

PRESENTAZIONE PROGETTO EPICODE

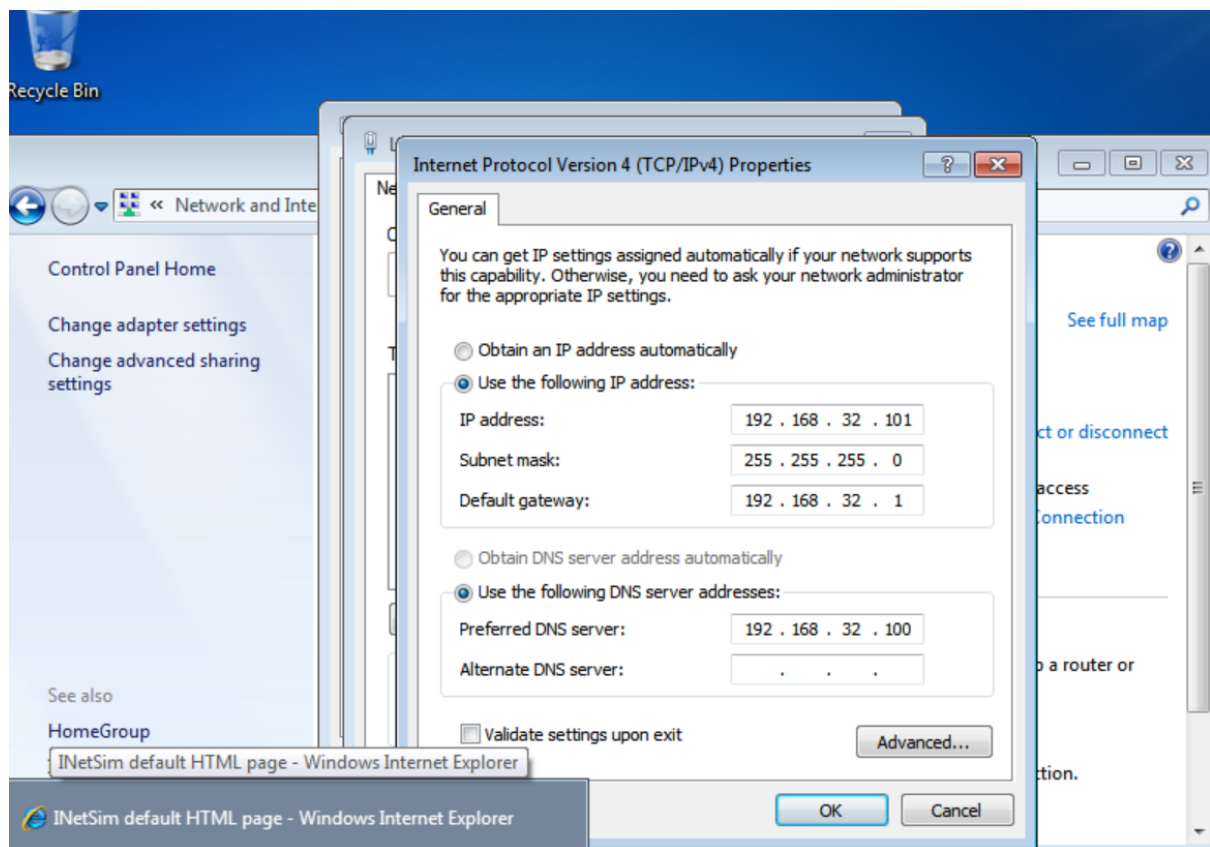
1- In questo passaggio andremo a cambiare Ip address alla macchina Kali-Linux

```
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
```

