

WEEKLY PROJECT REPORT

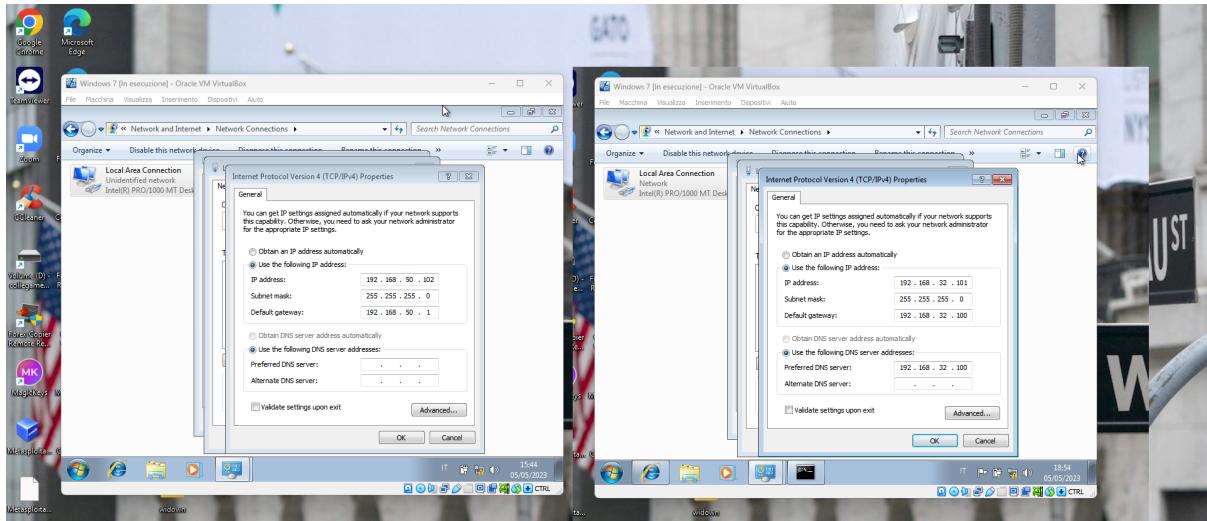
“SIMULAZIONE RETE COMPLESSA”

La prova di simulazione di una rete complessa è suddivisa, come da traccia, in diversi step da seguire per far sì che l'elaborato finale funzioni correttamente.

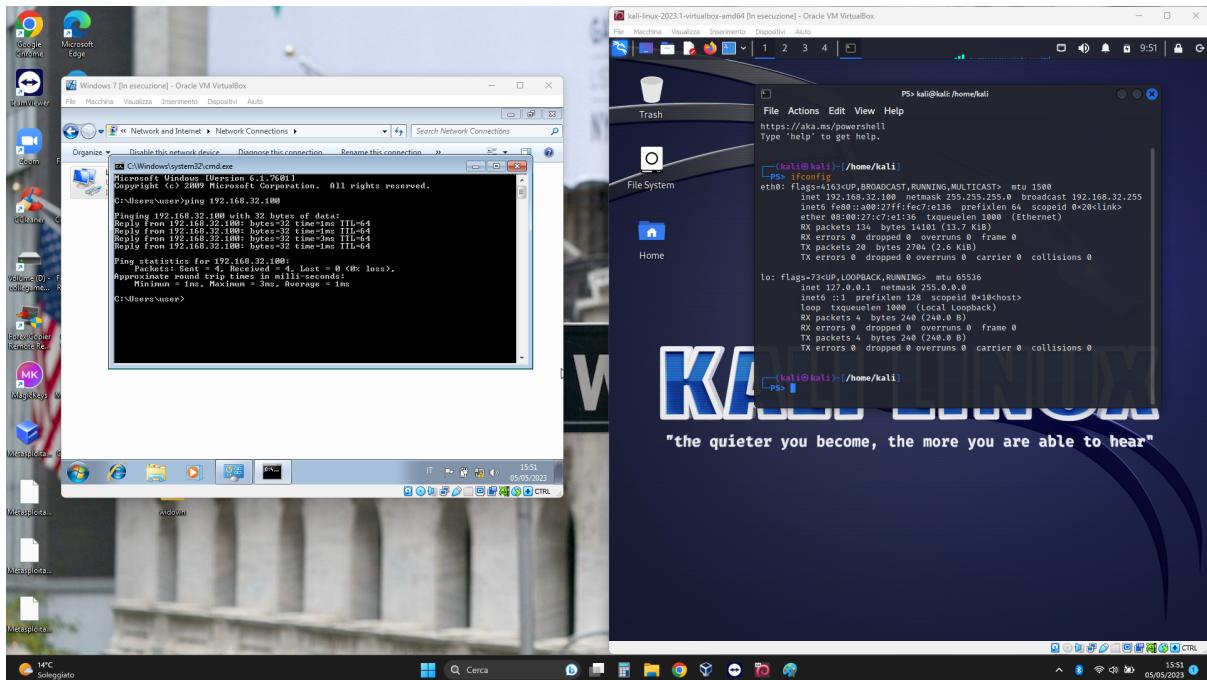
1. Il primo step consisteva nell'assegnazione ai client degli indirizzi IP statici. Nello specifico prevedeva che al client con Kali Linux venisse assegnato l'indirizzo 192.168.32.100 mentre al client con Windows l'indirizzo 192.168.32.101.

