

FINE MODULO

GIANMARCO ELIA

Impostare IP statico su Kali Linux (192.168.32.100)

Sudo nano /etc/network/interfaces

Auto eth0

Iface eth0 inet static

Address 192.168.32.100

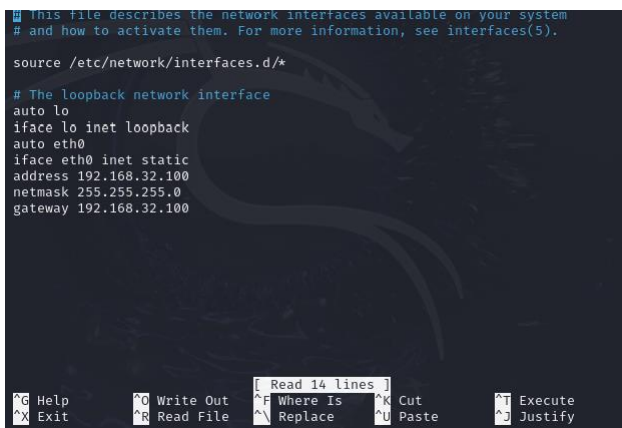
Netmask 255.255.255.0

Gateway 192.168.32.1

Successivamente premere CTRL+ o + invio

CTRL + x + invio

Sudo reboot per riavviare Kali



```
# 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
netmask 255.255.255.0
gateway 192.168.32.1
```

Attivazione servizi DNS, http, HTTPS su Kali Linux

Per l'attivazione dei servizi: HTTP, HTTPS e DNS, utili a richiamare le pagine web da Windows 7, è necessario configurare il servizio "inetSim" da Kali Linux:

Sudo nano /etc/inetsim/inetsim.conf

Per attivare i servizi richiesti dalla traccia bisogna aggiungere "#" quelli non richiesti. In questo caso basta attivare i servizi che si vedono in foto:

```
# 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 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
```

^G Help ^O Write Out ^F Where
 ^X Exit ^R Read File ^\ Repl

Per associare un IP ad inetSim è necessario togliere il cancelletto da “service_bind_address”. In questo caso si associa l’indirizzo 0.0.0.0 in modo da permettere all’applicazione di comunicare a tutti gli IP disponibili in base alle schede di rete configurate.

```
#start_service chargen_tcp
#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 0.0.0.0

#####
# service_run_as_user
#
# User to run services
#
```

```
#
# Default domain name to return with DNS replies
#
# Syntax: dns_default_domainname <domain name>
#
# Default: inetsim.org
#
#dns_default_domainname domain.name

#####
# dns_static
#
# Static mappings for DNS
#
# Syntax: dns_static <fqdn hostname> <IP address>
#
# Default: none
#
#dns_static www.foo.com 10.10.10.10
#dns_static ns1.foo.com 10.70.50.30
#dns_static epicode.internal 192.168.32.100
```

Per attivare il servizio DNS statico è utile cancellare il cancelletto da “dns_static”, inserendo la richiesta della traccia del fine modulo, perchè la chiamata deve essere fatta su “epicode.internal” relativo all’indirizzo IP 192.168.32.100 che sarà coincidente all’indirizzo associato su Kali, perchè dovrà fungere da server application.

