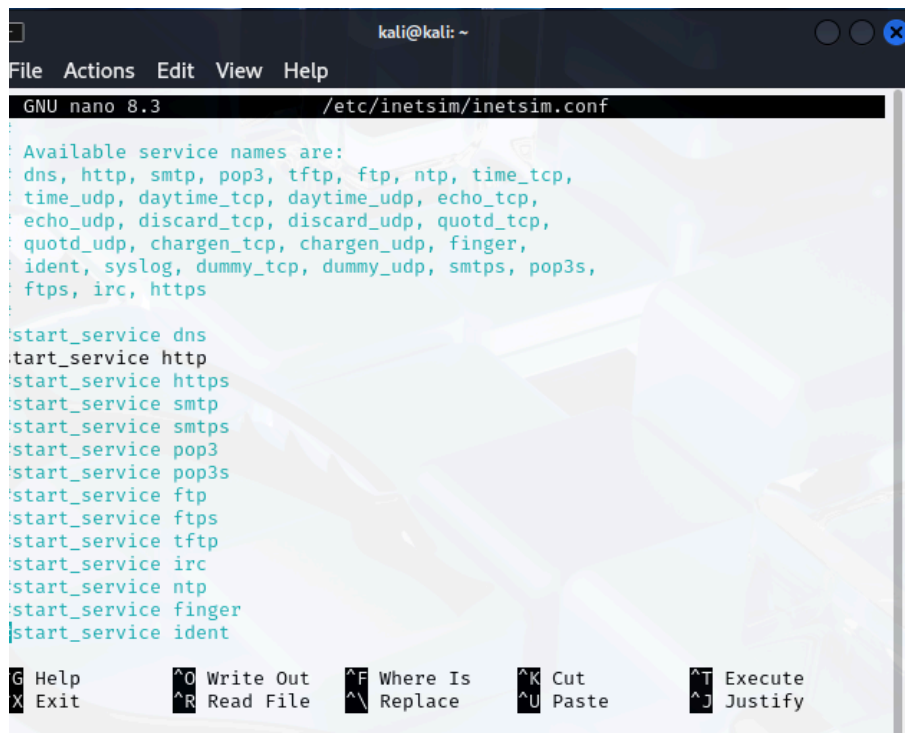
`sudo nano /etc/inetsim/inetsim.conf`

bisognerà modificare alcuni parametri per attivare il servizio



```
kali@kali: ~  
File Actions Edit View Help  
GNU nano 8.3 /etc/inetsim/inetsim.conf  
#  
# 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  
#start_service tftp  
  
^G Help      ^O Write Out  ^F Where Is   ^K Cut        ^T Execute  
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

per servizio https basta togliere l'asterisco, quindi attivare https



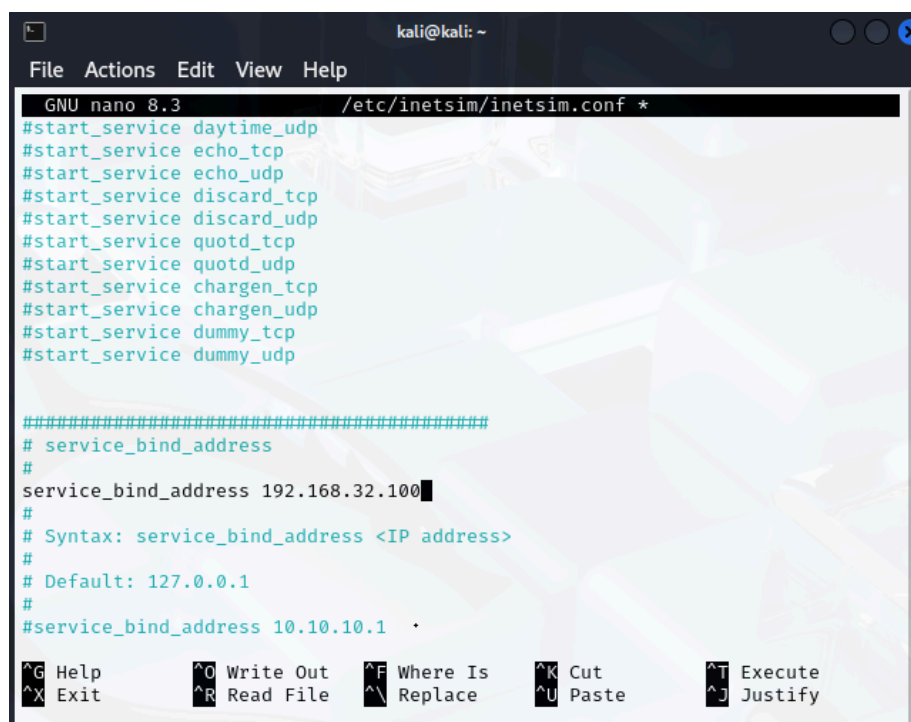
```
kali@kali: ~
File Actions Edit View Help
GNU nano 8.3 /etc/inetsim/inetsim.conf

