

Esercitazione fine Modulo 1

Corso Cybersecurity Analyst

TRACCIA

Nell'esercizio di oggi metteremo insieme le competenze acquisite finora.
Lo studente verrà valutato sulla base della risoluzione al problema seguente.

Requisiti e servizi:

- Kali Linux □ IP 192.168.32.100
- Windows 7 □ IP 192.168.32.101
- HTTPS server: attivo
- Servizio DNS per risoluzione nomi di dominio: attivo

Traccia:

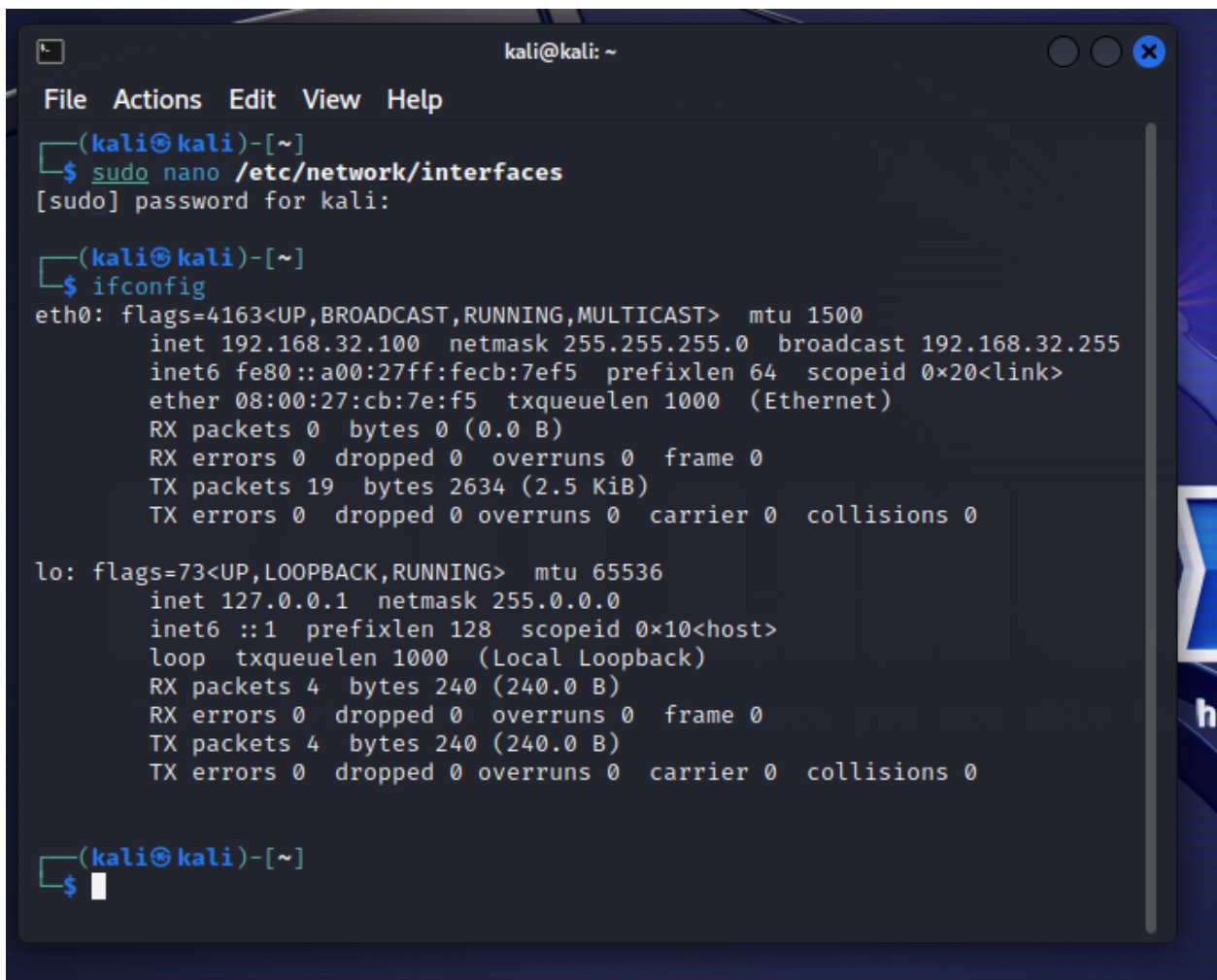
Simulare, in ambiente di laboratorio virtuale, un'architettura client server in cui un client con indirizzo 192.168.32.101 (Windows 7) richiede tramite web browser una risorsa all'hostname epicode.internal che risponde all'indirizzo 192.168.32.100 (Kali).

Si intercetti poi la comunicazione con Wireshark, evidenziando i MAC address di sorgente e destinazione ed il contenuto della richiesta HTTPS.

Ripetere l'esercizio, sostituendo il server HTTPS, con un server HTTP. Si intercetti nuovamente il traffico, evidenziando le eventuali differenze tra il traffico appena catturato in HTTP ed il traffico precedente in HTTPS. Spiegare, motivandole, le principali differenze se presenti.