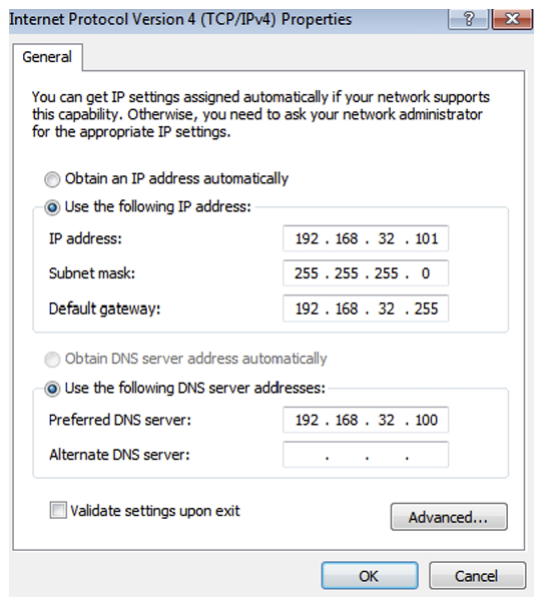
```
(kali@kali)~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.32.100 netmask 255.255.255.0 broadcast 192.168.32.255
    inet6 fe80::a00:27ff:fe86:72ff prefixlen 64 scopeid 0<link>
    ether 08:00:27:86:72:ff txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 17 bytes 2494 (2.4 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 8 bytes 480 (480.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8 bytes 480 (480.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Con la configurazione dei servizi, è possibile verificare la corretta esecuzione dei comandi scrivendo il comando “ifconfig”.

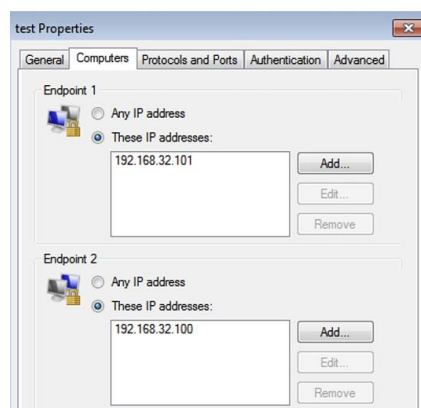
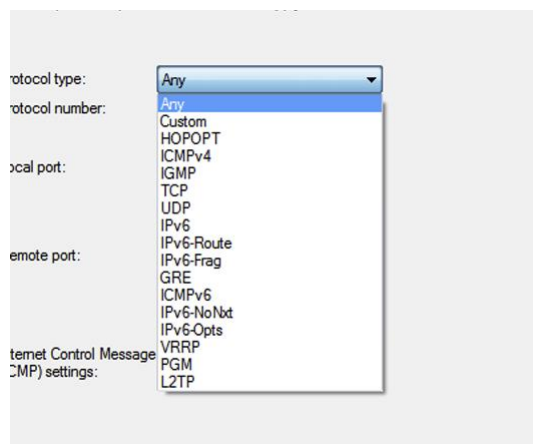
Impostare Ip Statico Windows 7 (192.168.32.101) (non ho usato Windows 10 perchè andava lento)

1. *Pannello di controllo*
2. *Network and Internet*
3. *Change adapter settings*
4. *(Tasto destro) Local Area Connection*
5. *Properties*
6. *Internet protocol Version 4 (TCP/IPv4)*
7. *Properties*: si apre la finestra di configurazione e si inserisce nella voce IP Address l’IP statico da associare, il default Gateway e l’indirizzo IP statico associato al DNS su Kali Linux per permettere a Windows 7 di collegarsi.



Firewall Windows

1. *Control Panel*
2. *System and Security*
3. *Advanced settings ->*
4. *Impostare inbound e outbound connection*



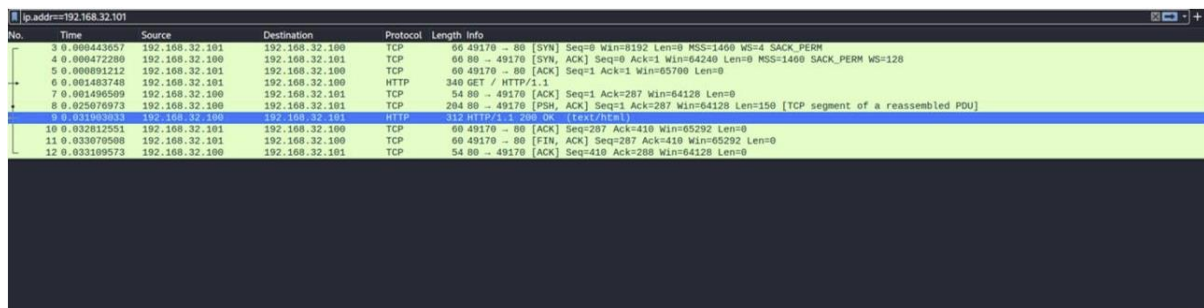
Avvio inetSim

Per l'avvio di HTTP, HTTPS e DNS digitare sul terminale di Kali Linux

Sudo inetsim

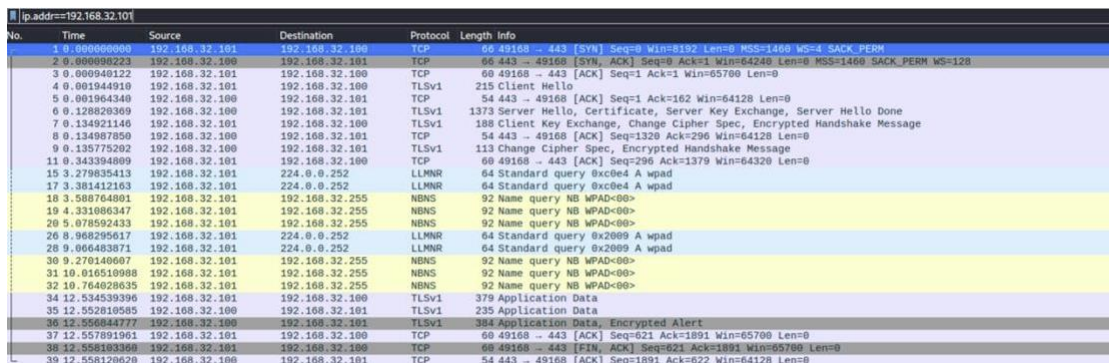
Intercettazione comunicazione con WireShark

Aprire WireShark su Kali Linux e avviare la registrazione del traffico e successivamente andare su Windows 7 ed fare la chiamata, da browser, HTTP. Dopo questa procedura WireShark registrerà il traffico di rete. Si nota che il dispositivo che chiama ha indirizzo IP 192.168.32.101 mentre il dispositivo che risponde ha come indirizzo IP 192.168.32.100. Si nota che si legge una chiamata HTTP “GET” e la risposta del server “200 OK”.



