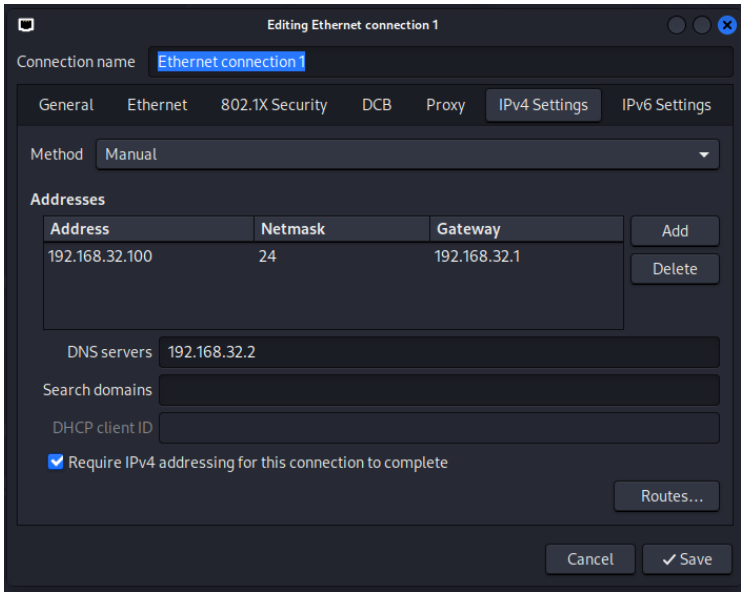
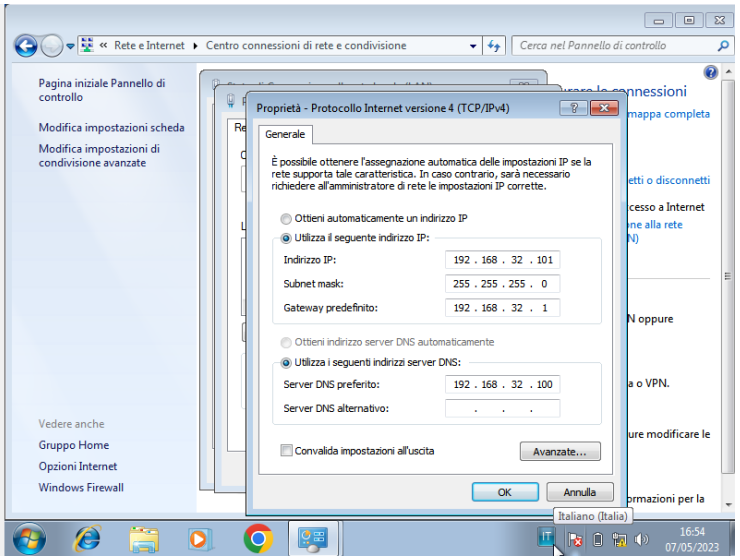
Report delle operazioni per generare ed avviare un server virtuale con tool Inet Sim su Kali Linux e conseguente analisi dei pacchetti scambiati tramite il tool Wireshark

Con questo report innanzitutto andremo a configurare due VM con due IP statici assegnati come segue.
Avviamo le macchine virtuali di WIN7 e Kali Linux su Oracle VM Machine, partendo dalla macchina Kali Linux definita server. Impostazione preliminare è effettuare il cambio della macchina da rete NAT a rete interna nella scheda impostazioni di rete del programma, subito dopo spostiamoci su terminale avviando il comando ***`sudo nano /etc/network/interfaces`*** o attraverso interfaccia GUI cliccando sul pulsante di interfaccia di rete con il tasto destro per poi entrare nel menù edit connection e aggiungendone una nuova modificando la sezione IPV4 settings in questo modo.



Una volta salvate le modifiche riavviamo la macchina per far sì che le stesse vengano attuate immediatamente sul sistema.
Nel frattempo spostiamoci sulla macchina WIN7 definita client ed effettuiamo la stessa operazione entrando nel menù della scheda di rete e nelle relative proprietà modificando la sezione definita <Protocollo internet versione 4 (TCP/IPV4) > come segue ed infine riavviamo.



