

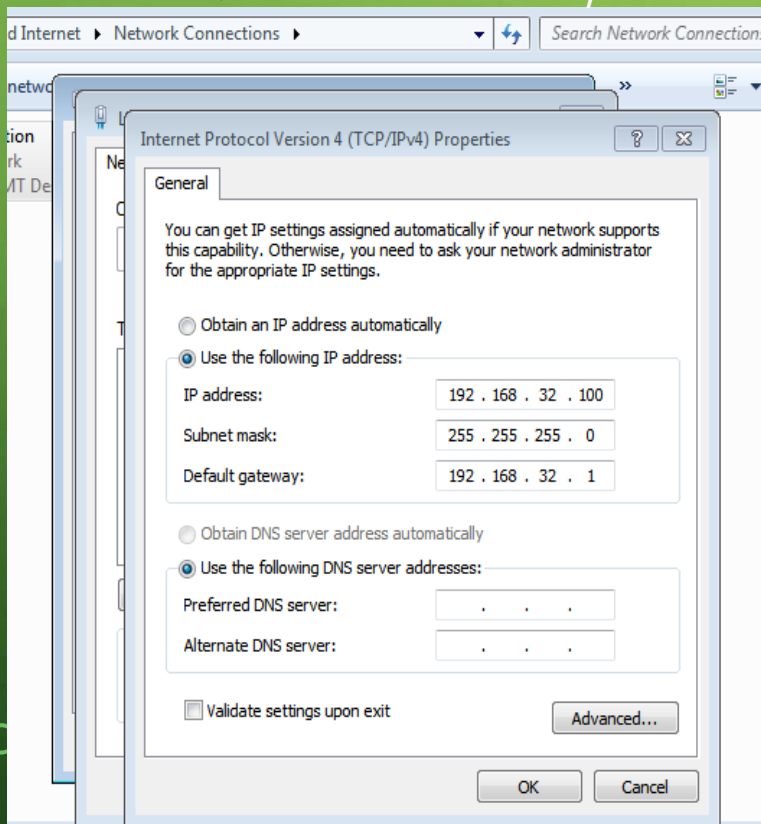
A decorative graphic on the left side of the slide, consisting of a network of thin, light green lines and small circles, resembling a circuit board or a neural network, extending from the top and bottom edges towards the center.