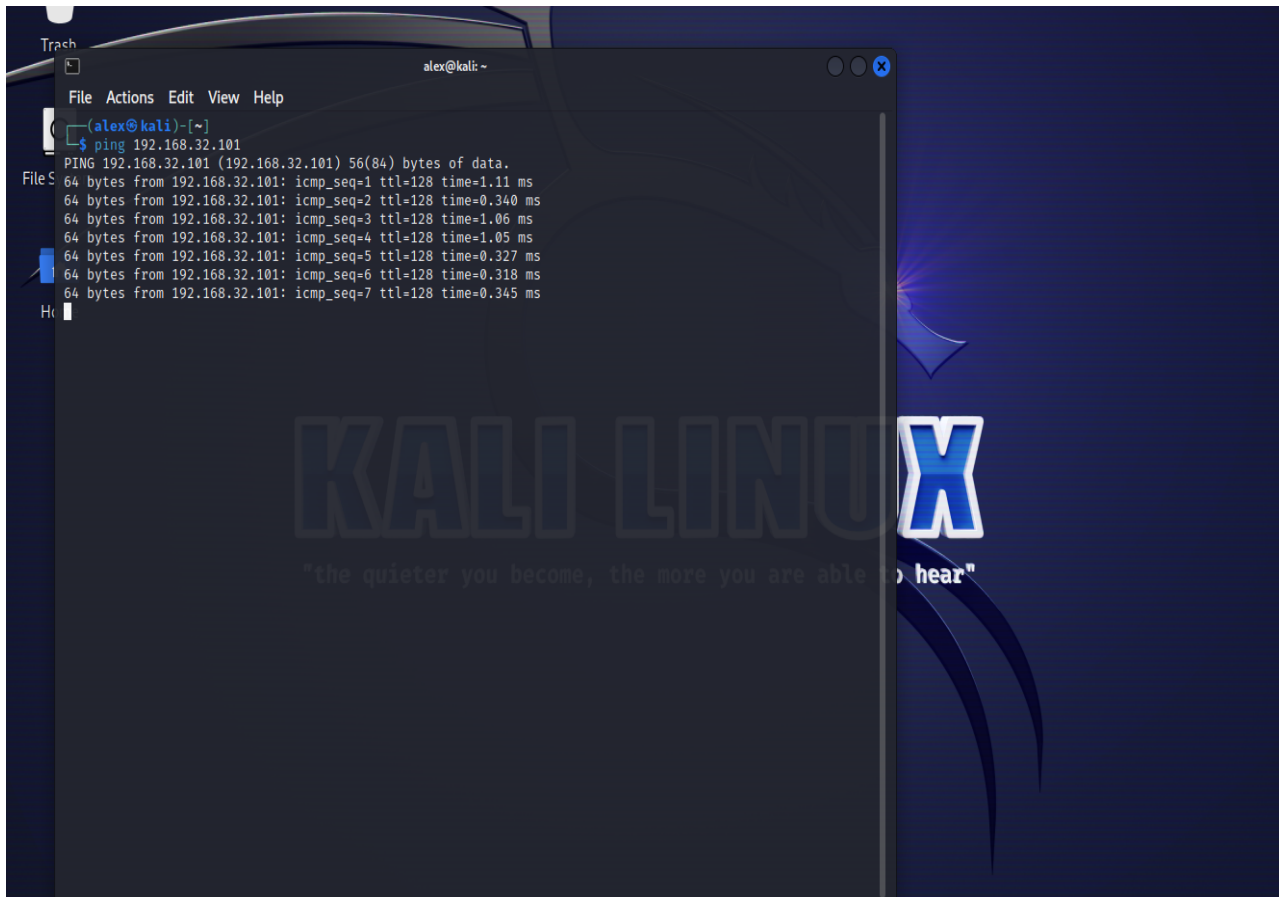
KALI LINUX
"the quieter you become, the more you are able to hear"

^G Help	^O Write Out	^W Where Is	[Read 15 lines]	^T Execute	^C Location	M-U Undo
^X Exit	^R Read File	^N Replace	^K Cut	^J Justify	^_ Go To Line	M-E Redo
			^U Paste			

2-Successivamente cambieremo anche Ip address alla macchina windows7



3- Adesso per dimostrare che le macchine comunicano tra di loro useremo il comando “ping” dal terminale kali-Linux , seguito dall’ IP address di windows7



The image shows a Kali Linux desktop environment with a terminal window open. The terminal window has a menu bar with 'File', 'Actions', 'Edit', 'View', and 'Help'. The prompt is '(alex@kali)~'. The user has entered the command 'ping 192.168.32.101'. The output shows a successful ping to 192.168.32.101 (192.168.32.101) with 56(84) bytes of data. The output includes seven lines of ping results, each showing 64 bytes from 192.168.32.101 with an icmp_seq and time. The times are: 1.11 ms, 0.340 ms, 1.06 ms, 1.05 ms, 0.327 ms, 0.318 ms, and 0.345 ms. The background of the desktop features the Kali Linux logo and the quote 'the quieter you become, the more you are able to hear'.

```
alex@kali: ~  
File Actions Edit View Help  
(alex@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=1.11 ms  
64 bytes from 192.168.32.101: icmp_seq=2 ttl=128 time=0.340 ms  
64 bytes from 192.168.32.101: icmp_seq=3 ttl=128 time=1.06 ms  
64 bytes from 192.168.32.101: icmp_seq=4 ttl=128 time=1.05 ms  
64 bytes from 192.168.32.101: icmp_seq=5 ttl=128 time=0.327 ms  
64 bytes from 192.168.32.101: icmp_seq=6 ttl=128 time=0.318 ms  
64 bytes from 192.168.32.101: icmp_seq=7 ttl=128 time=0.345 ms  
Hc
```