Report delle operazioni per generare ed avviare un server virtuale con tool Inet Sim su Kali Linux e conseguente analisi dei pacchetti scambiati tramite il tool Wireshark

Torniamo nella macchina Kali Linux che funge da server e configuriamo attraverso il tool preinstallato Inet Sim i relativi dati che ci permetteranno di rendere visibili i file predefiniti chiamando il dominio epicode.internal sia tramite protocollo HTTP che HTTPS oltrechè pingando lo stesso dominio nel terminale del client.
Per entrare nel menu di configurazione di Inet Sim digitiamo nel terminale della macchina Kali Linux il seguente comando ***`sudo nano /etc/inetsim/inetsim.conf`*** e spostiamoci subito nella maschera definita **service_bind_address** e modifichiamola come segue.

```
##########################################
# 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
```

Ulteriori modifiche le faremo come segue nelle maschere **dns_default_ip** e **dns_static**

```
##########################################
# dns_default_ip
#
# Default IP address to return with DNS replies
#
# Syntax: dns_default_ip <IP address>
#
# Default: 127.0.0.1
#
dns_default_ip 192.168.32.100
```

```
##########################################
# dns_static
#
# Static mappings for DNS
#
# Syntax: dns_static <fqdn hostname> <IP address>
#
# Default: none
#
dns_static epicode.internal 192.168.32.100
```

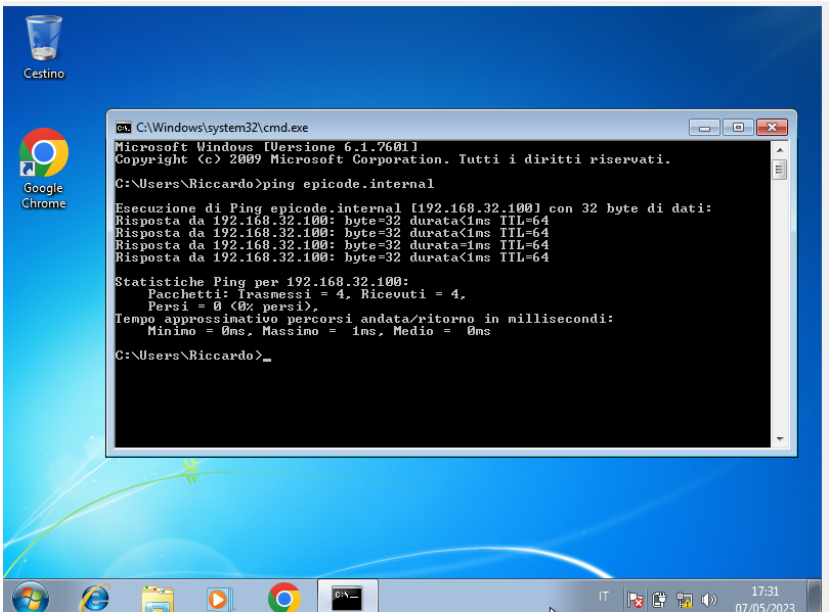
Report delle operazioni per generare ed avviare un server virtuale con tool Inet Sim su Kali Linux e conseguente analisi dei pacchetti scambiati tramite il tool Wireshark

Regola fondamentale, ricordiamo di eliminare l'hashtag per permettere il funzionamento del server DNS di Inet Sim.

Per permettere l'avvio del server DNS dobbiamo inserire nel terminale il comando ***`sudo inetsim`*** e visualizzare la seguente schermata, dopodichè tenendo aperta questa schermata sul terminale andremo ad effettuare i nostri test sul client.