SIMULAZIONE CLIENT/SERVER CON WINDOWS 7 E KALI LINUX

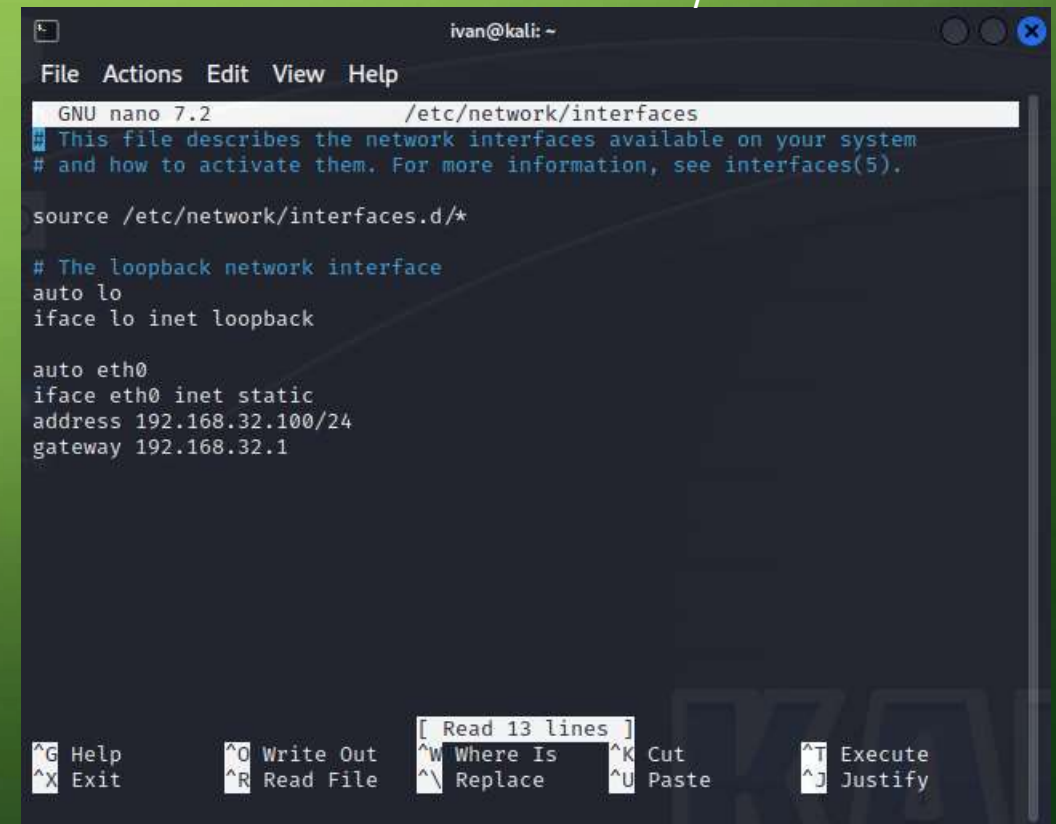
UTILIZZANDO VIRTUAL MACHINES

SI INIZIA CONFIGURANDO GLI INDIRIZZI IP DI WIN7 (CHE VERRÀ UTILIZZATO COME CLIENT) E KALI LINUX (CHE SERVIRÀ DA CLIENT)

WIN7: 192.168.32.100/24



KALI: 192.168.32.100/24



I DUE DEVICES SI FARANNO PINGARE A VICENDA PER CONTROLLARE SE COMUNICANO

COMANDO **PING** DA WIN7 A KALI LINUX

192.168.32.101 → 192.168.32.100

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

C:\Users\Ivan>ping 192.168.32.100

Pinging 192.168.32.100 with 32 bytes of data:
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
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 = 0ms, Average = 0ms

C:\Users\Ivan>
```

COMANDO **PING** DA KALI LINUX A WIN7

192.168.32.100 → 192.168.32.101

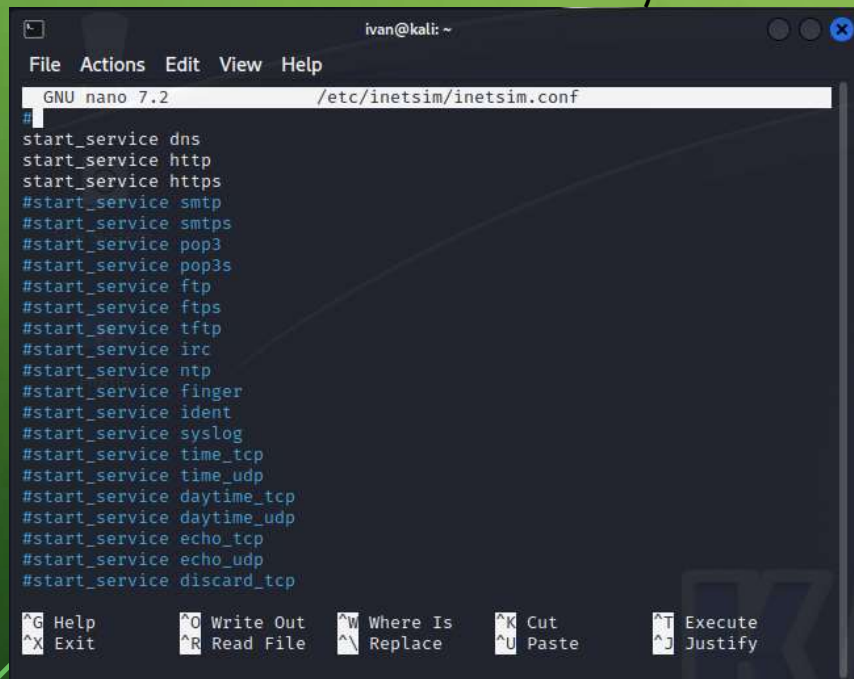
```
ivan@kali: ~
File Actions Edit View Help

(ivan@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=0.329 ms
64 bytes from 192.168.32.101: icmp_seq=2 ttl=128 time=0.255 ms
64 bytes from 192.168.32.101: icmp_seq=3 ttl=128 time=0.262 ms
64 bytes from 192.168.32.101: icmp_seq=4 ttl=128 time=0.286 ms
^C
— 192.168.32.101 ping statistics —
4 packets transmitted, 4 received, 0% packet loss, time 3066ms
rtt min/avg/max/mdev = 0.255/0.283/0.329/0.028 ms