Dopo aver effettuato questo primo passaggio si va ad effettuare una prova di ping tra le due macchine per verificare il corretto esito dell'operazione.

Di seguito si riportano gli screen delle operazioni effettuate.



Prima e dopo Windows

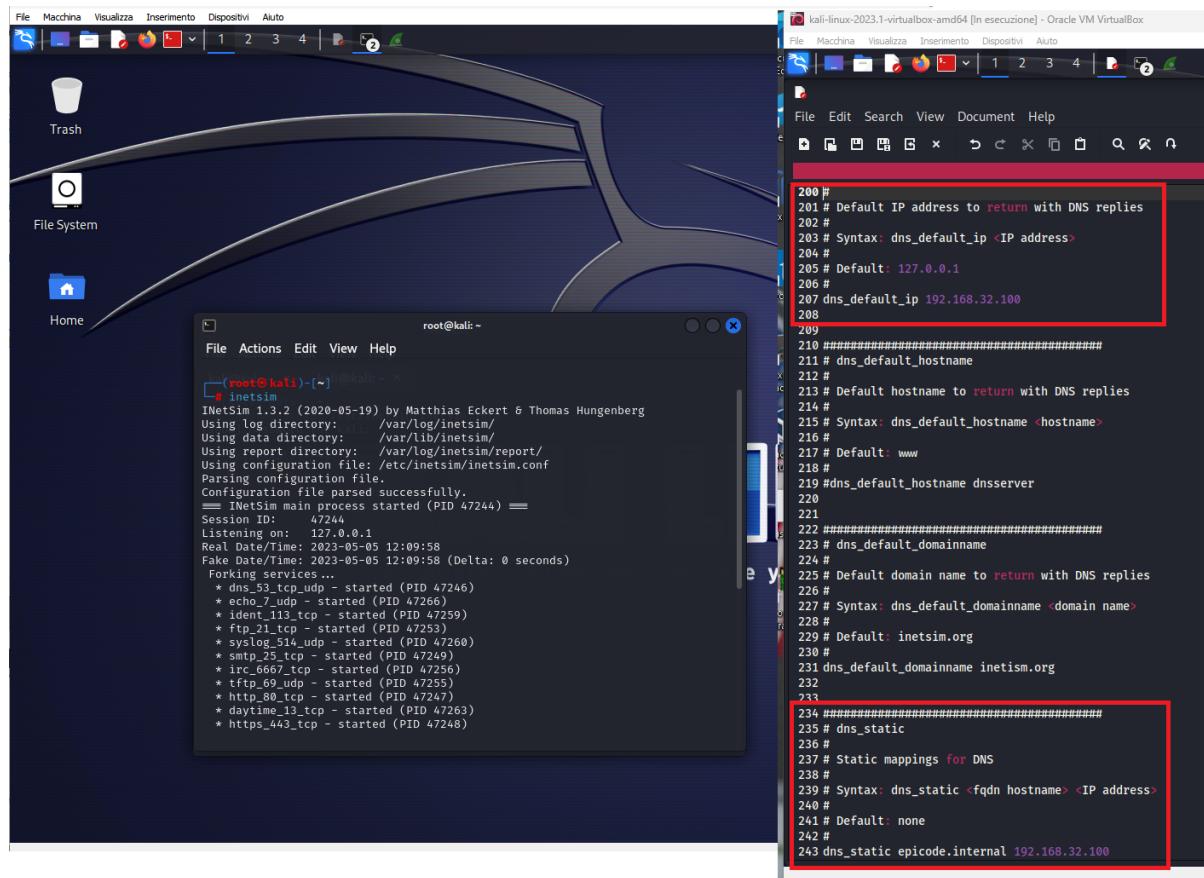


Nuovo IP Kali + prova ping

2. Il secondo step prevedeva invece l'installazione del tool InetSim, il quale si occupa di simulare i servizi internt su macchina Linux, e poi avviare tramite quest ultimo un server DNS e un server HTTPS.

Inoltre si doveva anche modificare il server DNS in modo tale che rispondesse al comando `epicode.internal`.

Di seguito si riporta lo screen delle operazioni effettuate.