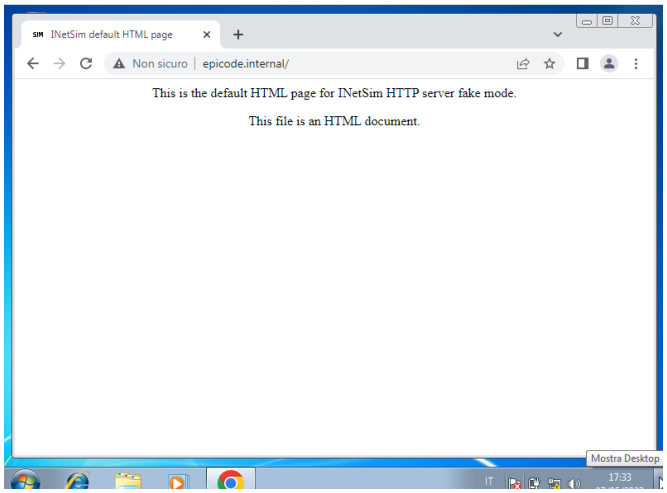
```
┌──(kali㉿kali)-[~]
└─$ sudo inetsim
INetSim 1.3.2 (2020-05-19) by Matthias Eckert & Thomas Hungenberg
Using log directory:      /var/log/inetsim/
Using data directory:     /var/lib/inetsim/
Using report directory:   /var/log/inetsim/report/
Using configuration file: /etc/inetsim/inetsim.conf
Parsing configuration file.
Warning: Unknown option ']############################################################' in configuration file '/etc/inetsim/inetsim.conf' line 1
Configuration file parsed successfully.
=== INetSim main process started (PID 21767) ===
Session ID:     21767
Listening on:   192.168.32.100
Real Date/Time: 2023-05-07 11:16:34
Fake Date/Time: 2023-05-07 11:16:34 (Delta: 0 seconds)
 Forking services...
  * irc_6667_tcp - started (PID 21787)
  * dns_53_tcp_udp - started (PID 21777)
  * ident_113_tcp - started (PID 21790)
  * ntp_123_udp - started (PID 21788)
  * smtps_465_tcp - started (PID 21781)
  * quotd_17_udp - started (PID 21801)
  * pop3s_995_tcp - started (PID 21783)
  * echo_7_udp - started (PID 21797)
  * time_37_tcp - started (PID 21792)
  * chargen_19_udp - started (PID 21803)
  * finger_79_tcp - started (PID 21789)
  * https_443_tcp - started (PID 21779)
  * tftp_69_udp - started (PID 21786)
  * discard_9_tcp - started (PID 21798)
  * ftps_990_tcp - started (PID 21785)
  * quotd_17_tcp - started (PID 21800)
  * pop3_110_tcp - started (PID 21782)
  * echo_7_tcp - started (PID 21796)
  * discard_9_udp - started (PID 21799)
  * smtp_25_tcp - started (PID 21780)
  * dummy_1_udp - started (PID 21805)
  * time_37_udp - started (PID 21793)
  * daytime_13_tcp - started (PID 21794)
  * syslog_514_udp - started (PID 21791)
  * http_80_tcp - started (PID 21778)
  * ftp_21_tcp - started (PID 21784)
  * daytime_13_udp - started (PID 21795)
  * chargen_19_tcp - started (PID 21802)
  * dummy_1_tcp - started (PID 21804)
 done.
Simulation running.
```

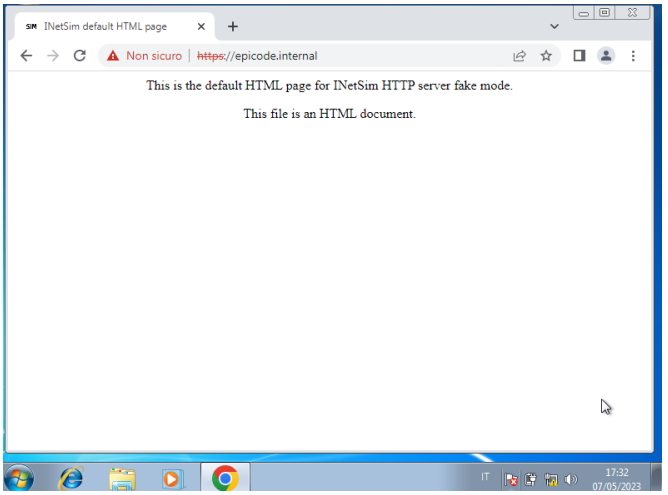
Adesso spostiamo la nostra attenzione sul client, quindi sulla macchina di WIN7. Dapprima pinghiamo in riga di comando su WIN7 il server quindi l'IP 192.168.32.100 e verifichiamo l'effettiva raggiungibilità dello stesso, dopodichè pinghiamo direttamente il record assegnato epicode.internal, se tutto sarà correttamente configurato avremo la seguente schermata nel terminale.