- CONFIGURAZIONI

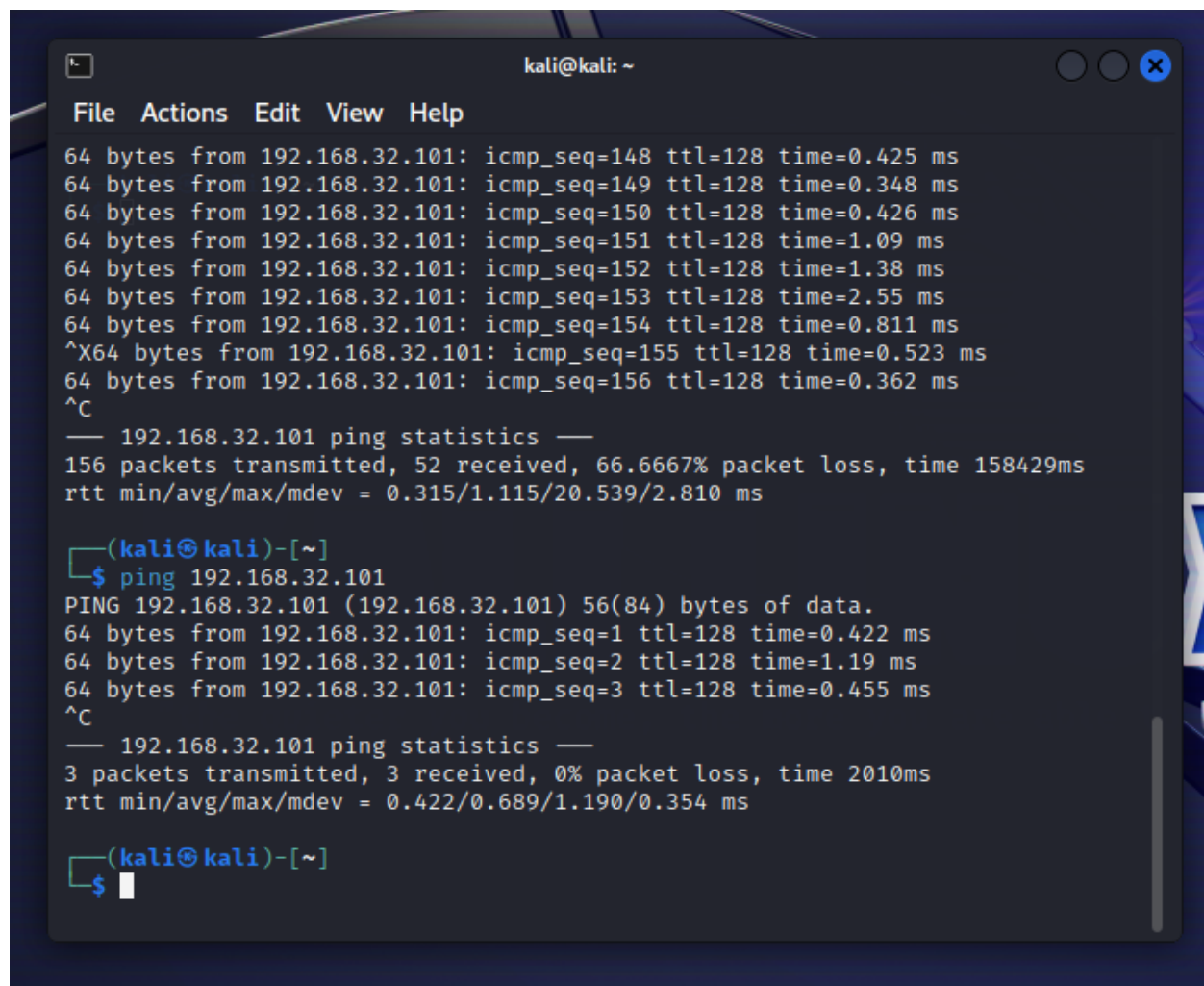
Come primo step devo configurare l'indirizzo IP delle due macchine virtuali utilizzando il comando `sudo nano /etc/network/interfaces`. Questo è il comando che mi permette di modificare l'interfaccia di rete con i privilegi di amministratore. Dopodiché con `ifconfig` vedo se sia stato preso l'indirizzo IP inserito (192.168.32.100 in questo caso è quello di Kali) inoltre troviamo anche l'indirizzo di broadcast, nonché indirizzo utilizzato per inviare dati a tutti i 255 dispositivi collegati in rete e troviamo anche la subnet mask che dimostra la grandezza della rete (rete composta da 255 device)



```
kali@kali: ~  
File Actions Edit View Help  
(kali@kali)-[~]  
$ sudo nano /etc/network/interfaces  
[sudo] password for kali:  
  
(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:feeb:7ef5 prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:cb:7e:f5 txqueuelen 1000 (Ethernet)  
    RX packets 0 bytes 0 (0.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 19 bytes 2634 (2.5 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 0x10<host>  
    loop txqueuelen 1000 (Local Loopback)  
    RX packets 4 bytes 240 (240.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 4 bytes 240 (240.0 B)  
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0  
  
(kali@kali)-[~]  
$
```

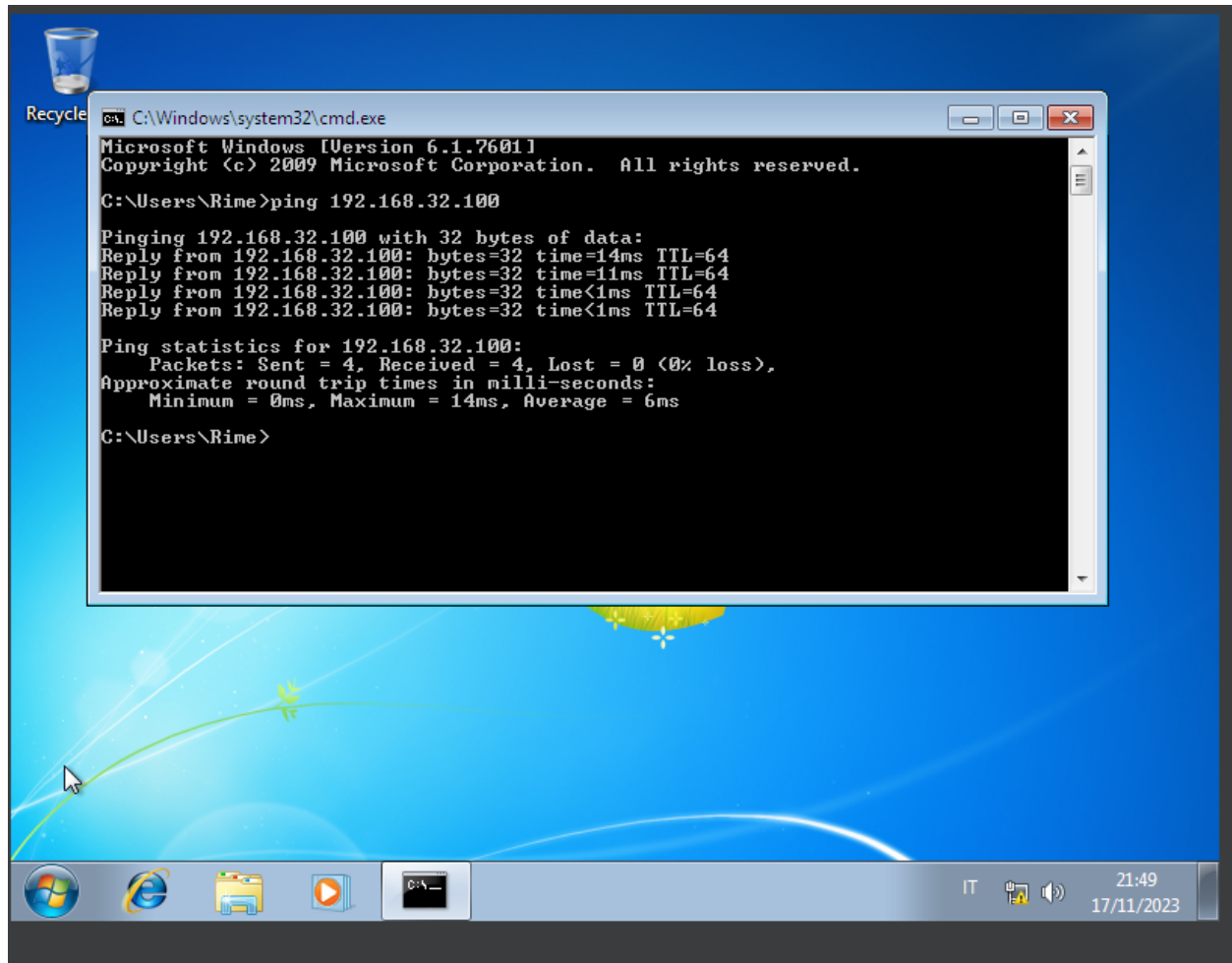
Stesso procedimento di configurazione dell'indirizzo IP va fatto anche su Windows 7 (nella sezione di advanced settings).