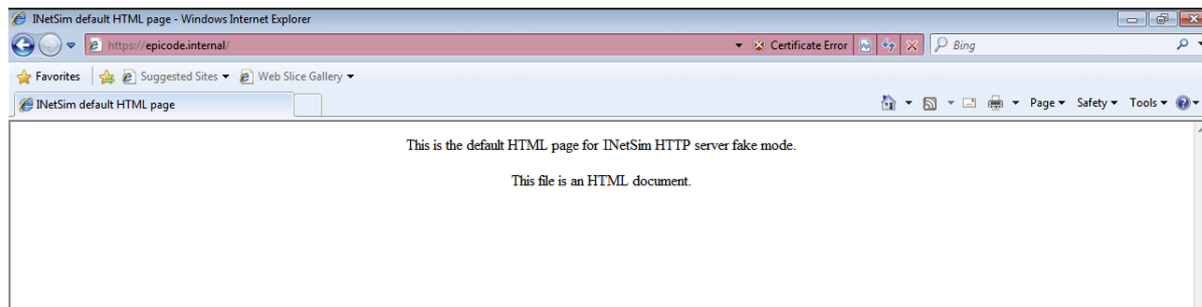
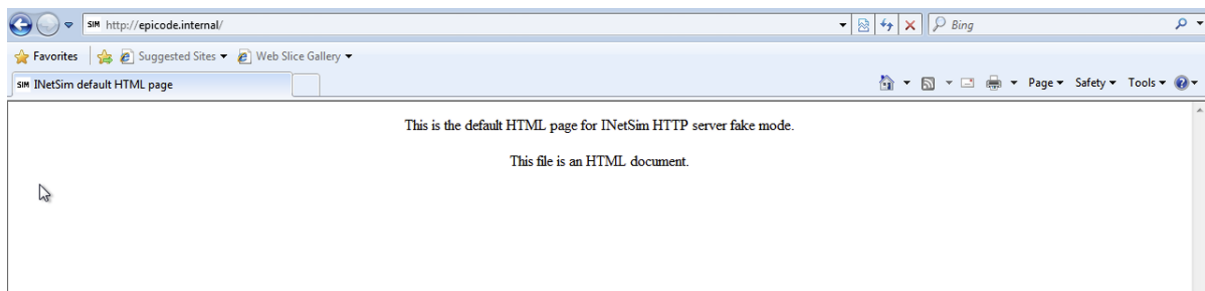
No.	Time	Source	Destination	Protocol	Length	Info
3	0.000443657	192.168.32.101	192.168.32.100	TCP	60	49170 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM
4	0.000472290	192.168.32.100	192.168.32.101	TCP	60	80 → 49170 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
5	0.000491212	192.168.32.101	192.168.32.100	TCP	60	49170 → 80 [ACK] Seq=1 Ack=1 Win=65700 Len=0
6	0.001483748	192.168.32.101	192.168.32.100	HTTP	340	GET / HTTP/1.1
7	0.001490509	192.168.32.100	192.168.32.101	TCP	54	80 → 49170 [ACK] Seq=1 Ack=287 Win=64128 Len=0
8	0.025076973	192.168.32.100	192.168.32.101	TCP	204	80 → 49170 [PSH, ACK] Seq=1 Ack=287 Win=64128 Len=150 [TCP segment of a reassembled PDU]
9	0.031195029	192.168.32.100	192.168.32.101	HTTP	112	200 OK (text/html)
10	0.032812551	192.168.32.101	192.168.32.100	TCP	60	49170 → 80 [ACK] Seq=287 Ack=410 Win=65292 Len=0
11	0.033070508	192.168.32.101	192.168.32.100	TCP	60	49170 → 80 [FIN, ACK] Seq=287 Ack=410 Win=65292 Len=0
12	0.033199573	192.168.32.100	192.168.32.101	TCP	54	80 → 49170 [ACK] Seq=410 Ack=288 Win=64128 Len=0

Intercettazione comunicazione con WireShark (https)



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.32.101	192.168.32.100	TCP	66	49168 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM
2	0.000030223	192.168.32.100	192.168.32.101	TCP	66	443 → 49168 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
3	0.000040122	192.168.32.101	192.168.32.100	TCP	60	49168 → 443 [ACK] Seq=1 Ack=1 Win=65700 Len=0
4	0.001944910	192.168.32.101	192.168.32.100	TLSv1	215	Client Hello
5	0.001964340	192.168.32.100	192.168.32.101	TCP	54	443 → 49168 [ACK] Seq=1 Ack=162 Win=64128 Len=0
6	0.128820369	192.168.32.100	192.168.32.101	TLSv1	1373	Server Hello, Certificate, Server Key Exchange, Server Hello Done