Report delle operazioni per generare ed avviare un server virtuale con tool Inet Sim su Kali Linux e conseguente analisi dei pacchetti scambiati tramite il tool Wireshark

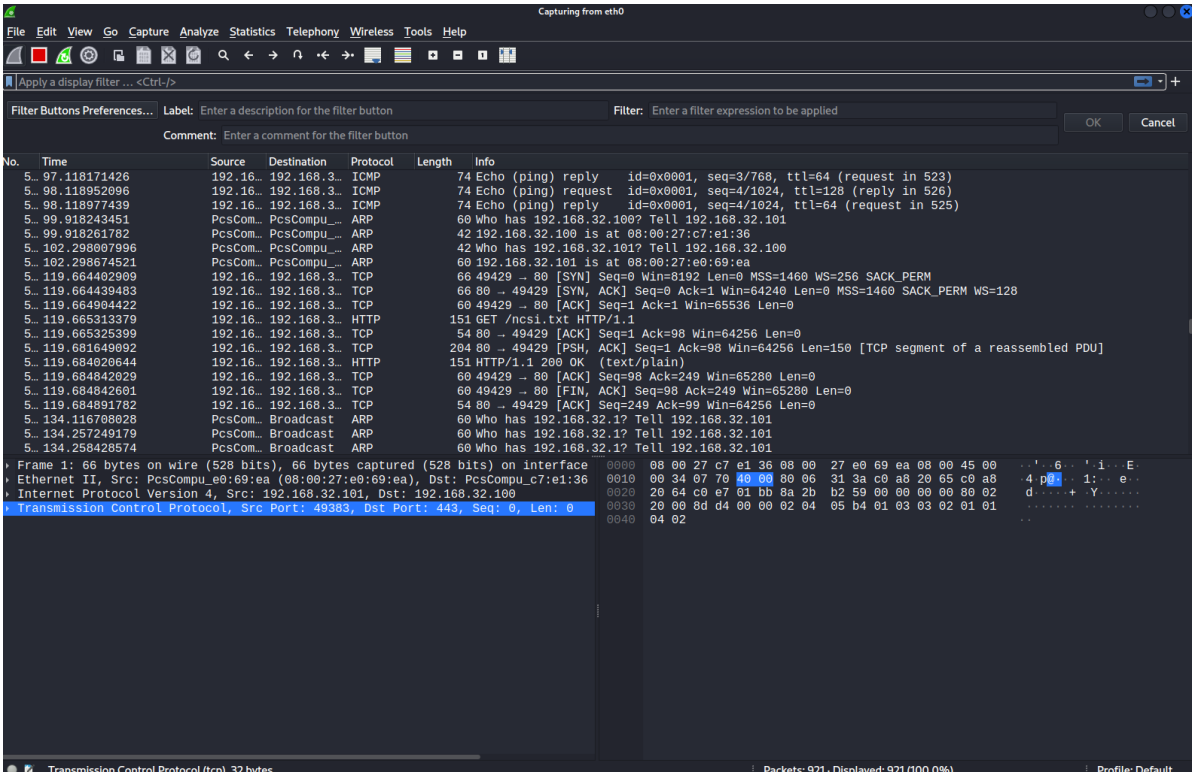
Sempre dal client apriamo una schermata del browser e dapprima richiamiamo il record con il protocollo HTTP e dopo con il protocollo HTTPS.



