

Week 1 - Esercizio 4

Configurazione Firewall Windows 7

Consegna

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:

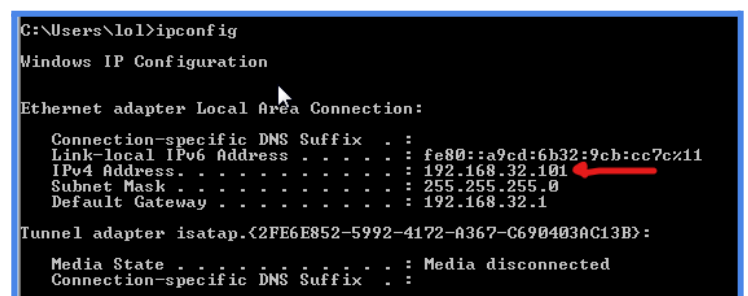
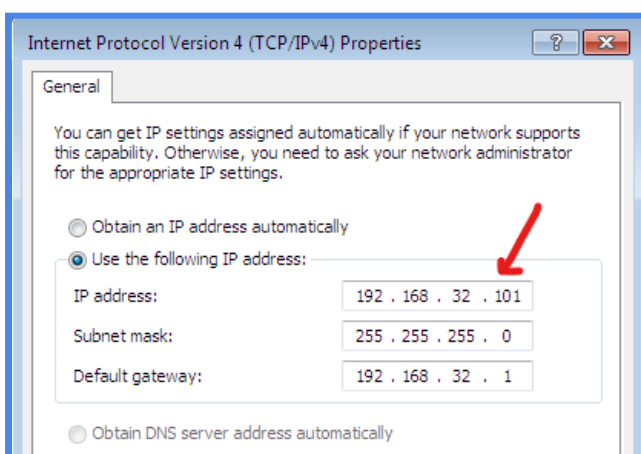
Simulare, in ambiente di laboratorio virtuale, un'architettura client server in cui un client con indirizzo 192.168.32.101 richiede tramite web browser una risorsa all'hostname epicode.internal che risponde all'indirizzo 192.168.32.100.

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 differenze tra il traffico appena catturato in HTTP ed il traffico precedente in HTTPS. Spiegare, motivandole, le principali differenze.

Procedimento

1. Vado innanzitutto a cambiare l'indirizzo della macchina Windows 7



2. Cambio anche quella di Kali

```
GNU nano 6.3 /etc/network/interfaces
# 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 192.168.32.100/24
gateway 192.168.32.1
```

3. Modifico poi le policy del firewall per fare in modo che le due macchine possano comunicare (Protocollo TCP sia in entrata che in uscita)

To which ports and protocols does this rule apply?

Protocol type: **TCP**

Protocol number: **6**

Local port: **All Ports**

Remote port: **All Ports**

Internet Control Message Protocol (ICMP) settings: **Customize...**

Allow Control Packets Properties

General | Programs and Services | Computers

Protocols and Ports | Scope | Advanced | Users

Local IP address

☐ Any IP address

☒ These IP addresses:

192.168.32.100
192.168.32.101

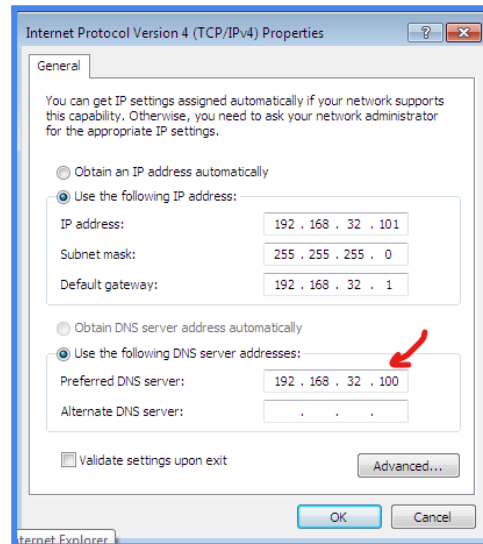
Add...
Edit...
Remove

4. Vado poi a configurare il DNS Service nel file **inetsim.conf** e lo cambio con **dns_static epicode.internal 192.168.32.100** come richiesto nella consegna. Stessa cosa con **service_bind_address**