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
start_service tftp
start_service irc
start_service ntp
start_service finger
start_service ident

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

per servizio http toglierlo su http



```
kali@kali: ~
File Actions Edit View Help
GNU nano 8.3 /etc/inetsim/inetsim.conf *

#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

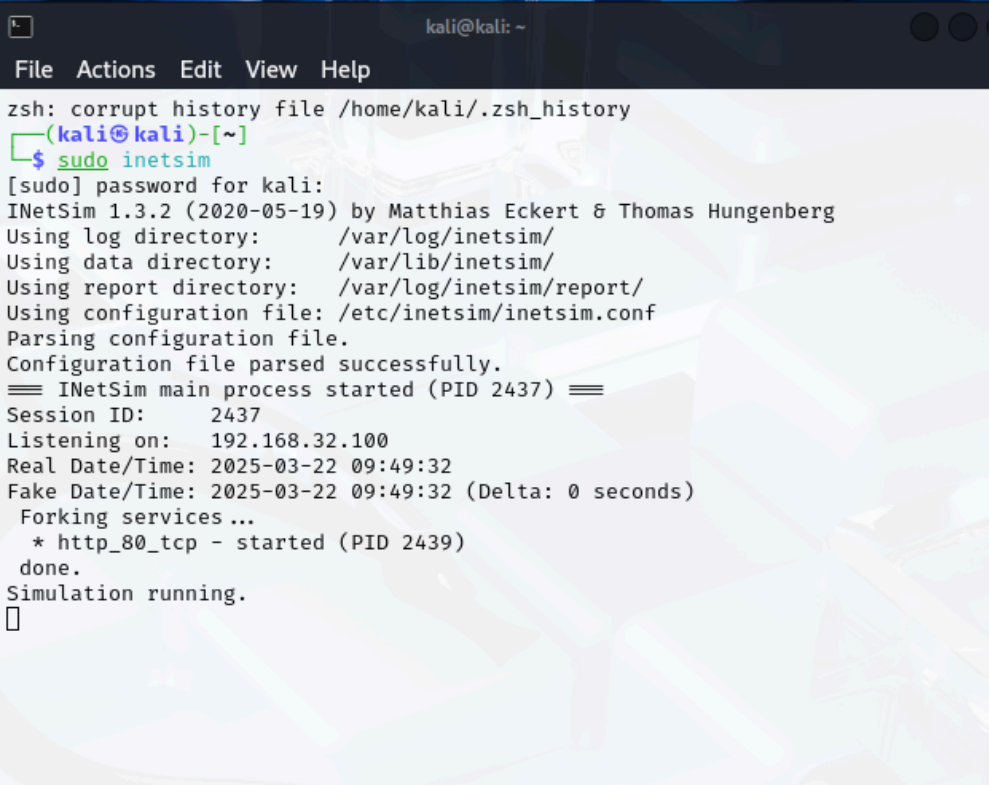
#####
# service_bind_address
#
service_bind_address 192.168.32.100
#
# Syntax: service_bind_address <IP address>
#
# Default: 127.0.0.1
#
#service_bind_address 10.10.10.1

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

per attivare il servizio bisogna bindare l'address all'ip, quindi di conseguenza inserire come da immagine

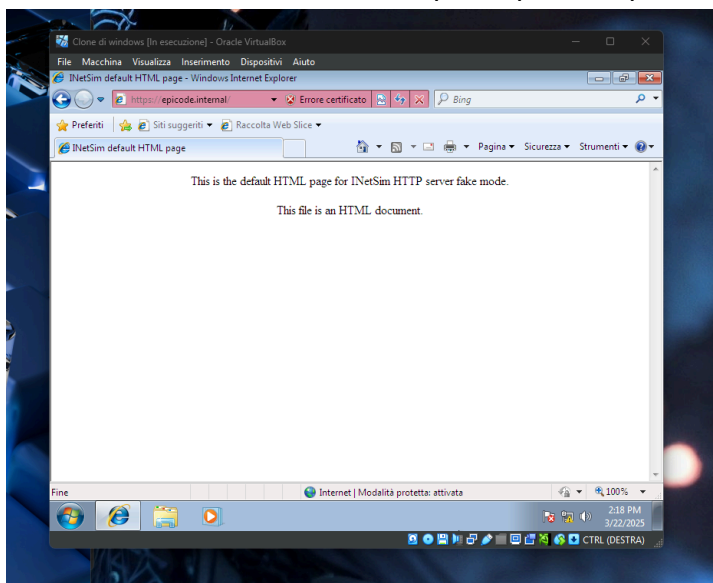
avviando inetsim da kali con

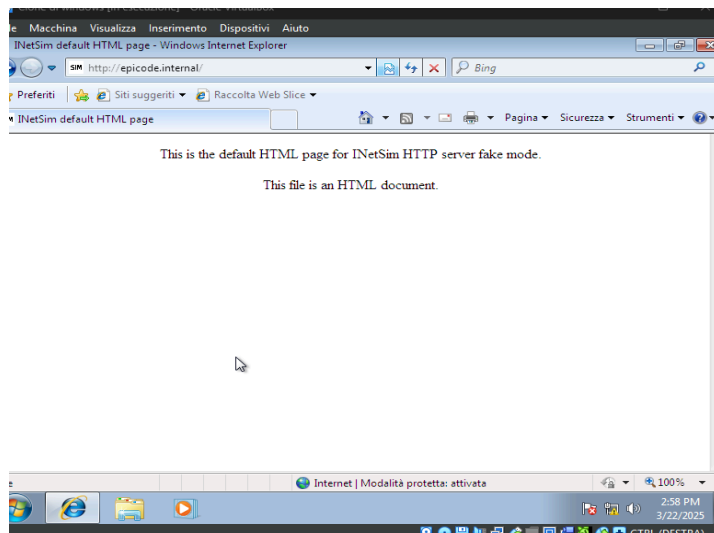
`sudo inetsim`

A terminal window on a Kali Linux system. The prompt is 'kali@kali: ~'. The user enters 'sudo inetsim'. The terminal shows the password prompt, then the inetsim version and authors. It lists the log, data, and report directories, and the configuration file. It confirms the configuration is parsed successfully and shows the main process starting with PID 2437. It also shows the session ID, listening IP (192.168.32.100), and the current date/time. Finally, it shows the services being forked, including http_80_tcp with PID 2439, and ends with 'Simulation running.'