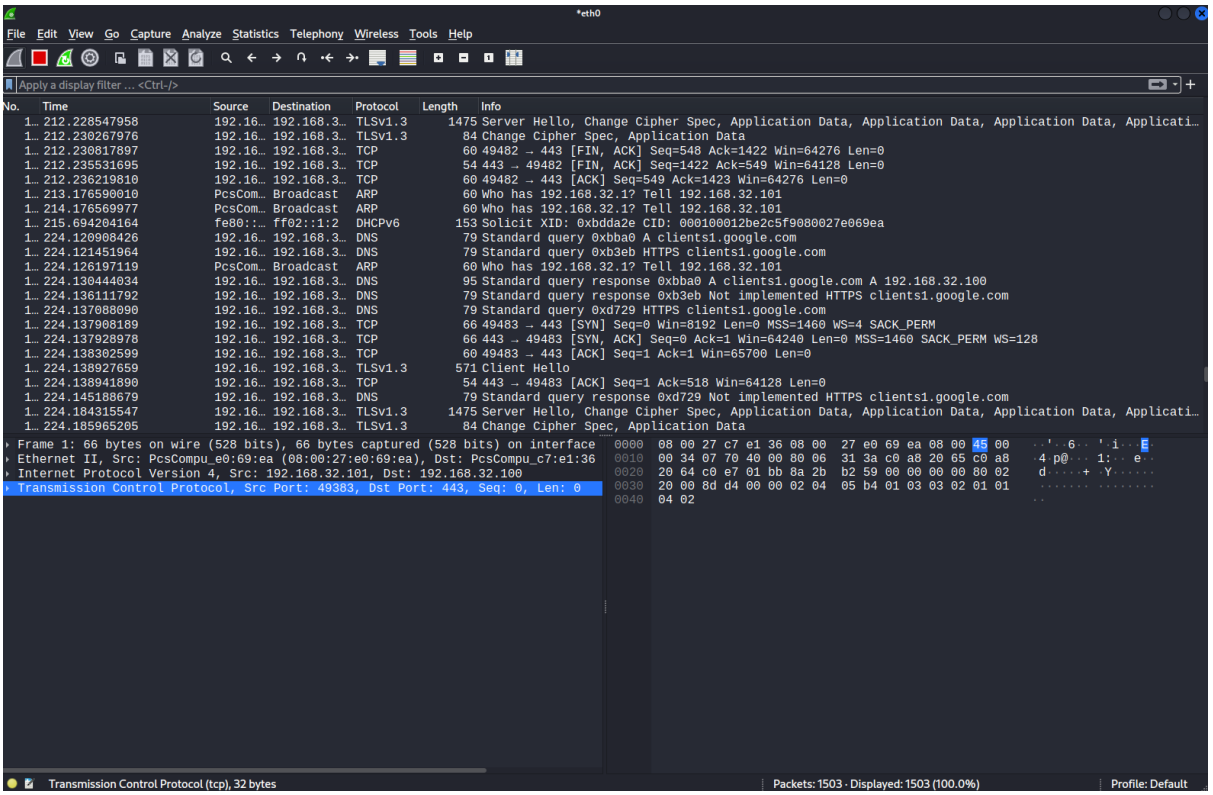


La principale differenza che viene evidenziata nelle due schermate è che sotto il protocollo HTTP il browser non avvisa l'utente dei potenziali rischi al quale è esposto navigando sulla pagina aperta, mentre sotto protocollo HTTPS il browser registra il dominio come non sicuro in quanto assieme allo stesso non è presente alcun certificato digitale che rende sicura la navigazione dell'utente sotto l'utilizzo del protocollo HTTPS.

Come ultimo passaggio del nostro report andiamo ad analizzare i vari pacchetti scambiati tra client e server attraverso l'utilizzo di Wireshark su Kali Linux.



Report delle operazioni per generare ed avviare un server virtuale con tool Inet Sim su Kali Linux e conseguente analisi dei pacchetti scambiati tramite il tool Wireshark



Dalle seguenti catture possiamo notare la differenza quando analizziamo i pacchetti con Wireshark, quello che subito salta all'occhio è che quando dal browser del client chiamiamo un server http, lo stesso è visualizzabile nella sezione protocollo di Wireshark, mentre quando chiamiamo un server HTTPS dal browser del client questo non appare nella sezione protocollo in quanto essendo una connessione verificata non è possibile visualizzarlo nella sezione, visto che si tratta comunque di una simulazione virtuale.

Oltre a ciò possiamo notare il continuo scambio tra server e client di pacchetti diversi come per esempio il pacchetto ARP che vuole conoscere il mac address del dell'host dell'indirizzo IP, o il pacchetto ICMP che trasmette informazioni riguardo lo stato del server (richiamabile tramite il comando di ping dal terminale) e per terminare ad esempio il pacchetto TCP che si occupa di controllare la trasmissione dei dati fra mittente e destinatario.

In conclusione attraverso questo report ho voluto dimostrare innanzitutto come è possibile simulare l'utilizzo di un server attraverso il tool Inet Sim, ma soprattutto la differenza che vi è a livello di trasmissione dei dati tramite il protocollo HTTP e HTTPS servendomi ovviamente di uno strumento molto valido quale Wireshark che mi ha permesso di capire quali possono essere le problematiche a livello di composizione di un pacchetto, di scambio degli stessi ed eventualmente la risoluzione delle stesse.