

# Esame di fine modulo, svolgimento logico

SACHELE

Traccia: Simulare, in ambiente di laboratorio virtuale, un'architettura client server in cui un client con indirizzo 192.168.50.102 Windows richiede tramite web browser una risorsa all'hostname epicode.internal che risponde all'indirizzo 192.168.50.100 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.

## RAGIONAMENTO

I requisiti sono: avere **Kali** e **Windows** già con un IP assegnato, un HTTPS server attivo e servizio DNS attivo.

IP assegnato a **Windows**

```
cmd Prompt dei comandi
Microsoft Windows [Versione 10.0.10240]
(c) 2015 Microsoft Corporation. Tutti i diritti sono riservati.

C:\Users\user>ipconfig

Configurazione IP di Windows

Scheda Ethernet Ethernet:

    Suffisso DNS specifico per connessione:
    Indirizzo IPv4. . . . . : 192.168.50.102
    Subnet mask . . . . . : 255.255.255.0
    Gateway predefinito . . . . . : 192.168.50.1

Scheda Tunnel isatap.{92D61F82-1D19-45C9-B7CF-2E5AF2D63627}:

    Stato supporto. . . . . : Supporto disconnesso
    Suffisso DNS specifico per connessione:

C:\Users\user>
```

IP assegnato a **Kali**

```
kali@kali: ~  
Session Azioni Modifica Visualizza Aiuto  
(kali@kali)-[~]  
$ ifconfig  
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
    inet 192.168.50.100 netmask 255.255.255.0 broadcast 192.168.50.255  
    inet6 fe80::dc13:6a31:8d77:2630 prefixlen 64 scopeid 0x20<link>  
    ether 08:00:27:e3:41:f0 txqueuelen 1000 (Ethernet)  
    RX packets 0 bytes 0 (0.0 B)  
    RX errors 0 dropped 0 overruns 0 frame 0  
    TX packets 23 bytes 2822 (2.7 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 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
```

Apriamo Inetsim per iniziare a capire come configurare un servizio DNS:

```
(kali@kali)-[~]  
$ sudo nano /etc/inetsim/inetsim.conf  
[sudo] password di kali:
```

Scorrendo le opzioni, troviamo start\_service DNS, lo modifichiamo e lo rendiamo attivo:

DNS service non attivo (in figura sotto)

```
#start_service dns  
start_service http  
start_service https
```

DNS service attivo:

```
start_service dns  
start_service http  
start_service https
```

Proseguendo nella nostra cartella troviamo altre impostazioni che sono state modificate per poter configurare correttamente il DNS service:

```

GNU nano 8.6
# dns_bind_port
#
# Port number to bind DNS service to
#
# Syntax: dns_bind_port <port number>
#
# Default: 53
#
dns_bind_port 53

#####
# dns_default_ip
#
# Default IP address to return with DNS replies
#
# Syntax: dns_default_ip <IP address>
#
# Default: 192.168.50.100
#
dns_default_ip 192.168.50.100

#####
# dns_default_hostname
#
# Default hostname to return with DNS replies
#
# Syntax: dns_default_hostname <hostname>
#
# Default: www.epicode.internal.com
#
dns_default_hostname epicode

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

#####

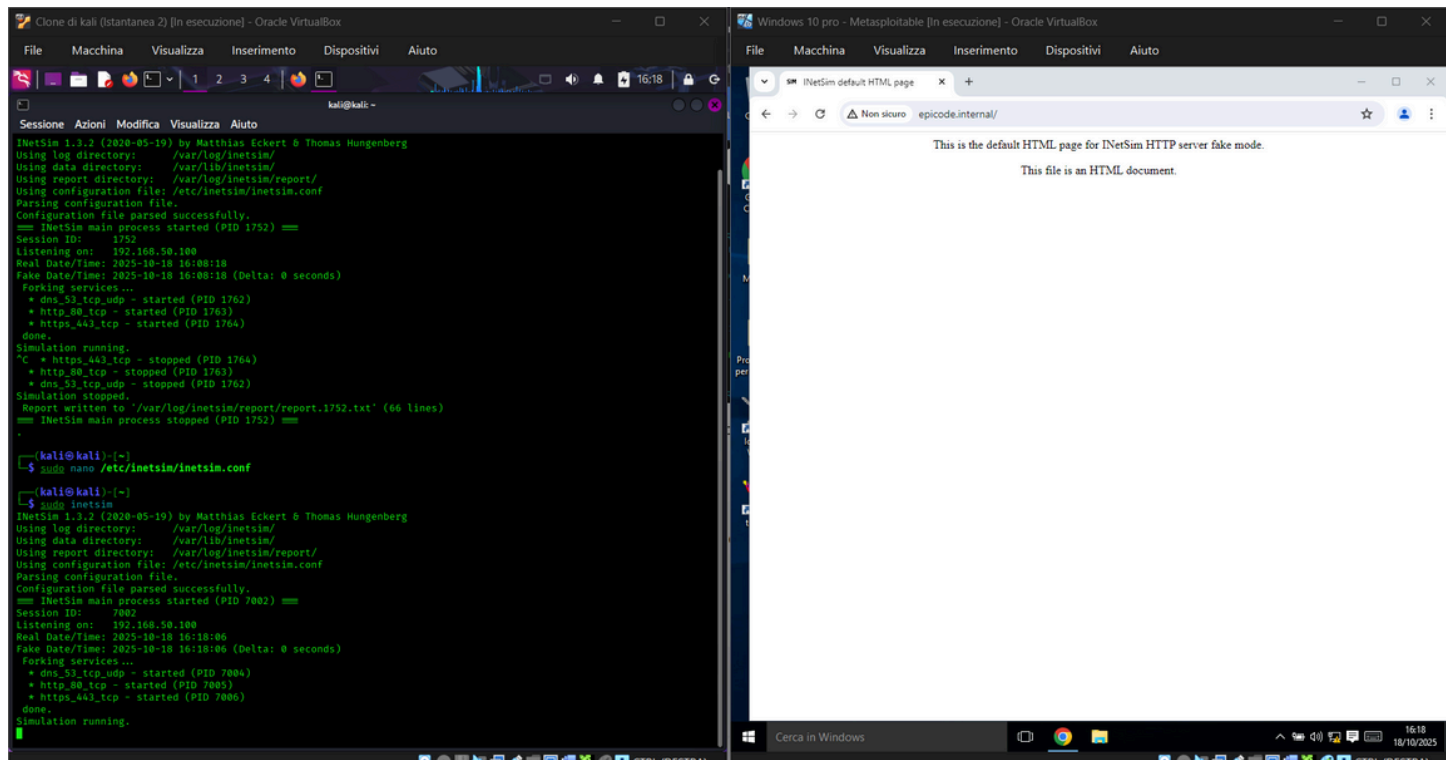
```