Fatto questo, faccio il ping delle due macchine in modo da far sì che possano comunicare. Il ping è un comando che permette ad un dispositivo di collegarsi/comunicare con un altro dispositivo o server. Ecco qui il ping di Kali verso Windows 7 (per permettere questo nelle impostazioni di Firewall di W7 ho impostato che si possono ricevere pacchetti da Kali cioè dall'indirizzo 192.168.32.100, altrimenti di default non viene permesso il collegamento).



```
kali@kali: ~  
File Actions Edit View Help  
64 bytes from 192.168.32.101: icmp_seq=148 ttl=128 time=0.425 ms  
64 bytes from 192.168.32.101: icmp_seq=149 ttl=128 time=0.348 ms  
64 bytes from 192.168.32.101: icmp_seq=150 ttl=128 time=0.426 ms  
64 bytes from 192.168.32.101: icmp_seq=151 ttl=128 time=1.09 ms  
64 bytes from 192.168.32.101: icmp_seq=152 ttl=128 time=1.38 ms  
64 bytes from 192.168.32.101: icmp_seq=153 ttl=128 time=2.55 ms  
64 bytes from 192.168.32.101: icmp_seq=154 ttl=128 time=0.811 ms  
^X64 bytes from 192.168.32.101: icmp_seq=155 ttl=128 time=0.523 ms  
64 bytes from 192.168.32.101: icmp_seq=156 ttl=128 time=0.362 ms  
^C  
— 192.168.32.101 ping statistics —  
156 packets transmitted, 52 received, 66.6667% packet loss, time 158429ms  
rtt min/avg/max/mdev = 0.315/1.115/20.539/2.810 ms  
  
(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=0.422 ms  
64 bytes from 192.168.32.101: icmp_seq=2 ttl=128 time=1.19 ms  
64 bytes from 192.168.32.101: icmp_seq=3 ttl=128 time=0.455 ms  
^C  
— 192.168.32.101 ping statistics —  
3 packets transmitted, 3 received, 0% packet loss, time 2010ms  
rtt min/avg/max/mdev = 0.422/0.689/1.190/0.354 ms  
  
(kali@kali)-[~]  
$
```

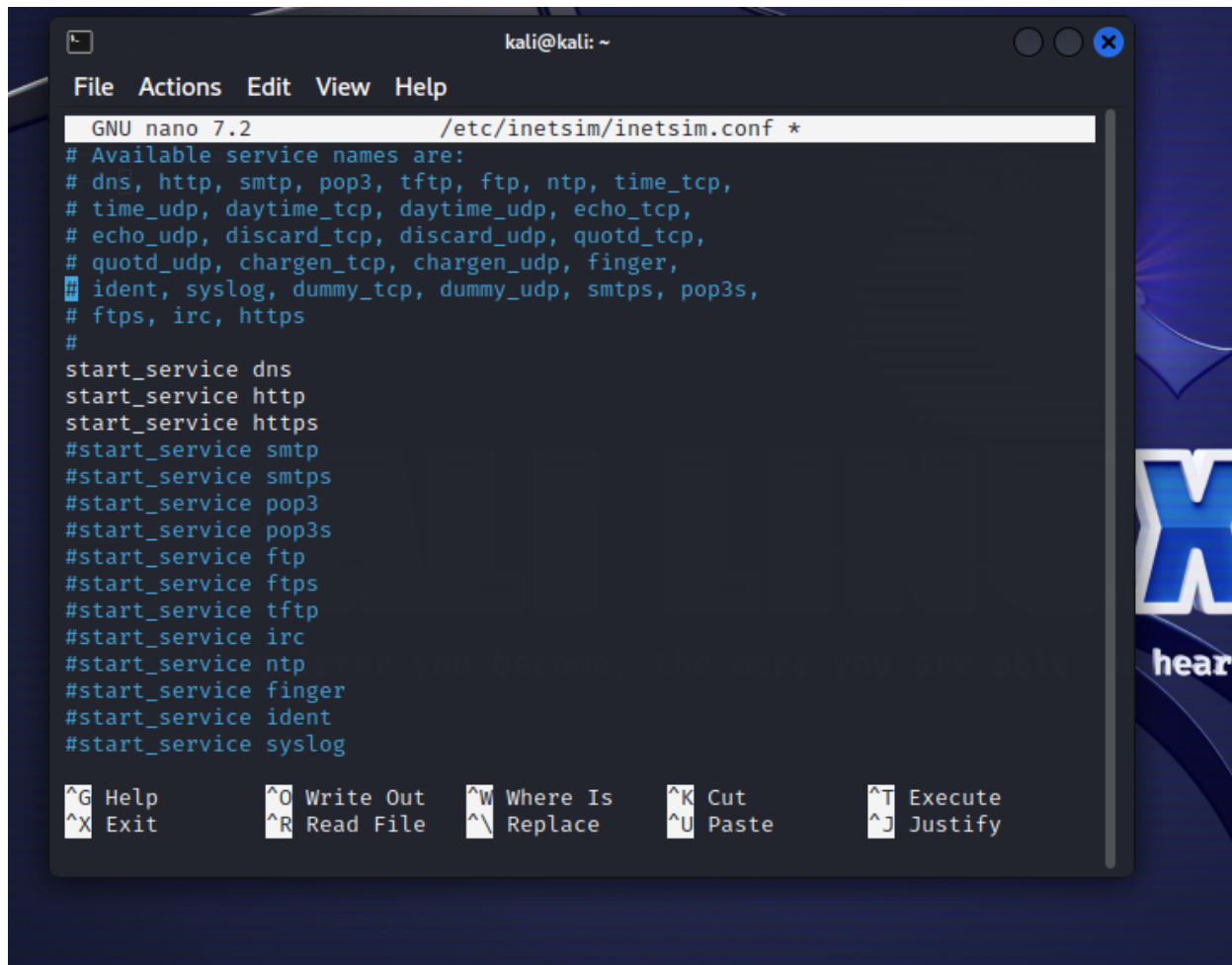
Qui sono su W7 e facendo il ping vedo che i pacchetti vengono mandati a Kali e quindi entrambe le macchine riescono a comunicare.