7	0.134921146	192.168.32.101	192.168.32.100	TLSv1	188	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
8	0.134987850	192.168.32.100	192.168.32.101	TCP	54	443 → 49168 [ACK] Seq=1320 Ack=296 Win=64128 Len=0
9	0.135775202	192.168.32.100	192.168.32.101	TLSv1	113	Change Cipher Spec, Encrypted Handshake Message
11	0.343394899	192.168.32.101	192.168.32.100	TCP	60	49168 → 443 [ACK] Seq=296 Ack=1379 Win=64320 Len=0
15	3.279835413	192.168.32.101	224.0.0.252	LLMNR	64	Standard query 0xc0e4 A wpad
17	3.381412103	192.168.32.101	224.0.0.252	LLMNR	64	Standard query 0xc0e4 A wpad
18	3.588764801	192.168.32.101	192.168.32.255	NBNS	92	Name query NB WPAD<00>
19	4.331086347	192.168.32.101	192.168.32.255	NBNS	92	Name query NB WPAD<00>
20	5.078592433	192.168.32.101	192.168.32.255	NBNS	92	Name query NB WPAD<00>
26	8.968295617	192.168.32.101	224.0.0.252	LLMNR	64	Standard query 0x2089 A wpad
28	9.066483871	192.168.32.101	224.0.0.252	LLMNR	64	Standard query 0x2089 A wpad
30	9.270149097	192.168.32.101	192.168.32.255	NBNS	92	Name query NB WPAD<00>
31	10.016510988	192.168.32.101	192.168.32.255	NBNS	92	Name query NB WPAD<00>
32	10.764028635	192.168.32.101	192.168.32.255	NBNS	92	Name query NB WPAD<00>
34	12.534539396	192.168.32.101	192.168.32.100	TLSv1	379	Application Data
35	12.552819505	192.168.32.100	192.168.32.101	TLSv1	235	Application Data
36	12.556844777	192.168.32.100	192.168.32.101	TLSv1	384	Application Data, Encrypted Alert
37	12.557891961	192.168.32.101	192.168.32.100	TCP	60	49168 → 443 [ACK] Seq=621 Ack=1891 Win=65700 Len=0
38	12.558103368	192.168.32.101	192.168.32.100	TCP	60	49168 → 443 [FIN, ACK] Seq=621 Ack=1891 Win=65700 Len=0
39	12.558120620	192.168.32.100	192.168.32.101	TCP	54	443 → 49168 [ACK] Seq=1891 Ack=622 Win=64128 Len=0

Richiesta servizi da Windows



Con la digitazione nella barra di inserimento si risponde all'attivazione dei servizi HTTPS e DNS su Kali linux, quindi la corretta comunicazione delle due macchine.