Salviamo le modifiche utilizzando il comando CTRL+O seguito da invio e CTRL+X

(nell'attivare InetSim ho riscontrato degli errori di compatibilità con perl, dopo svariate ricerche online, sono riuscito a trovare una soluzione downgradando perl e del dns di inetsim)

## TEST FUNZIONAMENTO

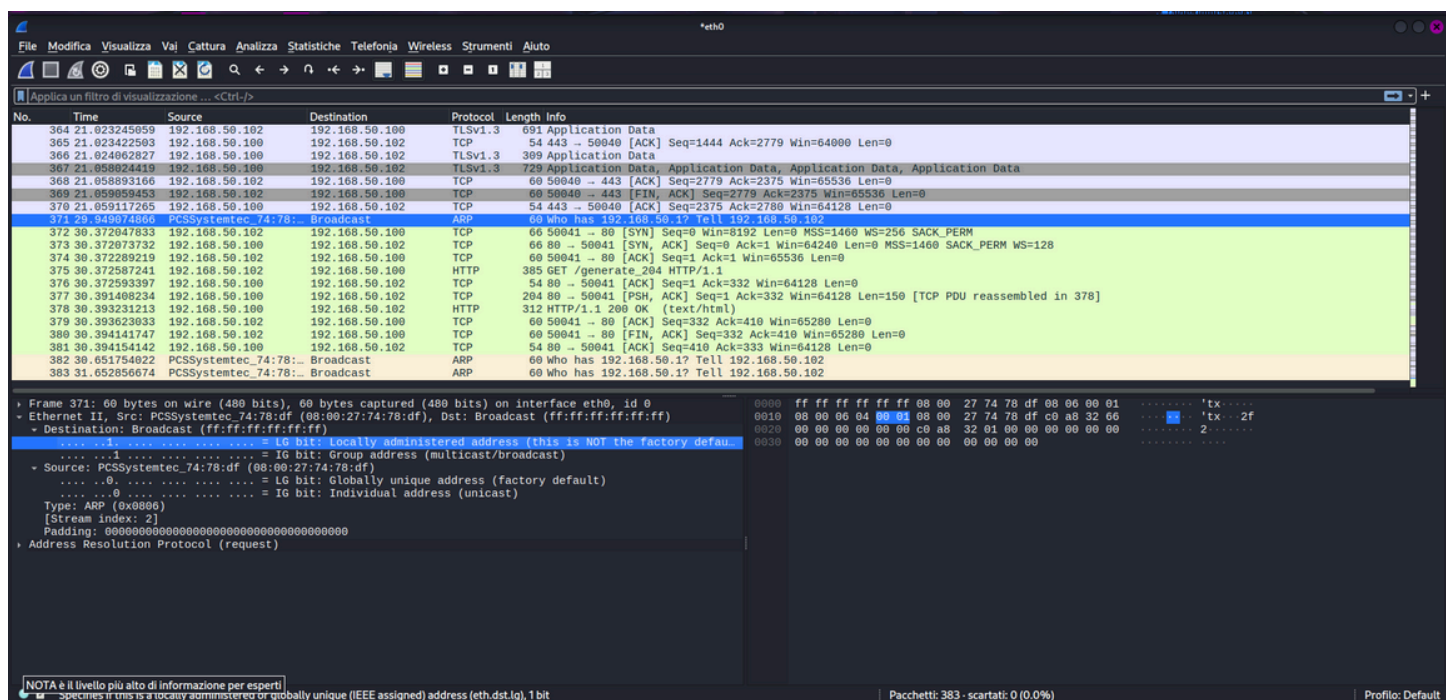
Attiviamo InetSim e verifichiamo che su Windows sia possibile accedere al nostro server.