Siccome è Windows che deve accedere alla risorsa `epicode.internal` presente sul web server di Kali, devo ricorrere al comando `Sudo nano /etc/inetsim/inetsim config`.

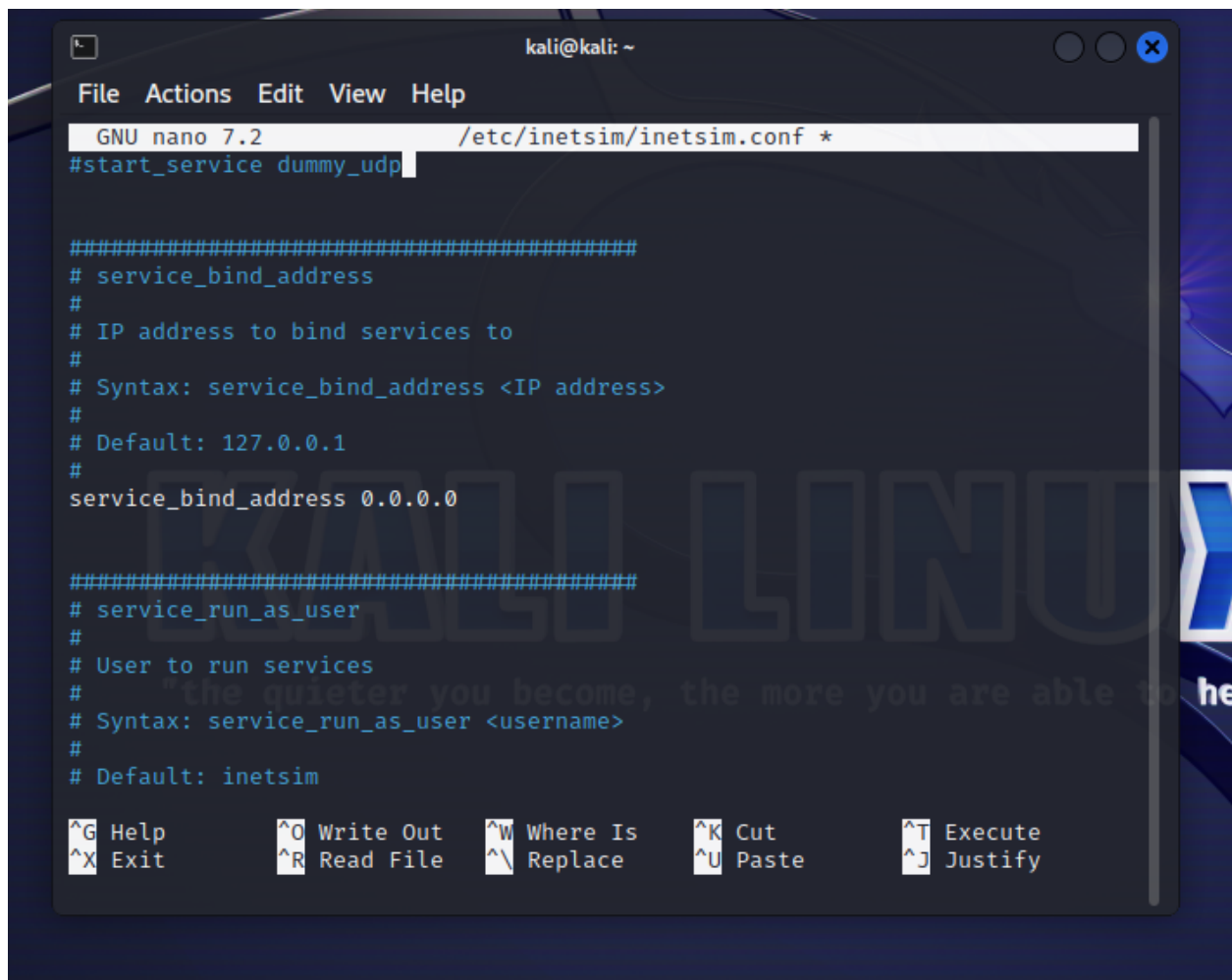
Con questo comando vado a simulare dei servizi internet. Siccome l'esercizio mi chiede di farsì che W7 acceda a `epicode.internal` con il server `http` e `https` allora li commento nella figura qui di seguito. Comento anche il `dns` perchè quest'ultimo

traduce i nomi in indirizzi ip (cosa che ci tornerà utile quando andremo su internet explorer da W7).



