

# Progetto M1

*Fabrizio Meini*

## 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.

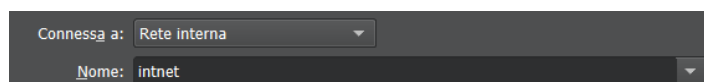
Ripetere l'esercizio, sostituendo il server HTTPS, con un server HTTP. Si intercetti nuovamente il traffico, evidenziando le eventuali differenze tra il traffico appena catturato in HTTP ed il traffico precedente in HTTPS. Spiegare, motivandole, le principali differenze.

## Requisiti e servizi:

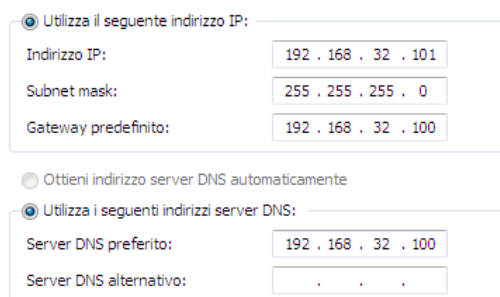
- Kali Linux ☐ IP 192.168.32.100
- Windows 7 ☐ IP 192.168.32.101
- HTTPS server: attivo
- Servizio DNS per risoluzione nomi di dominio: attivo

## Svolgimento:

-Per prima cosa settiamo le schede di rete in INTERNAL e gli indirizzi IP statici in W7 e Kali, facendo attenzione ad impostare l'indirizzo ip di Kali come Gateway e DNS preferito in W7. Altro requisito di partenza è la disattivazione del Firewall in W7 o la creazione di una regola che consenta di scambiare dati con Kali.



Connessa a: Rete interna  
Nome: intnet



☒ Utilizza il seguente indirizzo IP:  
Indirizzo IP: 192 . 168 . 32 . 101  
Subnet mask: 255 . 255 . 255 . 0  
Gateway predefinito: 192 . 168 . 32 . 100  
☐ Ottieni indirizzo server DNS automaticamente  
☒ Utilizza i seguenti indirizzi server DNS:  
Server DNS preferito: 192 . 168 . 32 . 100  
Server DNS alternativo: . . .

```
(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
```

-Attiviamo i servizi DNS,HTTPS ed HTTP (da ROOT) in Kali, utilizzando il VirtualHost INetSim.

```
# Main configuration
#####
# start_service
#
# The services to start
#
# Syntax: start_service <service name>
#
# Default: none
#
# 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
```

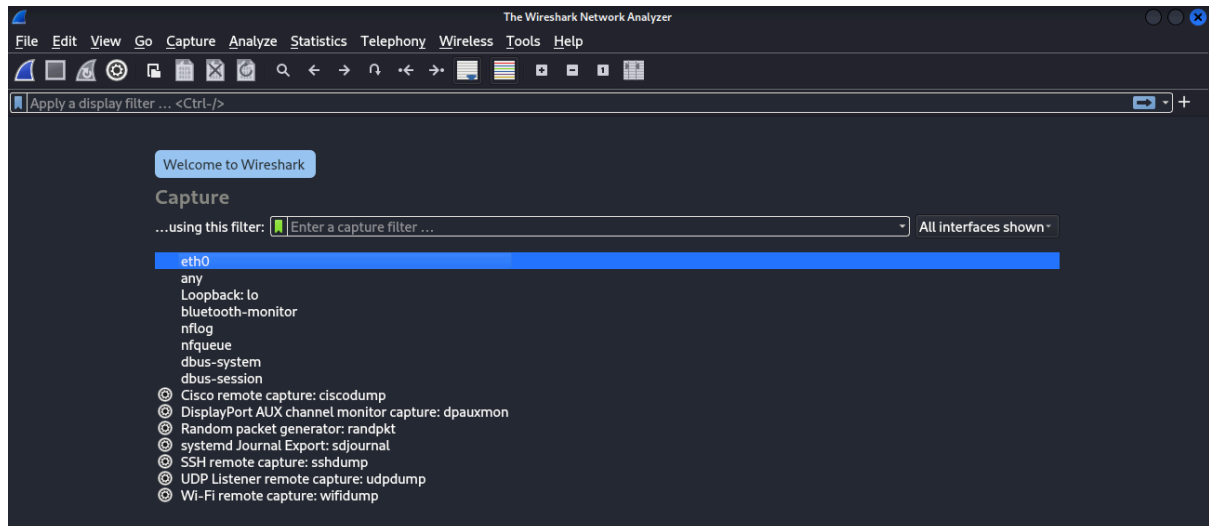
-Impostiamo il default ip del server DNS utilizzando lo stesso ip di Kali, poi facciamo la stessa cosa per risoluzione del dominio “epicode.internal”.