4-Dopo aver completato i passaggi precedenti andremo a selezionare (togliendo “#” che serve per commentare) tramite il tool inetSim le seguenti diciture: HTTP,HTTPS e Dns

```
# 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
```

[Read 1999 lines]

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

5- Associazione all'hostname epicode.internal che risponde all'indirizzo 192.168.32.100 (p.s. il seguente ip address sarà funzionare da server ovvero la macchina kali linux)

```
#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

#####
# service_bind_address "the quieter you become, the more you are able to hear"
#
# IP address to bind services to
#
# Syntax: service_bind_address <IP address>
#
# Default: 127.0.0.1
#
service_bind_address 192.168.32.100

#####
# service_run_as_user

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

```
# Default: www
#
#dns_default_hostname somehost

#####
# dns_default_domainname
#
# Default domain name to return with DNS replies
#
# Syntax: dns_default_domainname <domain name>
#
# Default: epicode.internal
#
dns_default_domainname epicode.internal

#####
# dns_static
#
# Static mappings for DNS
#
# Syntax: dns_static <fqdn hostname> <IP address>
#
# Default: none "the quieter you become, the more you are able to hear"
#
#
#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

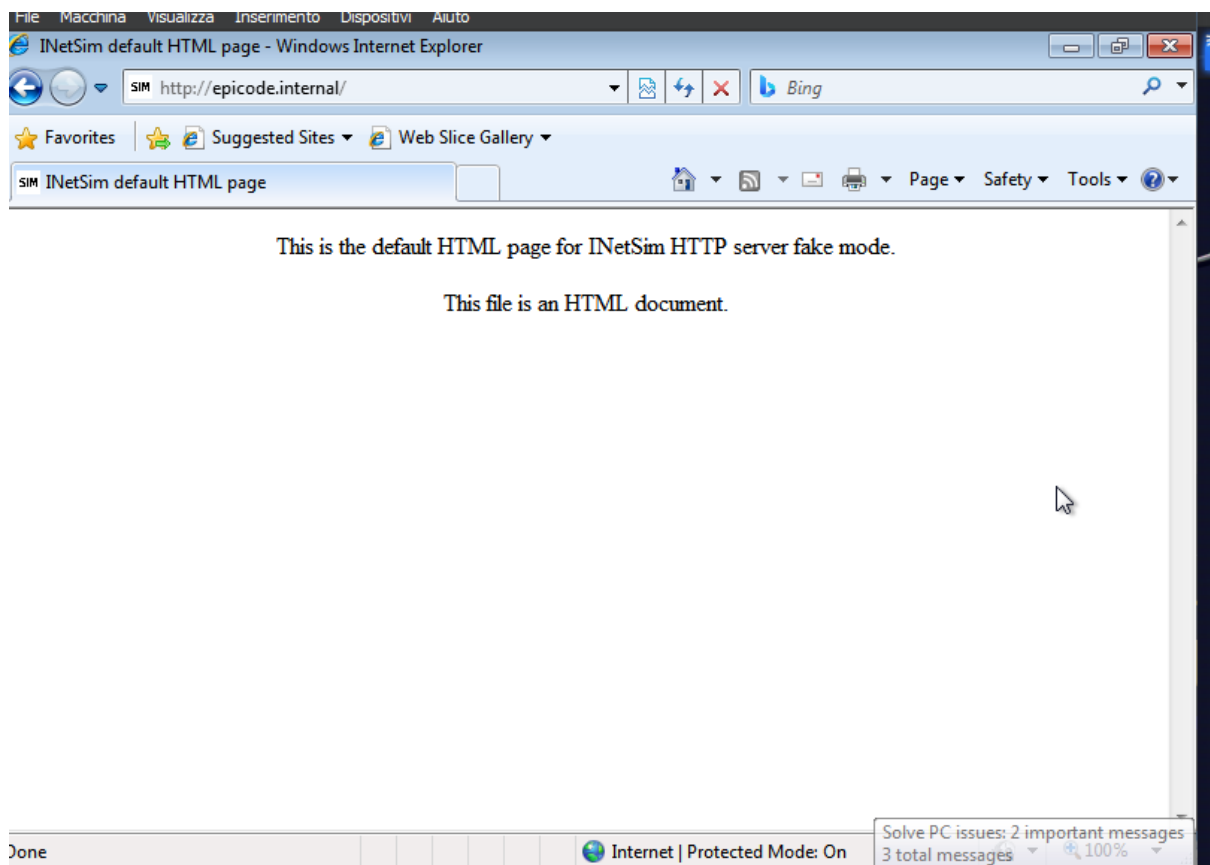
#####
# dns_version
#
# DNS version

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

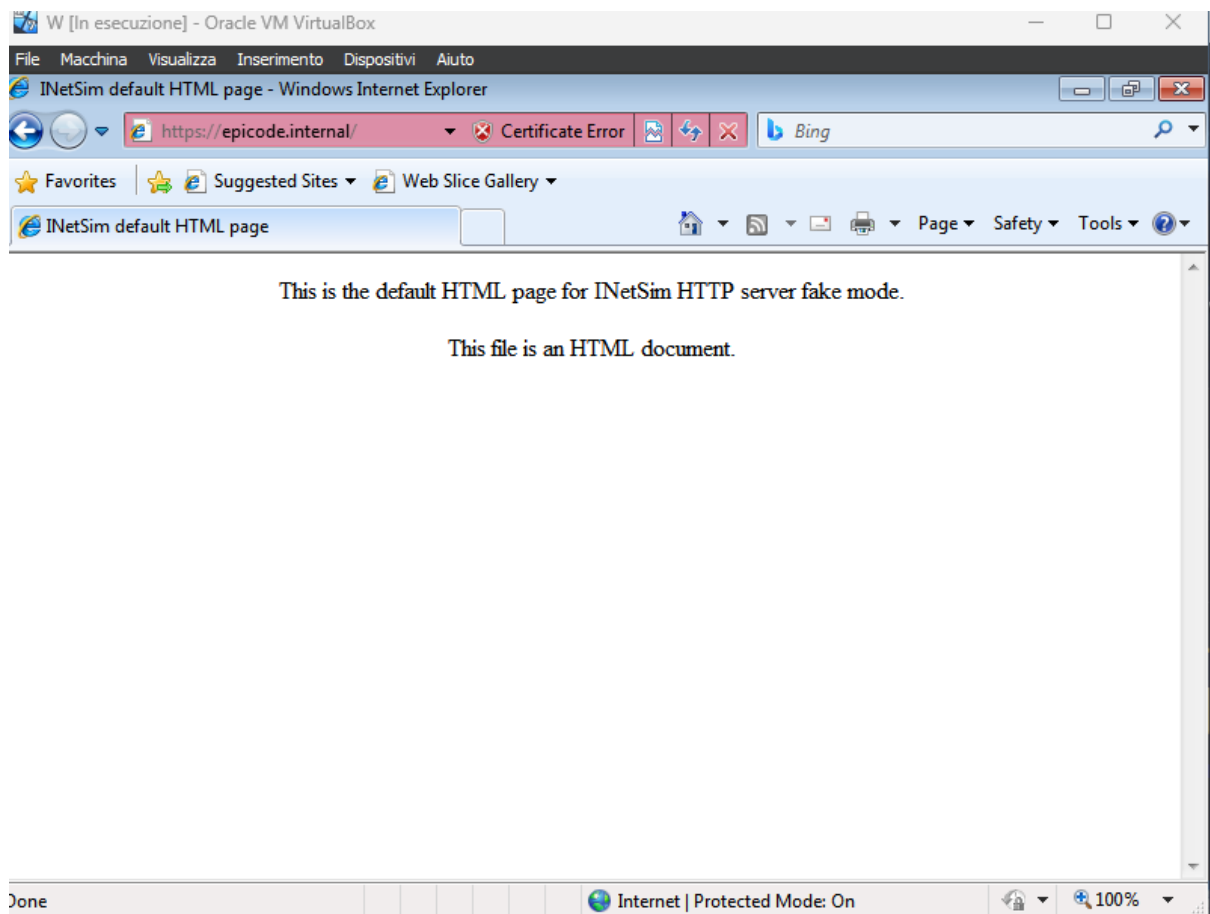
6-Andremo ad avviare il tool inetsim con il comando `sudo inetsim` (p.s. la dicitura “sudo “ serve per i permessi root della macchina)

```
alex@kali: ~  
File Actions Edit View Help  
alex@kali)-[~]  
$ sudo inetsim  
[sudo] password for alex:  
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 263  
Warning: Unknown option 'Service' in configuration file '/etc/inetsim/inetsim.conf' line 370  
Configuration file parsed successfully.  
≡ INetSim main process started (PID 1840) ≡  
Session ID: 1840  
Listening on: 192.168.32.100  
Real Date/Time: 2023-09-30 16:48:33  
Fake Date/Time: 2023-09-30 16:48:33 (Delta: 0 seconds)  
Forking services...  
* dns_53_tcp_udp - started (PID 1842)  
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 1843)  
* https_443_tcp - started (PID 1844)  
done.  
Simulation running.
```