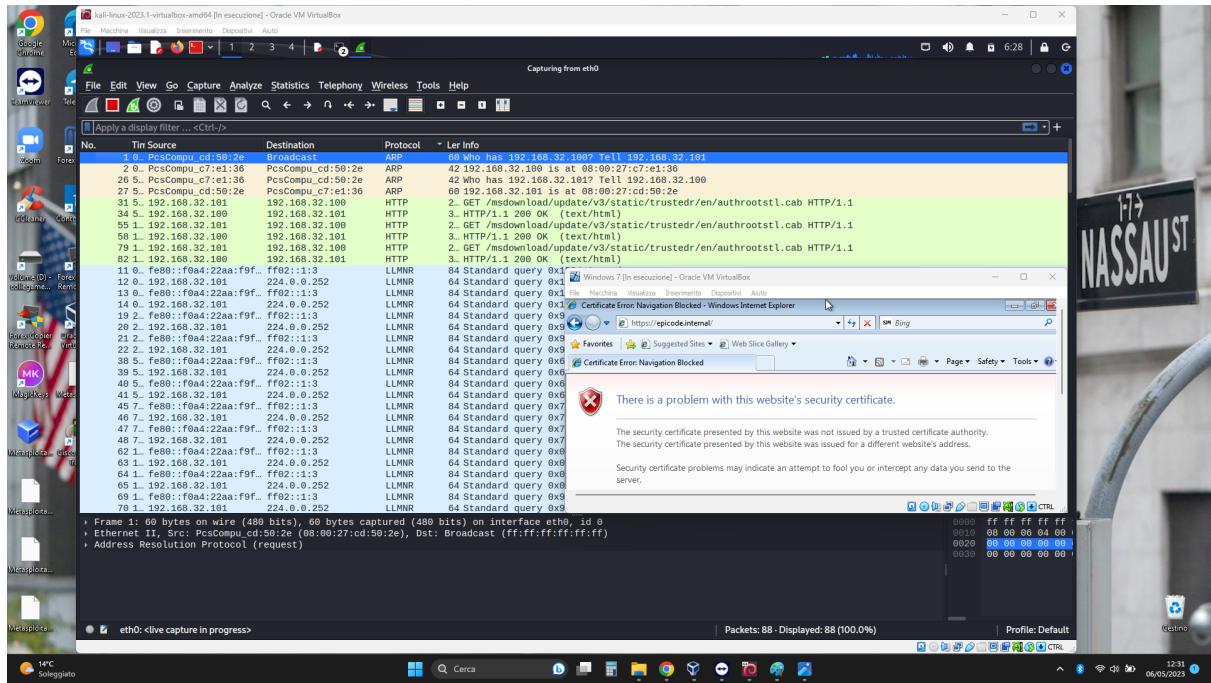
```
root@kali:~# inetsim
InetSim 1.3.2 (2020-05-19) by Matthias Eckert & Thomas Hungenberg
Using log directory: /var/log/inetsim/
Using data directory: /var/lib/inetsim/
Using report directory: /var/log/inetsim/report/
Using configuration file: /etc/inetsim/inetsim.conf
Parsing configuration file.
Configuration file parsed successfully.
= InetSim main process started (PID 47244) =
Session ID: 47244
Listening on: 127.0.0.1
Real Date/time: 2023-05-05 12:09:58
Fake Date/time: 2023-05-05 12:09:58 (Delta: 0 seconds)
Forking services
* dns_53_tcp_udp - started (PID 47246)
* echo_7_udp - started (PID 47266)
* ident_113_tcp - started (PID 47259)
* ftp_21_tcp - started (PID 47253)
* syslog_514_udp - started (PID 47268)
* smtp_25_tcp - started (PID 47249)
* irc_6667_tcp - started (PID 47256)
* tftp_69_udp - started (PID 47255)
* http_80_tcp - started (PID 47247)
* daytime_13_tcp - started (PID 47263)
* https_443_tcp - started (PID 47248)

root@kali:~# cat /etc/dns/dns.conf
200 #
201 # Default IP address to return with DNS replies
202 #
203 # Syntax: dns_default_ip <IP address>
204 #
205 # Default: 127.0.0.1
206 #
207 dns_default_ip 192.168.32.100
208
209
210 #####
211 # dns_default_hostname
212 #
213 # Default hostname to return with DNS replies
214 #
215 # Syntax: dns_default_hostname <hostname>
216 #
217 # Default: www
218 #
219 #dns_default_hostname dnsserver
220
221
222 #####
223 # dns_default_domainname
224 #
225 # Default domain name to return with DNS replies
226 #
227 # Syntax: dns_default_domainname <domain name>
228 #
229 # Default inetsim.org
230 #
231 dns_default_domainname inetsim.org
232
233
234 #####
235 # dns_static
236 #
237 # Static mappings for DNS
238 #
239 # Syntax: dns_static <fqdn hostname> <IP address>
240 #
241 # Default: none
242 #
243 dns_static epicode.internal 192.168.32.100
```