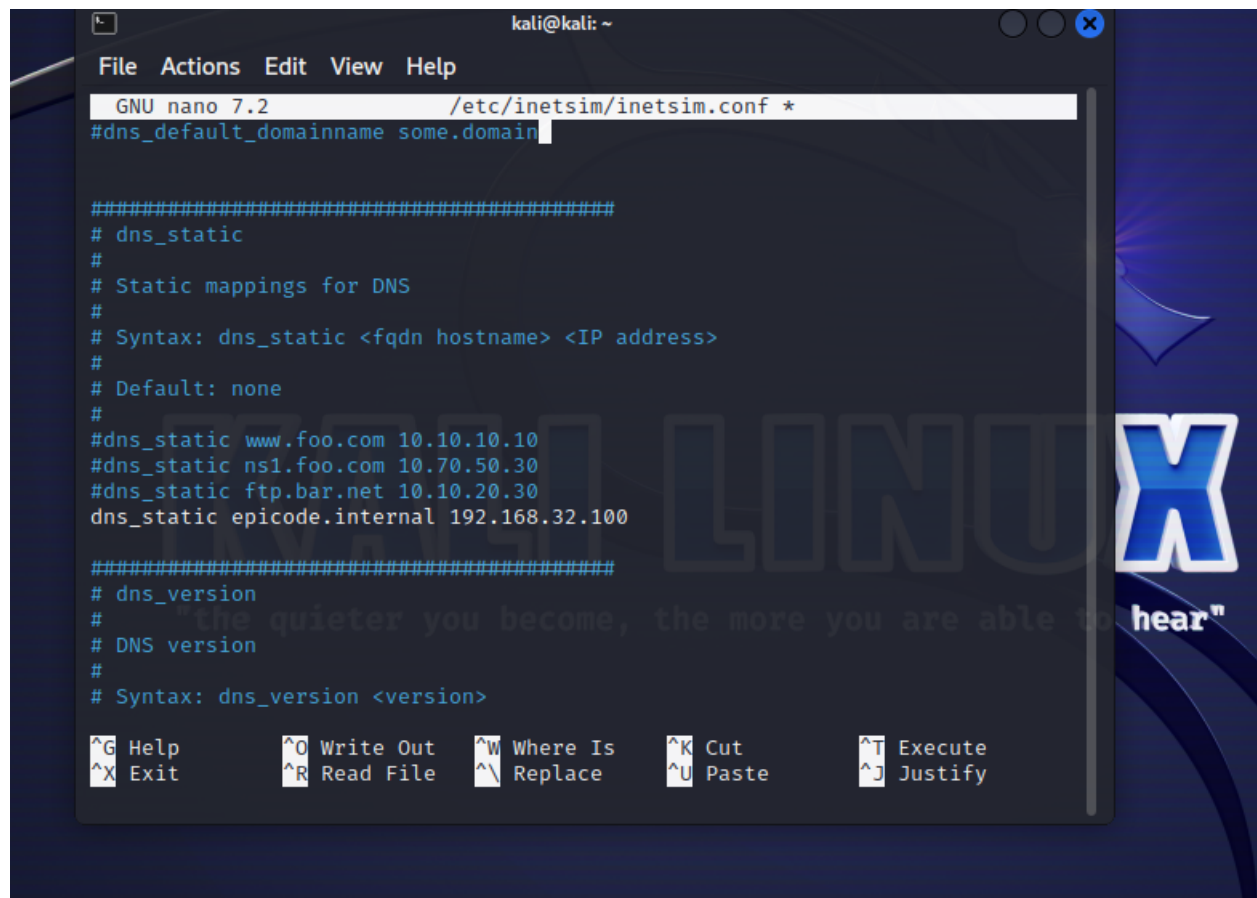
```
kali@kali: ~
File Actions Edit View Help
GNU nano 7.2 /etc/inetsim/inetsim.conf *
# Available service names are:
# dns, http, smtp, pop3, tftp, ftp, ntp, time_tcp,
# time_udp, daytime_tcp, daytime_udp, echo_tcp,
# echo_udp, discard_tcp, discard_udp, quotd_tcp,
# quotd_udp, chargen_tcp, chargen_udp, 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
#start_service syslog

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



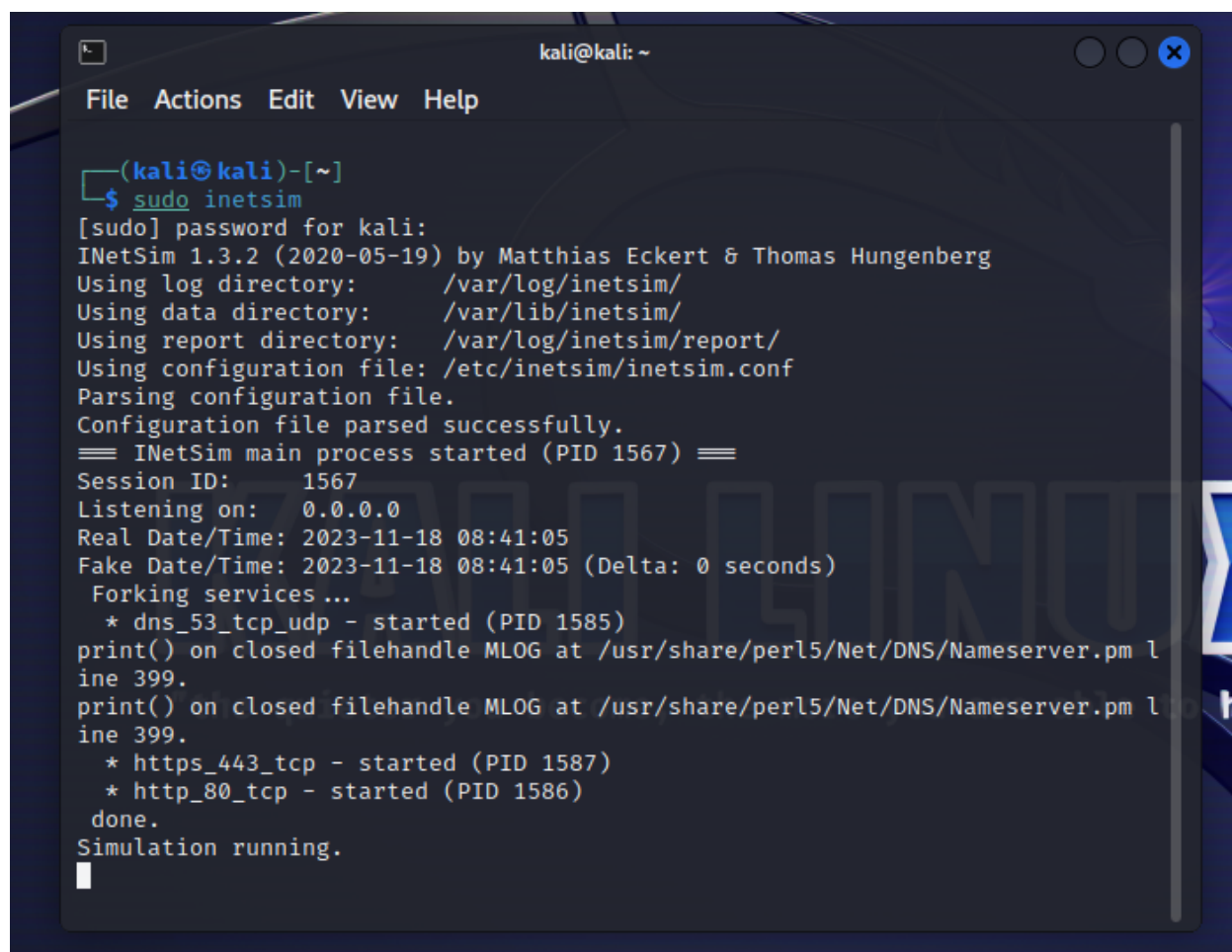
```
kali@kali: ~  
File Actions Edit View Help  
GNU nano 7.2 /etc/inetsim/inetsim.conf *  
#start_service dummy_udp  
  
#####  
# service_bind_address  
#  
# IP address to bind services to  
#  
# Syntax: service_bind_address <IP address>  
#  
# Default: 127.0.0.1  
#  
service_bind_address 0.0.0.0  
  
#####  
# service_run_as_user  
#  
# User to run services  
#  
# Syntax: service_run_as_user <username>  
#  
# Default: inetsim  
  
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute  
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

Il service bind address seguito da un IP si riferisce al fatto che il server web accetterà solo connessioni da parte di client che si vogliono collegare ad un certo IP. In questo caso impostando il service bind address su 0.0.0.0 si accettano connessioni verso tutti gli IP.



```
kali@kali: ~  
File Actions Edit View Help  
GNU nano 7.2 /etc/inetsim/inetsim.conf *  
#dns_default_domainname some.domain  
  
#####  
# 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 ftp.bar.net 10.10.20.30  
dns_static epicode.internal 192.168.32.100  
  
#####  
# dns_version  
#  
# "the quieter you become, the more you are able to hear"  
# DNS version  
#  
# Syntax: dns_version <version>  
  
^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute  
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify
```