<pre>##### # dns_default_ip # # Default IP address to return with DNS replies # # Syntax: dns_default_ip &lt;IP address&gt; # # Default: 127.0.0.1 # dns_default_ip 192.168.32.100</pre>	<pre>##### # dns_default_domainname # # Default domain name to return with DNS replies # # Syntax: dns_default_domainname &lt;domain name&gt; # # Default: inetsim.org # dns_default_domainname epicode.internal</pre>
--	--

-Ora attiviamo il VirtualHost INetSim (che simulerà i nostri DNS, HTTPS ed HTTP) facendo attenzione ad eseguirlo come amministratore.

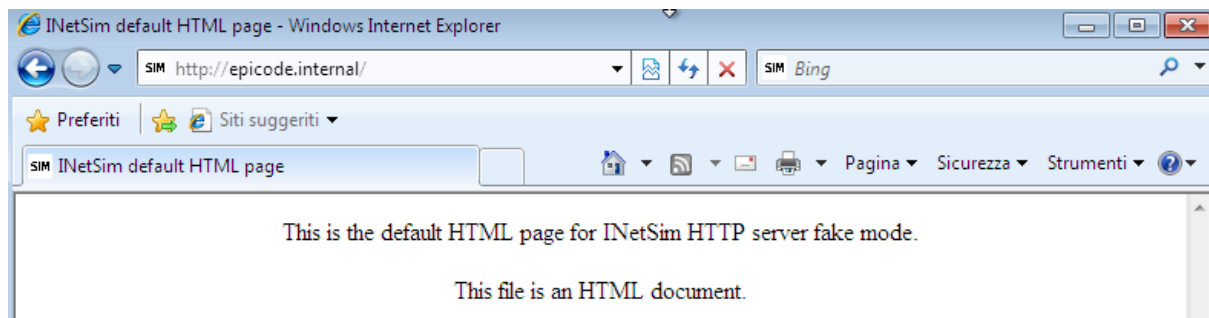
```
(kali㉿kali)-[~]
└─$ sudo inetsim
INetSim 1.3.2 (2020-05-19) by Matthias Eckert & Thomas Hungenberg
Using log directory: /var/log/inetsim/
Using data directory: /var/lib/inetsim/
Using report directory: /var/log/inetsim/report/
Using configuration file: /etc/inetsim/inetsim.conf
Parsing configuration file.
Configuration file parsed successfully.
== INetSim main process started (PID 29267) ==
Session ID: 29267
Listening on: 192.168.32.100
Real Date/Time: 2024-03-16 06:32:00
Fake Date/Time: 2024-03-16 06:32:00 (Delta: 0 seconds)
Forking services...
* dns_53_tcp_udp - started (PID 29277)
print() on closed filehandle MLOG at /usr/share/perl5/Net/DNS/Nameserver.pm line 399.
* http_80_tcp - started (PID 29278)
* https_443_tcp - started (PID 29279)
print() on closed filehandle MLOG at /usr/share/perl5/Net/DNS/Nameserver.pm line 399.
done.
Simulation running.
```

-Attiviamo Wireshark per l'intercettazione dei pacchetti in ricezione su Kali, utilizzando il filtro "eth0", in modo da poter visualizzare i pacchetti in transito.

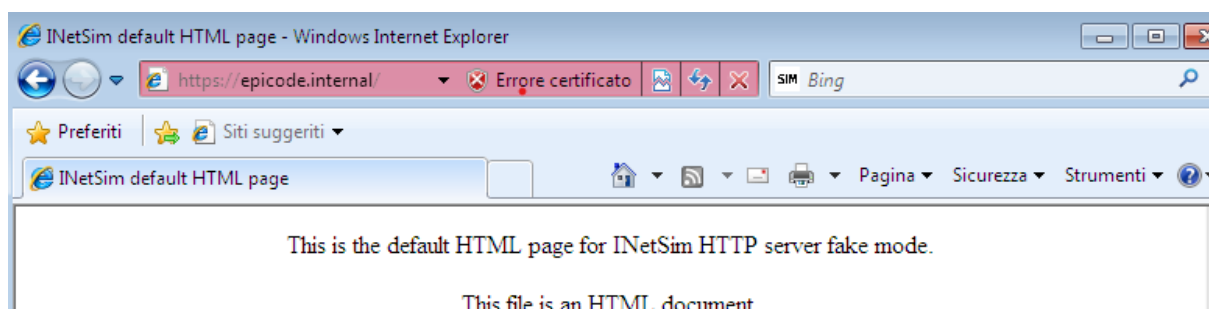


-Da W7, entriamo nel browser e richiamiamo il domain name "epicode.internal", richiedendolo prima in HTTP e poi in HTTPS.