```
kali@kali: ~  
File Actions Edit View Help  
zsh: corrupt history file /home/kali/.zsh_history  
(kali@kali)-[~]  
$ sudo inetsim  
[sudo] password for kali:  
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 2437) ==  
Session ID: 2437  
Listening on: 192.168.32.100  
Real Date/Time: 2025-03-22 09:49:32  
Fake Date/Time: 2025-03-22 09:49:32 (Delta: 0 seconds)  
Forking services ...  
* http_80_tcp - started (PID 2439)  
done.  
Simulation running.  
█
```

ora effettuiamo i test browser sia per http che https





come possiamo notare i test sono effettuati con successo . ora procediamo allo step successivo.

Step 5

Sniffing con wireshark

```
Section number: 1
▶ Interface id: 0 (eth0)
Encapsulation type: Ethernet (1)
Arrival Time: Mar 22, 2025 09:58:54.610592734 EDT
UTC Arrival Time: Mar 22, 2025 13:58:54.610592734 UTC
Epoch Arrival Time: 1742651934.610592734
[Time shift for this packet: 0.000000000 seconds]
[Time delta from previous captured frame: 0.000397658 sec]
[Time delta from previous displayed frame: 0.000397658 sec]
[Time since reference or first frame: 0.000978169 seconds]
Frame Number: 4
Frame Length: 333 bytes (2664 bits)
Capture Length: 333 bytes (2664 bits)
[Frame is marked: False]
[Frame is ignored: False]
[Protocols in frame: eth:ethertype:ip:tcp:http]
[Coloring Rule Name: HTTP]
```

```
▼ Address Resolution Protocol (Request)
  Hardware type: Ethernet (1)
  Protocol type: IPv4 (0x0800)
  Hardware size: 6
  Protocol size: 4
  Opcode: request (1)
  Sender MAC address: PCSSystemtec_6e:7a:00 (08:00:27:6e:7a:00)
  Sender IP address: 192.168.32.100
  Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
  Target IP address: 192.168.32.101
```

Step 6

Analisi generale

il protocollo https è criptato, quindi i pacchetti in transito quindi non è possibile scoprire il contenuto. le informazioni che invece da HTTP possono essere reperibili come ad esempio

- dominio
- ip
- porte utilizzate
- indirizzi MAC
- data e ora della richiesta dei pacchetti
- tipo di browser
- sistema operativo
- url completo
- contenuto dei messaggi