Inetsim è attivo, di conseguenza Windows riesce a connettersi ad epicode tramite DNS hostname epicode.internal.

Passiamo adesso all'analisi dei pacchetti tramite Wireshark evidenziando le principali differenze tra l'utilizzo del protocollo http ed https.

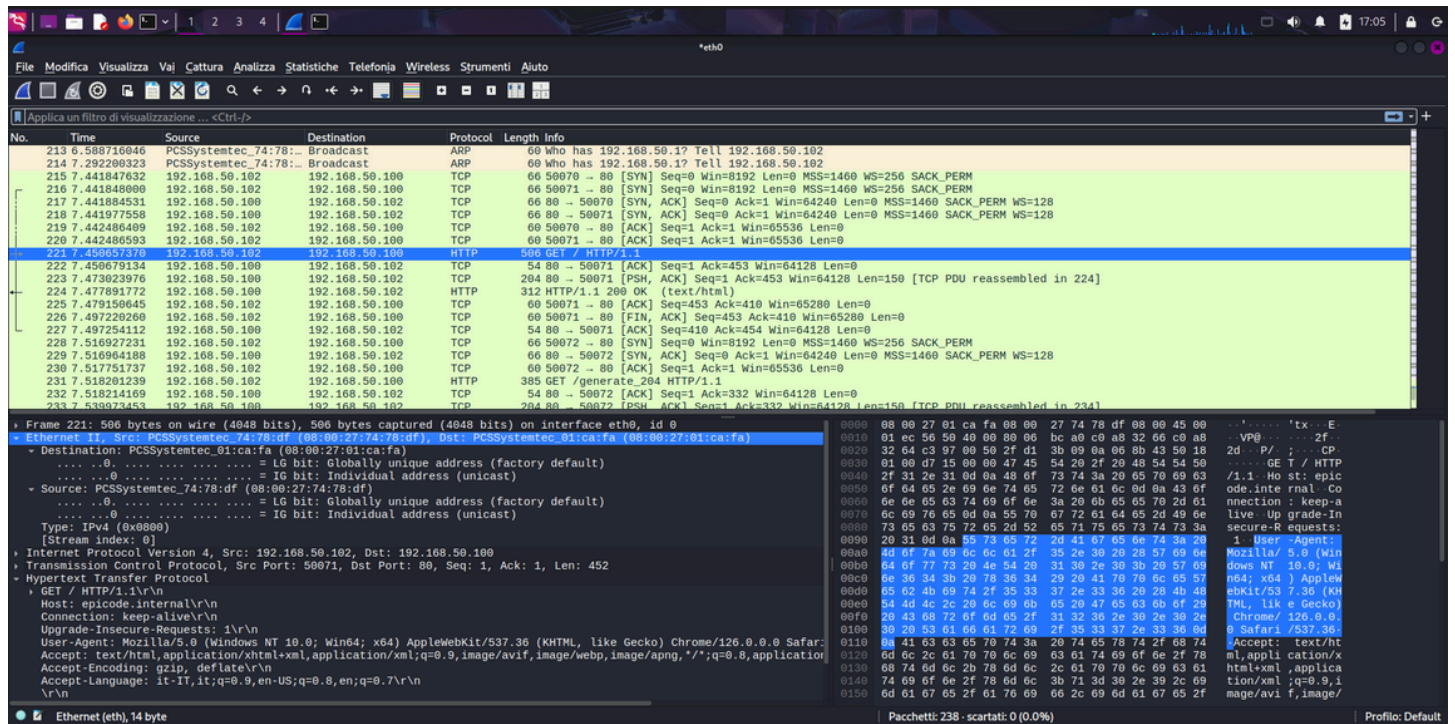
## PROTOCOLLO HTTPS



Possiamo notare come sia possibile vedere il mac address del server ma non della macchina che richiede la connessione ( windows 10 con ip 192.168.50.102)

Tutte le informazioni sui pacchetti trasmessi vengono criptate.

## PROTOCOLLO HTTP



Nel protocollo http sono visibili molte più informazioni come il mac address di entrambe le macchine. I pacchetti non vengono criptati e possiamo visualizzare ulteriori informazioni come sistema operativo, risposta del server.