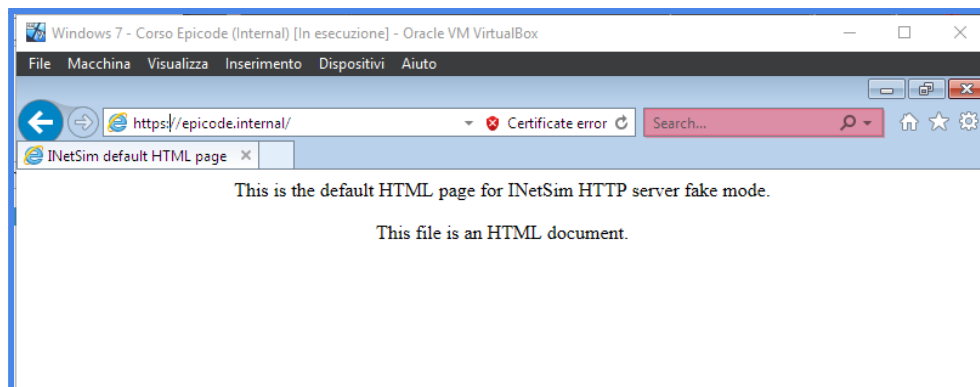
```
#####
# dns_static
#
# Static mappings for DNS
#
# Syntax: dns_static <fqdn hostname> <IP address>
#
# Default: none
#
#dns_static www.foo.com 10.10.10.10
#dns_static ns1.foo.com 10.70.50.30
#dns_static ftp.bar.net 10.10.20.30
dns_static epicode.internal 192.168.32.100
#####
```

```
#####
# service_bind_address
#
# 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
#
```

5. Avvio la simulazione attraverso il comando **inetsim** con la nuova configurazione
6. Successivamente vado a cambiare il server DNS a cui si deve collegare la macchina con Windows 7



7. A questo punto ricerco dalla macchina Windows **epicode.internal** per verificare che la richiesta https può essere effettuata senza problemi



8. Apro Wireshark e vado a controllare i pacchetti scambiati tra le due macchine e Ricarico la pagina per analizzarli

HTTP →

No.	Time	Source	Destination	Protocol	Length	Info
90	0.000000	192.168.32.100	192.168.32.101	ARP	62	Who has 192.168.32.100? Tell 192.168.32.101
91	0.000000	192.168.32.100	192.168.32.101	ARP	44	192.168.32.100 is at 08:00:27:22:46:4f
90	0.000000	192.168.32.100	80	TCP	68	49222 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_
91	0.000000	80	49222	TCP	68	80 → 49222 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
90	0.000000	192.168.32.100	80	TCP	62	49222 → 80 [ACK] Seq=1 Ack=1 Win=65536 Len=0
90	0.000000	192.168.32.100	80	HTTP	308	GET / HTTP/1.1
91	0.000000	80	49222	TCP	56	80 → 49222 [ACK] Seq=1 Ack=253 Win=64128 Len=0
91	0.000000	80	49222	TCP	206	80 → 49222 [PSH, ACK] Seq=1 Ack=253 Win=64128 Len=150 [TCP
90	0.000000	192.168.32.100	80	TCP	62	49222 → 80 [ACK] Seq=253 Ack=151 Win=65536 Len=0
91	0.000000	192.168.32.100	80	HTTP	314	HTTP/1.1 200 OK (text/html)
90	0.000000	192.168.32.100	80	TCP	62	49222 → 80 [ACK] Seq=253 Ack=409 Win=65280 Len=0
91	0.000000	80	49222	TCP	56	80 → 49222 [FIN, ACK] Seq=409 Ack=253 Win=64128 Len=0
90	0.000000	192.168.32.100	80	TCP	62	49222 → 80 [ACK] Seq=253 Ack=410 Win=65280 Len=0
90	0.000000	192.168.32.100	80	TCP	62	49222 → 80 [FIN, ACK] Seq=253 Ack=410 Win=65280 Len=0
91	0.000000	80	49222	TCP	56	80 → 49222 [ACK] Seq=410 Ack=254 Win=64128 Len=0
90	0.000000	192.168.32.100	80	TCP	68	49226 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=256 SACK_
91	0.000000	80	49226	TCP	68	80 → 49226 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460