7-Adesso andremo a provare il tutto sulla macchina windows7 (http)



HTTPS



8-Sussessivamente andremo a intercettare con il tool Wireshark i pacchetti .

Prima con http e si noterà che i pacchetti non saranno criptati come nell'HTTPS .Infatti sarranno leggibili

The image shows a Wireshark network traffic capture on the interface *eth0. The filter is set to ip.addr == 192.168.32.100. The packet list shows a sequence of packets: a SYN packet (No. 5), an ACK packet (No. 6), a GET request (No. 8), and an OK response (No. 13). The selected packet (No. 13) is an HTTP 200 OK response. The packet details pane shows the structure of the HTTP response, including the status line (200 OK) and the body content. The packet bytes pane shows the raw data of the response body, which is HTML text. The status bar at the bottom indicates that 93 packets were captured, and 10 (10.8%) are displayed.

No.	Time	Source	Destination	Protocol	Length	Info
5	2.498646768	192.168.32.101	192.168.32.100	TCP	66	49204 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SAC
6	2.498664485	192.168.32.100	192.168.32.101	TCP	66	80 → 49204 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1
7	2.498987788	192.168.32.101	192.168.32.100	TCP	60	49204 → 80 [ACK] Seq=1 Ack=1 Win=65700 Len=0
8	2.498987835	192.168.32.101	192.168.32.100	HTTP	338	GET / HTTP/1.1
9	2.499027730	192.168.32.100	192.168.32.101	TCP	54	80 → 49204 [ACK] Seq=1 Ack=285 Win=64128 Len=0
10	2.507084599	192.168.32.100	192.168.32.101	TCP	204	80 → 49204 [PSH, ACK] Seq=1 Ack=285 Win=64128 Len=150
13	2.508646593	192.168.32.100	192.168.32.101	HTTP	312	HTTP/1.1 200 OK (text/html)
14	2.509068639	192.168.32.101	192.168.32.100	TCP	60	49204 → 80 [ACK] Seq=285 Ack=410 Win=65292 Len=0
15	2.509068676	192.168.32.101	192.168.32.100	TCP	60	49204 → 80 [FIN, ACK] Seq=285 Ack=410 Win=65292 Len=0
16	2.509084908	192.168.32.100	192.168.32.101	TCP	54	80 → 49204 [ACK] Seq=410 Ack=286 Win=64128 Len=0

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</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: 93 · Displayed: 10 (10.8%) Profile: Default

```
* dns_53_tcp_udp - started (PID 35291)\nprint() on closed filehandle MLOG at /usr/share/perl5/Net/DNS/Nameserver.pm line 399.\nprint() on closed filehandle MLOG at /usr/share/perl5/Net/DNS/Nameserver.pm line 399.\n* http_80_tcp - started (PID 35292)\n* https_443_tcp - started (PID 35293)\ndone.\nSimulation running.\n
```

Https (criptato) non leggibili

Wireshark network traffic capture showing an HTTP GET request and response. The packet list shows a GET request for /text/html. The packet details show the HTML content of the response, which is a default page for INetSim. The packet bytes show the raw data of the response.