(ivan@kali)-[~]
$
```

VERRÀ AVVIATO **INETSIM**, UN SOFTWARE DI KALI LINUX UTILE A SIMULARE I SERVIZI INTERNET IN UN LABORATORIO VIRTUALE, E QUEST'ULTIMO VERRÀ IMPOSTATO COME SERVER

SI LASCERANNO ATTIVI I SERVIZI **DNS**, **HTTP** E **HTTPS** (QUESTI ULTIMI DUE IN BASE ALLE NECESSITÀ)

A terminal window titled 'ivan@kali: ~' showing the configuration of the inetsim service. The window is running GNU nano 7.2 editing /etc/inetsim/inetsim.conf. The configuration file contains a list of services to be started, with 'dns', 'http', and 'https' being the primary focus. The list includes: 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, and #start_service discard_tcp. The bottom of the window shows nano editor shortcuts: ^G Help, ^X Exit, ^O Write Out, ^R Read File, ^W Where Is, ^L Replace, ^K Cut, ^U Paste, ^T Execute, and ^J Justify.

```
ivan@kali: ~
File Actions Edit View Help
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
^G Help  ^O Write Out  ^W Where Is  ^K Cut      ^T Execute
^X Exit  ^R Read File  ^L Replace   ^U Paste    ^J Justify
```

SUCCESSIVAMENTE SI IMPOSTERÀ L'IP DI KALI COME SERVER E VERRÀ ASSOCIATO AL DNS «**EPICODE.INTERNAL**»

A terminal window titled 'ivan@kali: ~' showing the configuration of static DNS mappings in the inetsim.conf file. The window is running GNU nano 7.2 editing /etc/inetsim/inetsim.conf. The configuration includes a section for static mappings with the following lines: # dns_static, # Static mappings for DNS, # Syntax: dns_static <fqdn hostname> <IP address>, # Default: none, and three specific mappings: dns_static epicode.internal 192.168.32.100, dns_static ns1.foo.com 10.70.50.30, and dns_static ftp.bar.net 10.10.20.30. The bottom of the window shows nano editor shortcuts: ^G Help, ^X Exit, ^O Write Out, ^R Read File, ^W Where Is, ^L Replace, ^K Cut, ^U Paste, ^T Execute, and ^J Justify.

```
ivan@kali: ~
File Actions Edit View Help
GNU nano 7.2 /etc/inetsim/inetsim.conf
#
# Default:
#
#dns_default_domainname some.domain

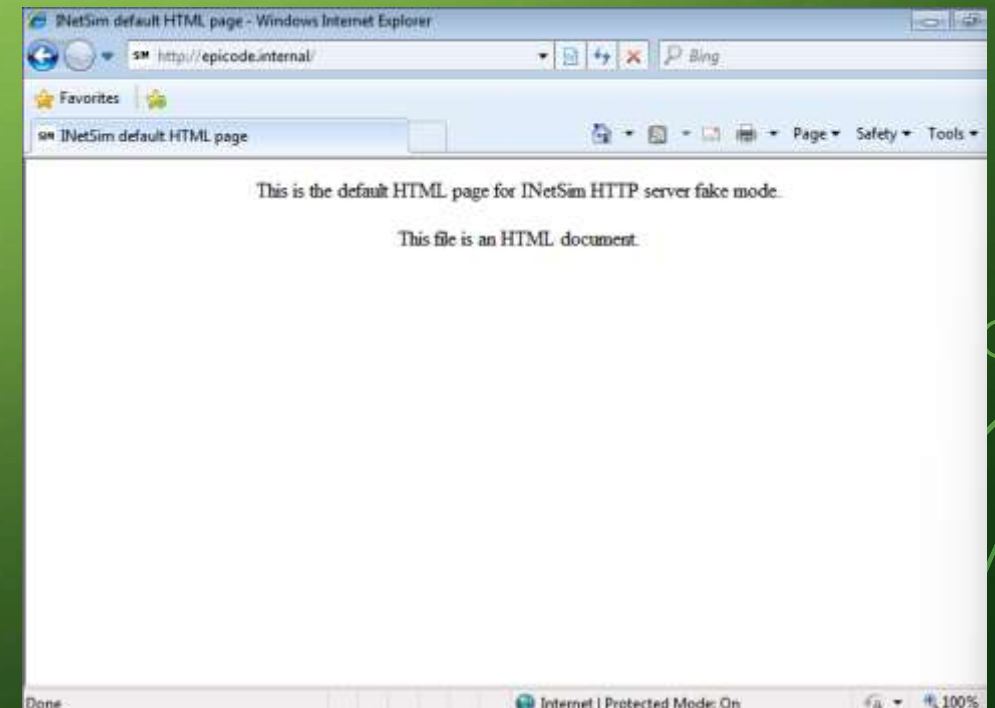
#####
# dns_static
#
# Static mappings for DNS
#
# Syntax: dns_static <fqdn hostname> <IP address>
#
# Default: none
#
dns_static epicode.internal 192.168.32.100
dns_static ns1.foo.com 10.70.50.30
dns_static ftp.bar.net 10.10.20.30
#
#####
# dns_version
#
# DNS version
#
# Syntax: dns_version <version>
#
# Default: "INetSim DNS Server"
#
^G Help  ^O Write Out  ^W Where Is  ^K Cut      ^T Execute  ^L Location
^X Exit  ^R Read File  ^L Replace   ^U Paste    ^J Justify  ^G Go To Line
```


FINITA LA CONFIGURAZIONE, VERRÀ AVVIATO INETSIM E SI PROVERÀ AD ACCEDERE ALL'IP **192.168.32.100** TRAMITE IL DNS «**EPICODE.INTERNAL**»

AVVIO DI INETSIM TRAMITE COMANDO
«**SUDO INETSIM**»