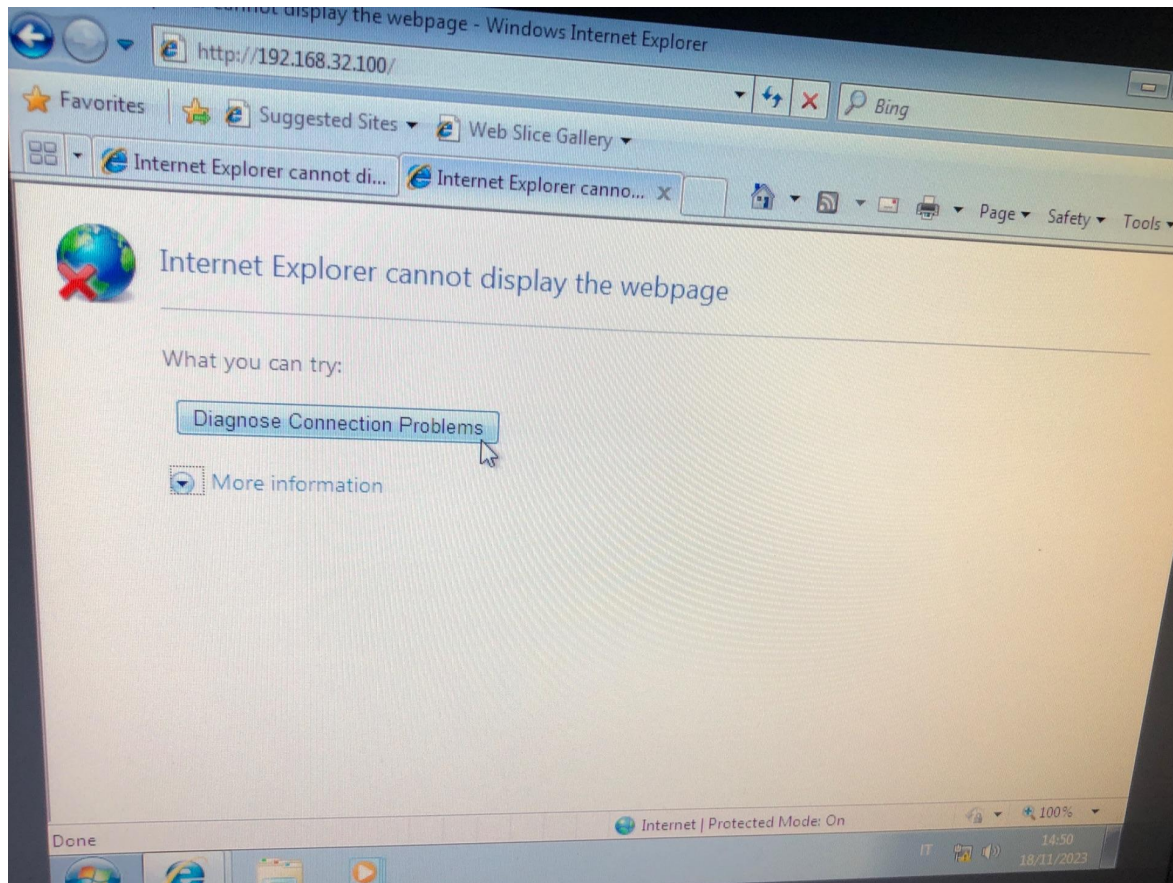
Dato che il DNS traduce i nomi in ip, in questo caso sto associando il nome epicode.internal all'ip 192.168.32.100, nonchè IP di Kali.

A terminal window titled 'kali@kali: ~' with a menu bar (File, Actions, Edit, View, Help). The terminal shows the command 'sudo inetsim' being executed. The output includes the password prompt, version information (INetSim 1.3.2), directory paths, configuration file parsing, and the start of the main process (PID 1567). It lists services like dns_53_tcp_udp, https_443_tcp, and http_80_tcp starting at various PIDs. The terminal ends with 'Simulation running.' and a cursor.