No.	Time	Source	Destination	Protocol	Length	Info
5	2.498646768	192.168.32.101	192.168.32.100	TCP	66	49204 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SAC
6	2.498664485	192.168.32.100	192.168.32.101	TCP	66	80 → 49204 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1
7	2.498987788	192.168.32.101	192.168.32.100	TCP	60	49204 → 80 [ACK] Seq=1 Ack=1 Win=65700 Len=0
8	2.498987835	192.168.32.101	192.168.32.100	HTTP	338	GET / HTTP/1.1
9	2.499027730	192.168.32.100	192.168.32.101	TCP	54	80 → 49204 [ACK] Seq=1 Ack=285 Win=64128 Len=0
10	2.507084599	192.168.32.100	192.168.32.101	TCP	204	80 → 49204 [PSH, ACK] Seq=1 Ack=285 Win=64128 Len=150 [
13	2.508646593	192.168.32.100	192.168.32.101	HTTP	312	HTTP/1.1 200 OK (text/html)
14	2.509068639	192.168.32.101	192.168.32.100	TCP	60	49204 → 80 [ACK] Seq=285 Ack=410 Win=65292 Len=0
15	2.509068676	192.168.32.101	192.168.32.100	TCP	60	49204 → 80 [FIN, ACK] Seq=285 Ack=410 Win=65292 Len=0
16	2.509084908	192.168.32.100	192.168.32.101	TCP	54	80 → 49204 [ACK] Seq=410 Ack=286 Win=64128 Len=0

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

Frame (312 bytes) Reassembled TCP (408 bytes)

0000 08 00 27 60 6a 9e 08 00 27 8d 45 28 08 00 45 00 ...j... 'E(...
0010 01 2a 61 53 40 00 40 06 16 61 c0 a8 20 64 c0 a8 ...aS@ @. a.. d
0020 20 65 00 50 c0 34 28 cf 3d 06 71 38 71 5b 50 19 ...e P.4(= q8q[
0030 01 f5 c3 36 00 00 3c 68 74 6d 6c 3e 0a 20 20 3c ...6 <h tml>.
0040 68 65 61 64 3e 0a 20 20 20 20 3c 74 69 74 6c 65 head>... <tit
0050 3e 49 4e 65 74 53 69 6d 20 64 65 66 61 75 6c 74 >INetSim defau
0060 20 48 54 4d 4c 20 70 61 67 65 3c 2f 74 69 74 6c HTML pa ge</ti
0070 65 3e 0a 20 20 3c 2f 68 65 61 64 3e 0a 20 20 3c e>... </h ead>.
0080 62 6f 64 79 3e 0a 20 20 20 20 3c 70 3e 3c 2f 70 body>... <p><
0090 3e 0a 20 20 20 20 3c 70 20 61 6c 69 67 6e 3d 22 >... <p align

dns_53_tcp_udp - started (PID 35291)
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 35292)
* https_443_tcp - started (PID 35293)
done.
Simulation running.
[]

9-Infine come ultimo passaggio vedremo i mac-address delle macchine ,sia mittente che destinatario

The image shows a Wireshark network traffic capture. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The toolbar contains various icons for packet capture and analysis. The packet list shows a table of captured packets with columns for No., Time, Source, Destination, Protocol, Length, and Info. The packet details pane shows the selected packet's structure, including Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol. The packet bytes pane shows the raw data of the selected packet. The terminal window at the bottom displays the output of a simulation, including the start of dns_53_tcp_udp, http_80_tcp, and https_443_tcp, and the end of the simulation.

No.	Time	Source	Destination	Protocol	Length	Info
92	44.324356428	PcsCompu_60:6...	Broadcast	ARP	60	Who has 192.168.32.1? Tell 192.168.32.101
93	45.536581548	192.168.1.56	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1
94	45.537806347	192.168.1.56	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
95	45.916306214	SernetSu_3e:a...	Broadcast	ARP	60	Who has 192.168.1.56? Tell 192.168.1.254
96	46.539092537	192.168.1.56	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
97	46.539092727	192.168.1.56	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1
98	47.540484476	192.168.1.56	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1
99	47.540485006	192.168.1.56	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1
100	48.540875045	192.168.1.56	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1
101	48.540875584	192.168.1.56	239.255.255.250	SSDP	217	M-SEARCH * HTTP/1.1

Frame 54: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface eth0
Ethernet II, Src: PcsCompu_8d:45:28 (08:00:27:8d:45:28), Dst: PcsCompu_60:6a:9e (08:00:27:60:6a:9e)
Destination: PcsCompu_60:6a:9e (08:00:27:60:6a:9e)
Source: PcsCompu_8d:45:28 (08:00:27:8d:45:28)
Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 192.168.32.100, Dst: 192.168.32.101
Transmission Control Protocol, Src Port: 443, Dst Port: 49199, Seq: 300000000, Win: 65535, Len: 0

Ethernet (eth), 14 byte(s) Packets: 101 - Displayed: 101 (100.0%) Profile: Default

```
* dns_53_tcp_udp - started (PID 35291)
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 35292)
* https_443_tcp - started (PID 35293)
done.
Simulation running.
```