```
ivan@kali: ~  
File Actions Edit View Help  
└─$ sudo inetsim  
[sudo] password for ivan:  
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 261  
Configuration file parsed successfully.  
== INetSim main process started (PID 38856) ==  
Session ID: 38856  
Listening on: 192.168.32.100  
Real Date/Time: 2023-09-29 14:10:40  
Fake Date/Time: 2023-09-29 14:10:40 (Delta: 0 seconds)  
Forking services ...  
* dns_53_tcp_udp - started (PID 38866)  
* http_80_tcp - started (PID 38867)  
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.  
* https_443_tcp - started (PID 38868)  
done.  
Simulation running.  
█
```

WIN7, TRAMITE IL PROPRIO BROWSER, AVRÀ
ACCESSO A QUELL'IP, MOSTRANDO LA
PAGINA HTML DI INETSIM



INFINE SI UTILizzerÀ **WIRESHARK** PER INTERCETTARE LA COMUNICAZIONE TRA CLIENT E SERVER

INIZIALMENTE, IMPOSTANDOSTANDO **HTTP** COME PROTOCOLLO DI TRASFERIMENTO, SI
NOTERÀ COME SIA POSSIBILE VISUALIZZARE TUTTE LE INFORMAZIONI CONTENUTE NEL
PACCHETTO, TRA CUI IL **MESSAGGIO** ED I **MAC** DI CLIENT E SERVER

The image shows a Wireshark network traffic capture. The top pane displays a list of network packets. The middle pane shows the details of the selected packet (No. 71), including Ethernet II, Internet Protocol Version 4, Transmission Control Protocol, and Hypertext Transfer Protocol. The bottom pane shows the raw packet data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
65	0.227970168	192.168.32.101	192.168.32.100	TCP	60	49231 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM
66	0.228004352	192.168.32.100	192.168.32.101	TCP	60	80 → 49231 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
67	0.228227760	192.168.32.101	192.168.32.100	TCP	60	49231 → 80 [ACK] Seq=1 Ack=1 Win=65700 Len=0
68	0.228339953	192.168.32.101	192.168.32.100	HTTP	344	GET / HTTP/1.1
69	0.228343317	192.168.32.100	192.168.32.101	TCP	54	80 → 49231 [ACK] Seq=1 Ack=291 Win=64128 Len=0
70	0.239164484	192.168.32.100	192.168.32.101	TCP	284	80 → 49231 [PSH, ACK] Seq=1 Ack=291 Win=64128 Len=156 [TCP segment of a reassembled PDU]
71	0.241255402	192.168.32.100	192.168.32.101	HTTP	112	HTTP/1.1 200 OK (text/html)
72	0.241462831	192.168.32.101	192.168.32.100	TCP	60	49231 → 80 [ACK] Seq=291 Ack=418 Win=65292 Len=0
73	0.241701178	192.168.32.101	192.168.32.100	TCP	60	49231 → 80 [FIN, ACK] Seq=291 Ack=418 Win=65292 Len=0
74	0.241728068	192.168.32.100	192.168.32.101	TCP	54	80 → 49231 [ACK] Seq=418 Ack=292 Win=64128 Len=0

Frame 71: 312 bytes on wire (2496 bits), 312 bytes captured (2496 bits) on interface eth0, id 0
Ethernet II, Src: PcsCompu_f4:b2:fc (08:00:27:f4:b2:fc), Dst: PcsCompu_14:ad:78 (08:00:27:14:ad:78)
Internet Protocol Version 4, Src: 192.168.32.100, Dst: 192.168.32.101
Transmission Control Protocol, Src Port: 80, Dst Port: 49231, Seq: 151, Ack: 291, Len: 268