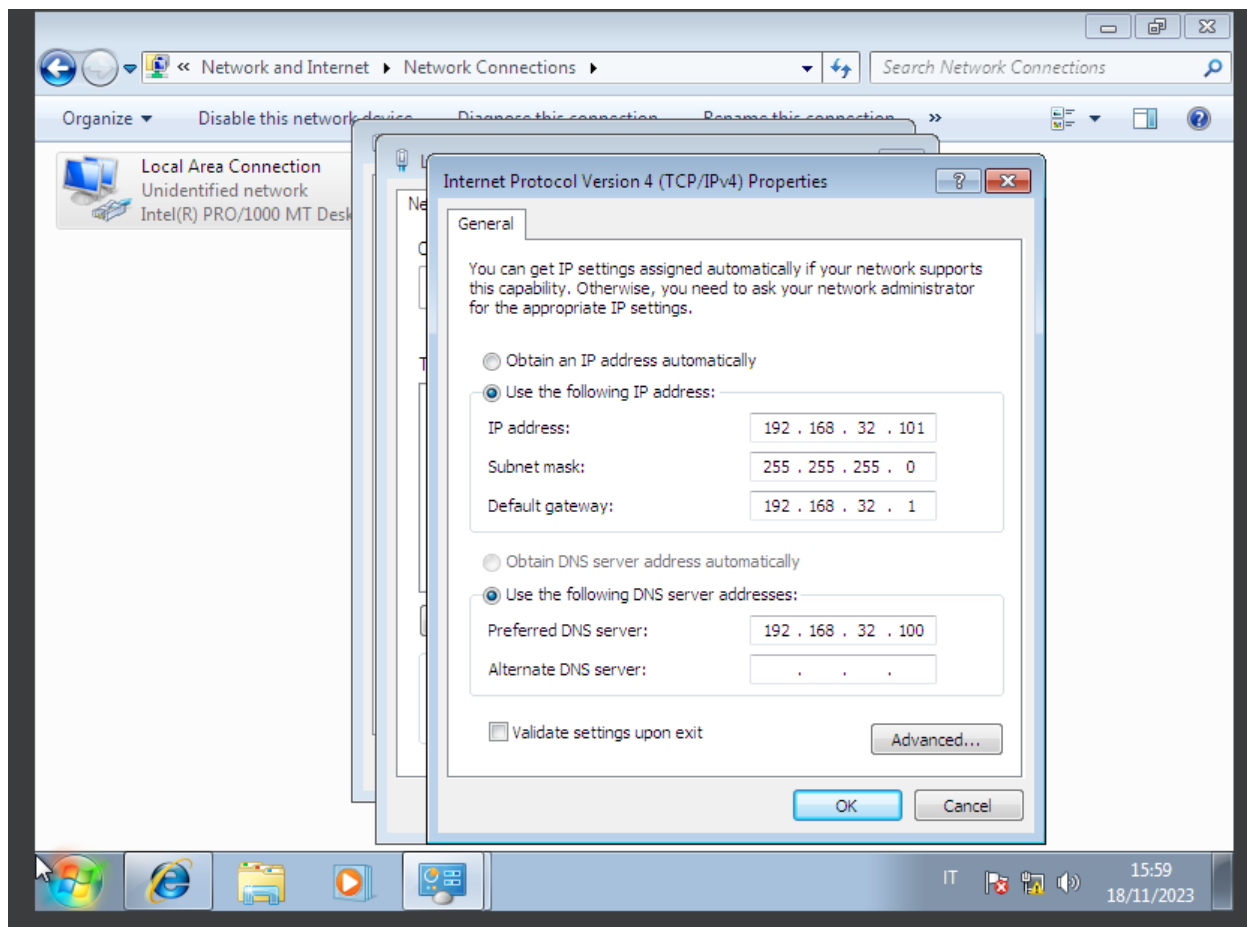
```
(kali@kali)-[~]
$ sudo inetsim
[sudo] password for kali:
INetSim 1.3.2 (2020-05-19) by Matthias Eckert & Thomas Hungenberg
Using log directory:      /var/log/inetsim/
Using data directory:     /var/lib/inetsim/
Using report directory:   /var/log/inetsim/report/
Using configuration file: /etc/inetsim/inetsim.conf
Parsing configuration file.
Configuration file parsed successfully.
≡ INetSim main process started (PID 1567) ≡
Session ID:      1567
Listening on:    0.0.0.0
Real Date/Time:  2023-11-18 08:41:05
Fake Date/Time: 2023-11-18 08:41:05 (Delta: 0 seconds)
Forking services ...
  * dns_53_tcp_udp - started (PID 1585)
print() on closed filehandle MLOG at /usr/share/perl5/Net/DNS/Nameserver.pm l
ine 399.
print() on closed filehandle MLOG at /usr/share/perl5/Net/DNS/Nameserver.pm l
ine 399.
  * https_443_tcp - started (PID 1587)
  * http_80_tcp - started (PID 1586)
done.
Simulation running.
█
```

Una volta configurato inetsim, col seguente comando verifico quanto fatto poc'anzi sia stato preso correttamente e poi avvio la simulazione.

Finita la parte di configurazione, vado su W7 e cerco di accedere sul browser ad epicode.internal, però mi esce questa schermata.

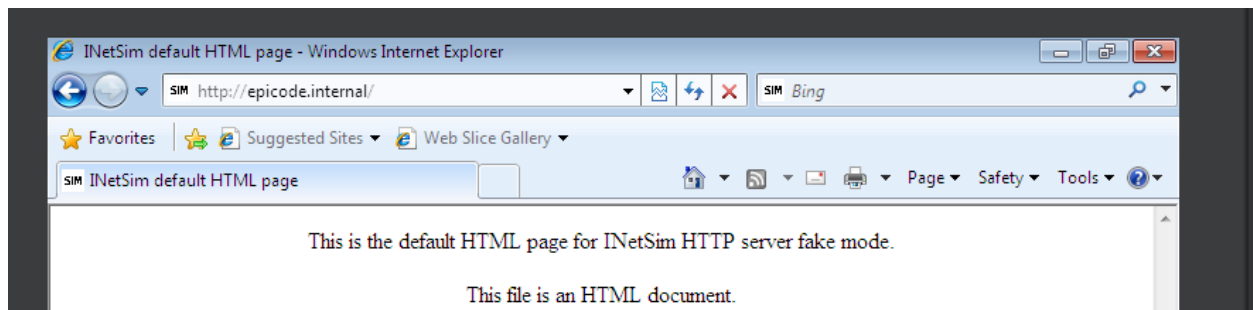
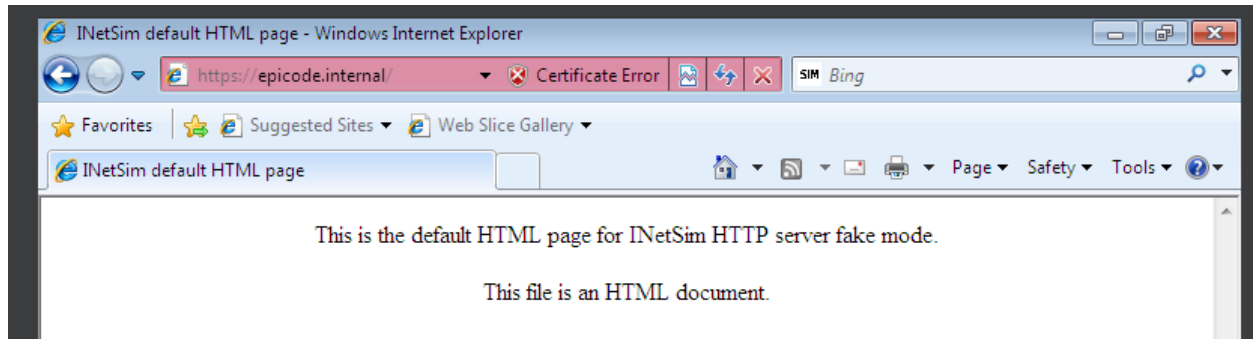


Nella sezione advanced settings inserisco 192.168.32.100 come dns server preferito (vedi seguente figura)



Non capisco il motivo, anche perché nella configurazione di inetsim dedicata al dns ho specificato l'ip seguente 192.168.32.100.

Fatto questo, vado sul web browser di W7 per mandare la richiesta http e https, così poi da tracciare il traffico se wireshark.