HTTP:

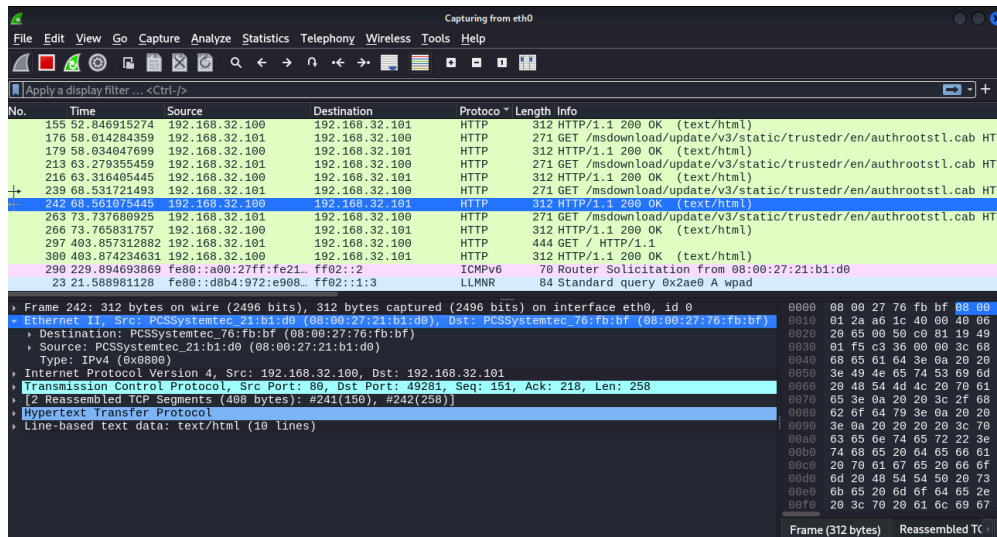


HTTPS:

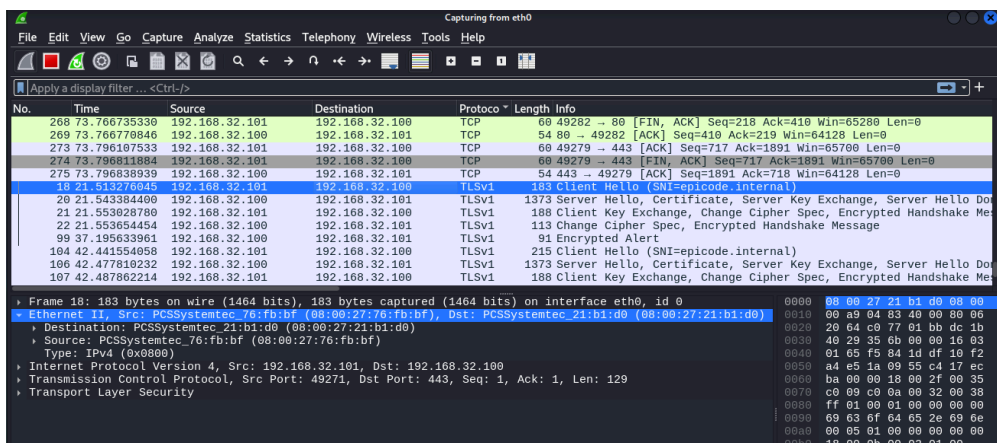


-Infine, torniamo su Kali, e da Wireshark intercettiamo i pacchetti http ed https.

## HTTP:



## HTTPS:



Come possiamo notare, ci sono diverse differenze:

- Nella richiesta HTTP, viene utilizzato il protocollo HTTP, mentre nella richiesta HTTPS, viene utilizzato il protocollo TLSv1
- Per quanto riguarda il contenuto, nei pacchetti HTTP sarà leggibile in quanto non cifrato, a differenza dei pacchetti TLSv1 (dove il testo viene cifrato)
- A causa della cifratura i pacchetti TLSv1 risulteranno più pesanti rispetto ai pacchetti HTTP.

Ci sono però anche similitudini, come nel caso del MAC (destination/source) che risulta lo stesso in entrambi i casi.