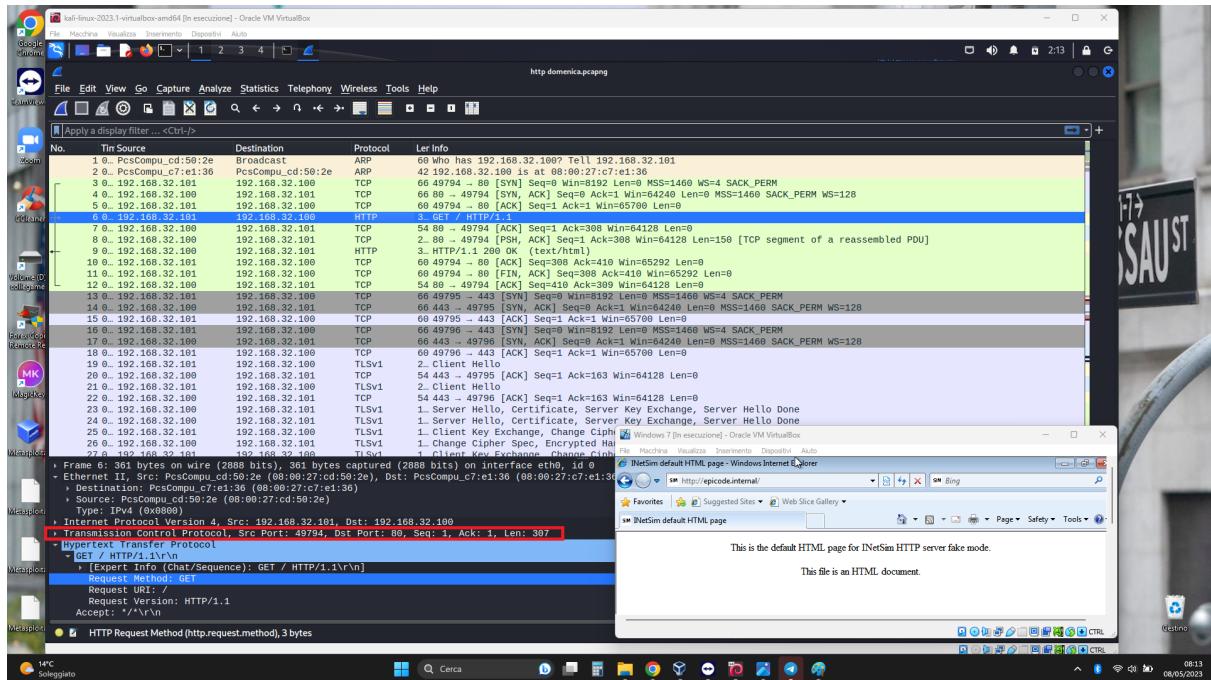
3. Il terzo ed ultimo step prevedeva invece di catturare tramite il tool Wireshark (su client Kali Linux), che è un software per l'analisi di protocolli (o packet sniffer), i MAC address del client sorgente e quello destinatario e il contenuto della richiesta fatta da client Windows prima con il protocollo HTTPS e poi con quello HTTP.

Nel passaggio da protocollo HTTPS a quello HTTP è stato anche necessario modificare il file .conf del server.

Di seguito si riportano gli screen delle operazioni effettuate.



Richiesta risoluzione di episode.internal con protocollo HTTPS su client Windows + software Wireshark (MAC address dei due client in giallo, richieste HTTPS in verde).



Richiesta di risoluzione episode.internal con protocollo HTTP su client Windows + software Wireshark (MAC address dei due client in giallo, richieste HTTPS in verde).

Conclusioni

Dopo aver completato tutti gli step ed aver avuto conferma del funzionamento della rete, si è potuto notare come utilizzando il protocollo HTTPS (che consiste nella comunicazione tramite il protocollo HTTP all'interno di una connessione criptata), andando ad effettuare un'analisi dei pacchetti nel software Wireshark si nota come sia la porta 443 dedicata per l'appunto a tale protocollo, a essere utilizzata per lo scambio dei dati.

Utilizzando il protocollo HTTP si nota invece che quella utilizzata questa volta è la porta 80, e osservando meglio il pacchetto nel riquadro in basso sia presente "in chiaro" nella sezione HyperText transfer protocol tutto il contenuto della pagina richiesta.

Inoltre quando si utilizza il protocollo HTTPS il browser va a bloccare l'esecuzione della ricerca segnalando all'utente che c'è un problema con il certificato di sicurezza del sito; utilizzando invece il protocollo HTTP, il browser va ad aprire tranquillamente il collegamento richiesto non garantendo così all'utente un adeguato grado di protezione.

Si consiglia di conseguenza l'utilizzo del protocollo HTTPS per la creazione di qualsiasi elaborato web.