[2 Reassembled TCP Segments (408 bytes): #70(150), #71(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>\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

0000 08 00 27 14 ad 78 08 00 27 f4 b2 fc 08 00 45 00 . . p E
0010 01 2a bb bc 40 00 40 00 b0 f7 c0 a8 20 04 c0 a8 d .
0020 20 65 00 50 c0 4f 36 b1 6f 4f 0f e0 48 0f 50 19 e P 00 o0o M P
0030 01 f5 c3 36 00 00 3c 68 74 6d 6c 3e 0a 20 20 3c 6 . <h tml> <
0040 08 05 61 64 3e 0a 20 20 20 20 3c 74 69 74 6c 05 head> <title
0050 3e 49 4e 65 74 53 69 6d 20 64 65 66 01 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> </h ead> <
0080 02 0f 64 79 3e 0a 20 20 20 20 3c 70 3e 3c 2f 70 body> <p></p
0090 3e 8a 20 20 20 20 3c 70 20 61 6c 69 67 6e 3d 22 > <p align="ce
00a0 63 65 6e 74 65 72 22 3e 54 68 69 73 28 69 73 29 center"> This is
00b0 74 68 65 29 64 65 66 61 75 6c 74 20 48 54 4d 4c the defa ult HTML
00c0 20 70 61 67 65 28 66 6f 72 28 48 4e 65 74 53 69 page fo r InetSi
00d0 6d 20 48 54 54 58 20 73 65 72 76 65 72 20 66 61 m HTTP s erve r fa
00e0 6b 65 20 6d 6f 64 65 2e 3c 2f 70 3e 8a 20 20 29 ke mode. </p>
00f0 20 3c 70 20 61 6c 69 67 6e 3d 22 63 65 6e 74 65 <p align="cente
0100 72 22 3e 54 68 69 73 20 66 69 6c 65 29 69 73 20 r">This file is
0110 61 6e 20 48 54 4d 4c 20 64 6f 63 75 6d 65 6e 74 an HTML document
0120 2e 3c 2f 70 3e 0a 20 20 3c 2f 62 6f 64 79 3e 8a .</p> </body>
0130 3c 2f 68 74 6d 6c 3e 0a </html>

Frame (312 bytes) Reassembled TCP (408 bytes)

eth0: <live capture in progress> Packets: 309 - Displayed: 10 (3.2%) Profile: Default

COL PROTOCOLLO **HTTPS** SI PUÒ INVECE NOTARE UNA DIFFERENZA IMPORTANTE

CON IL PROTOCOLLO **HTTPS** SI NOTERÀ INVECE COME GLI INDIRIZZI MAC SIANO
UGUALMENTE VISIBILI, MA IL MESSAGGIO VERRÀ **CRIPTATO**, RENDENDOLO «ILLEGIBILE»

Wireshark packet capture showing an HTTPS transaction. The packet list shows a TLSv1 record (frame 138) with encrypted application data. The packet details pane shows the TLSv1 record structure, including the encrypted application data. The packet bytes pane shows the raw data of the encrypted application data, which is illegible.

Frame 138: 379 bytes on wire (3032 bits), 379 bytes captured (3032 bits) on interface eth0, id 0

- Ethernet II, Src: PcsCompu_14:ad:70 (08:00:27:14:ad:70), Dst: PcsCompu_f4:b2:fc (08:00:27:f4:b2:fc)
- Internet Protocol Version 4, Src: 192.168.32.101, Dst: 192.168.32.100
- Transmission Control Protocol, Src Port: 49224, Dst Port: 443, Seq: 389, Ack: 1379, Len: 325
- Transport Layer Security
 - TLSv1 Record Layer: Application Data Protocol: Hypertext Transfer Protocol
 - Content Type: Application Data (23)
 - Version: TLS 1.0 (0x0301)
 - Length: 320
 - Encrypted Application Data: dc9462aabd3c13d7a19511a0e3c41f01c4770b95ca863b7e0b5a7ad5fce99bf97b4112a

The packet bytes pane shows the raw data of the encrypted application data, which is illegible.