No.	Time	Source	Destination	Protocol	Length	Info
183	1130.7391603...	192.168.32.101	192.168.32.255	NBNS	92	Name query NB WPAD<00>
184	1131.4890052...	192.168.32.101	192.168.32.255	NBNS	92	Name query NB WPAD<00>
185	1132.2492757...	192.168.32.101	192.168.32.100	DNS	80	Standard query 0x7711 A watson.microsoft.com
186	1132.3306574...	192.168.32.100	192.168.32.101	DNS	96	Standard query response 0x7711 A watson.microsoft.com A 192.168.32.100
187	1132.3343205...	192.168.32.101	192.168.32.100	TCP	66	49219 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM
188	1132.3343528...	192.168.32.100	192.168.32.101	TCP	66	80 → 49219 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=64
189	1132.3350517...	192.168.32.101	192.168.32.100	TCP	60	49219 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
190	1132.3350520...	192.168.32.101	192.168.32.100	HTTP	405	GET /StageOne/Generic/WindowsUpdateFailure/7_5_7601_17514/80072f76/D67661EB-2...
191	1132.3350944...	192.168.32.100	192.168.32.101	TCP	54	80 → 49219 [ACK] Seq=1 Ack=352 Win=64128 Len=0
192	1132.3914996...	192.168.32.100	192.168.32.101	TCP	204	80 → 49219 [PSH, ACK] Seq=1 Ack=352 Win=64128 Len=150 [TCP segment of a reass...
193	1132.3994943...	192.168.32.100	192.168.32.101	HTTP	312	HTTP/1.1 200 OK (text/html)
194	1132.4000481...	192.168.32.101	192.168.32.100	TCP	60	49219 → 80 [ACK] Seq=352 Ack=410 Win=65280 Len=0
195	1132.4000485...	192.168.32.101	192.168.32.100	TCP	60	49219 → 80 [FIN, ACK] Seq=352 Ack=410 Win=65280 Len=0
196	1132.4001006...	192.168.32.100	192.168.32.101	TCP	54	80 → 49219 [ACK] Seq=410 Ack=353 Win=64128 Len=0

Frame 167: 225 bytes on wire (1800 bits), 225 bytes captured (1800 bits) on	0000	08 00 27 cb 7e f5 08 00	27 0b 4d 00 08 00 45 00	M...E..
Ethernet II, Src: PcsCompu_9b:4d:00 (08:00:27:9b:4d:00), Dst: PcsCompu_cb:7	0010	00 d3 04 c0 40 00 00 06	33 4b c0 a8 20 65 c0 a8	@...3K...e..
Internet Protocol Version 4, Src: 192.168.32.101, Dst: 192.168.32.100	0020	20 64 c0 42 00 50 a7 6b	a9 c3 5f 37 fd 57 50 18	d B P k ..._7 WP..
Transmission Control Protocol, Src Port: 49218, Dst Port: 80, Seq: 1, Ack:	0030	01 00 17 75 00 00 48 45	41 44 20 2f 76 39 2f 77u...HE AD /v9/w
Hypertext Transfer Protocol	0040	69 6e 64 6f 77 73 75 70	64 61 74 65 2f 72 65 64	indowsup date/red
	0050	69 72 2f 6d 75 76 34 77	75 72 65 64 69 72 2e 63	ir/muv4w uredir.c
	0060	61 62 3f 32 33 31 31 31	38 31 38 35 32 20 48 54	ab?23111 81852 HT
	0070	54 50 2f 31 2e 31 0d 0a	43 6f 6e 6e 65 63 74 69	TP/1.1... Connecti
	0080	6f 6e 3a 20 4b 65 65 70	2d 41 6c 69 76 65 0d 0a	on: Keep -Alive..
	0090	41 63 63 65 70 74 3a 20	2a 2f 2a 0d 0a 55 73 65	Accept: /*...Use
	00a0	72 2d 41 67 65 6e 74 3a	20 57 69 6e 64 6f 77 73	r-Agent: Windows
	00b0	2d 55 70 64 61 74 65 2d	41 67 65 6e 74 0d 0a 48	-Update- Agent H
	00c0	6f 73 74 3a 20 77 77 77	2e 75 70 64 61 74 65 2e	ost: www .update.
	00d0	6d 69 63 72 6f 73 6f 66	74 2e 63 6f 6d 0d 0a 0d	microsof t.com...
	00e0	0a		

Nella figura qui sopra stiamo analizzando il traffico secondo il protocollo http, in cui passano informazioni in chiaro. Possiamo vedere i tipi di richiesta effettuati nel browser, le porte...

Vediamo anche i flag SYN e ACK, flag di controllo che indicano l'inizio e la conferma delle connessioni tra due dispositivi.

- MAC ADDRESS DI KALI E W7

Ho preso un pacchetto da analizzare per vedere se gli indirizzi mac indicati coincidessero con quelli di Kali e W7.

Mac address di Kali:08:00:27:cb:7e:f5

Mac address di W7:08:00:27:9b:4d:00

Entrambi coincidono alla voce Source e Destination sotto la voce ethernet 2.

169	631.348190368	192.168.32.100	192.168.32.101	TCP
170	631.871309952	192.168.32.101	192.168.32.100	TCP
171	631.871368833	192.168.32.100	192.168.32.101	TCP
172	632.387829693	192.168.32.101	192.168.32.100	TCP
173	632.387885400	192.168.32.100	192.168.32.101	TCP
174	636.390534897	PcsCompu_cb:7e:f5	PcsCompu_9b:4d:00	ARP
175	636.390872514	PcsCompu_9b:4d:00	PcsCompu_cb:7e:f5	ARP

Frame 169: 54 bytes on wire (432 bits), 54 bytes captured (432 bits)	0000
Ethernet II, Src: PcsCompu_cb:7e:f5 (08:00:27:cb:7e:f5), Dst: PcsCompu_9b:4d:00 (08:00:27:9b:4d:00)	0010
Destination: PcsCompu_9b:4d:00 (08:00:27:9b:4d:00)	0020
Source: PcsCompu_cb:7e:f5 (08:00:27:cb:7e:f5)	0030
Type: IPv4 (0x0800)	
Internet Protocol Version 4, Src: 192.168.32.100, Dst: 192.168.32.101	
Transmission Control Protocol, Src Port: 80, Dst Port: 49159, Seq: 169190368	

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

C:\Users\Rime>ipconfig /all

Windows IP Configuration

    Host Name . . . . . : Rime-PC
    Primary Dns Suffix . . . . . :
    Node Type . . . . . : Hybrid
    IP Routing Enabled. . . . . : No
    WINS Proxy Enabled. . . . . : No

Ethernet adapter Local Area Connection:

    Connection-specific DNS Suffix . :
    Description . . . . . : Intel(R) PRO/1000 MT Desktop Adapter
    Physical Address. . . . . : 08-00-27-9B-4D-00
    DHCP Enabled. . . . . : No
    Autoconfiguration Enabled . . . . : Yes
    Link-local IPv6 Address . . . . . : fe80::a525:a52:688f:64%11(Preferred)
```

```
kali@kali: ~
File Actions Edit View Help
(kali@kali)-[~] 7:cb:7e:f5
$ 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:feeb:7ef5 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:cb:7e:f5 txqueuelen 1000 (Ethernet)
    RX packets 1177 bytes 104955 (102.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 340 bytes 59335 (57.9 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