WINDOWS

```
Link-layer address length: 6
Source: PcsCompu_bb:db:07 (08:00:27:bb:db:07)
Unused: 0000
Protocol: IPv4 (0x0800)
```

KALI

```
Link-layer address length: 6
Source: PcsCompu_22:46:4f (08:00:27:22:46:4f)
Unused: 0000
Protocol: IPv4 (0x0800)
```

HTTPS →

No.	Time	Source	Destination	Protocol	Length	Info
192.168.32.100	0.000000	192.168.32.100	443	TCP	62	49240 → 443 [FIN, ACK] Seq=1 Ack=1 Win=164
192.168.32.101	0.000000	192.168.32.101	443	TLSv1.2	87	Encrypted Alert
192.168.32.100	0.000000	192.168.32.100	443	TCP	62	49240 → 443 [RST, ACK] Seq=2 Ack=32 Win=0
192.168.32.100	0.000000	192.168.32.100	443	TCP	68	49243 → 443 [SYN] Seq=0 Win=8192 Len=0 MSS
192.168.32.101	0.000000	192.168.32.101	49243	TCP	68	443 → 49243 [SYN, ACK] Seq=0 Ack=1 Win=642
192.168.32.100	0.000000	192.168.32.100	443	TCP	62	49243 → 443 [ACK] Seq=1 Ack=1 Win=65700 Le
192.168.32.100	0.000000	192.168.32.100	443	TLSv1.2	273	Client Hello
192.168.32.101	0.000000	192.168.32.101	49243	TCP	56	443 → 49243 [ACK] Seq=1 Ack=218 Win=64128
192.168.32.101	0.000000	192.168.32.101	443	TLSv1.2	1823	Server Hello, Certificate, Server Key Exch
192.168.32.100	0.000000	192.168.32.100	443	TCP	62	49243 → 443 [ACK] Seq=218 Ack=1768 Win=657
192.168.32.100	0.000000	192.168.32.100	443	TLSv1.2	374	Client Key Exchange, Change Cipher Spec, E
192.168.32.101	0.000000	192.168.32.101	443	TCP	56	443 → 49243 [ACK] Seq=1768 Ack=536 Win=641
192.168.32.101	0.000000	192.168.32.101	443	TLSv1.2	107	Change Cipher Spec, Encrypted Handshake Me
192.168.32.100	0.000000	192.168.32.100	443	TCP	62	49243 → 443 [ACK] Seq=536 Ack=1819 Win=656
97	0.000000	192.168.32.100	192.168.32.101	ARP	62	Who has 192.168.32.1? Tell 192.168.32.101
97	0.000000	192.168.32.101	192.168.32.100	ARP	62	Who has 192.168.32.1? Tell 192.168.32.101
97	0.000000	192.168.32.100	192.168.32.101	ARP	62	Who has 192.168.32.1? Tell 192.168.32.101

WINDOWS

```
Link-layer address length: 6
Source: PcsCompu_bb:db:07 (08:00:27:bb:db:07)
Unused: 0000
```

KALI

```
Link-layer address length: 6
Source: PcsCompu_22:46:4f (08:00:27:22:46:4f)
Unused: 0000
```