

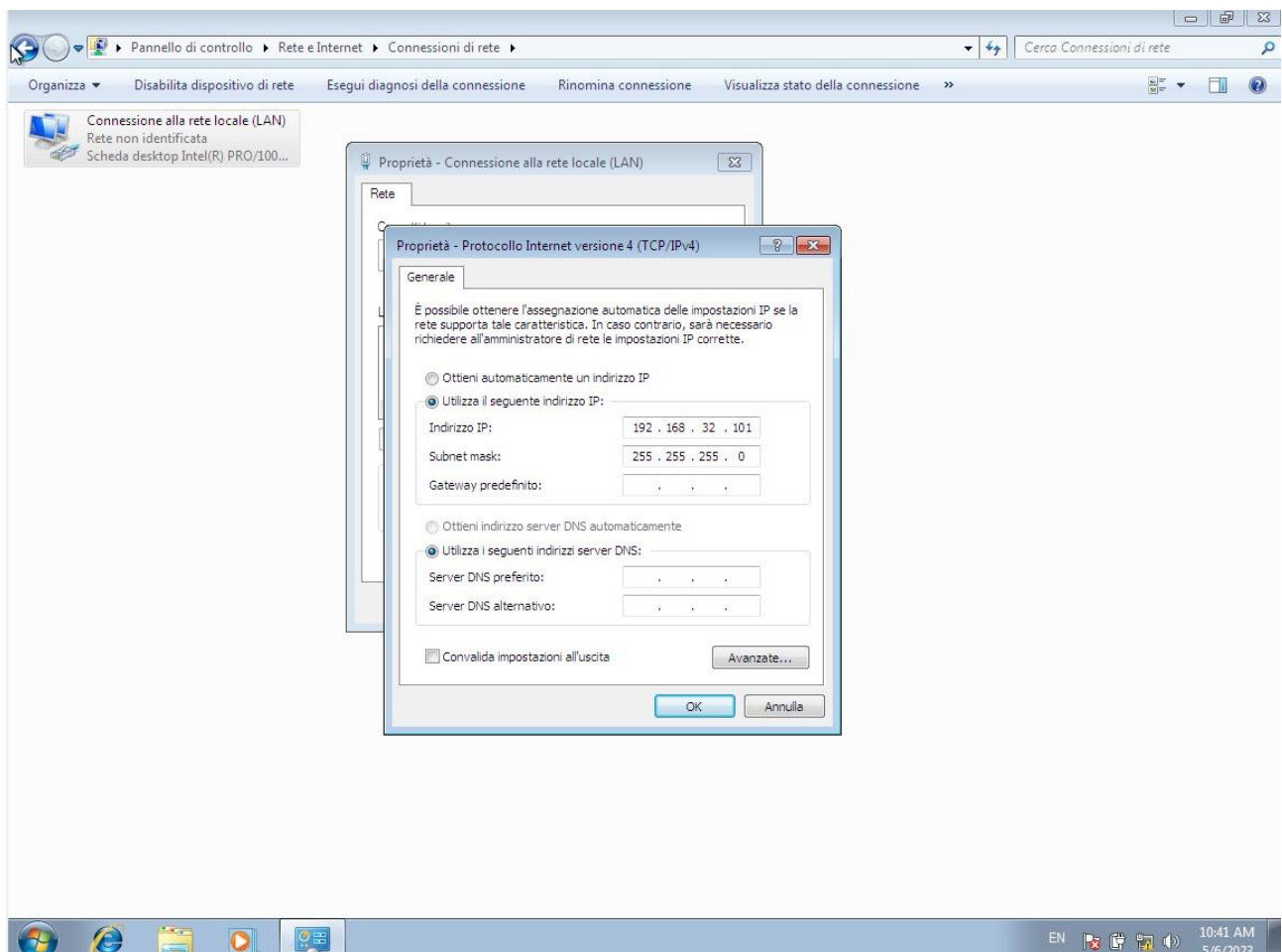
ESERCIZIO WEEK 1

1. IMPOSTARE I NUOVI INDIRIZZI IP KALI/WINDOWS

Da WINDOWS, cliccando con il mouse destro sull'icona della rete il percorso da seguire è

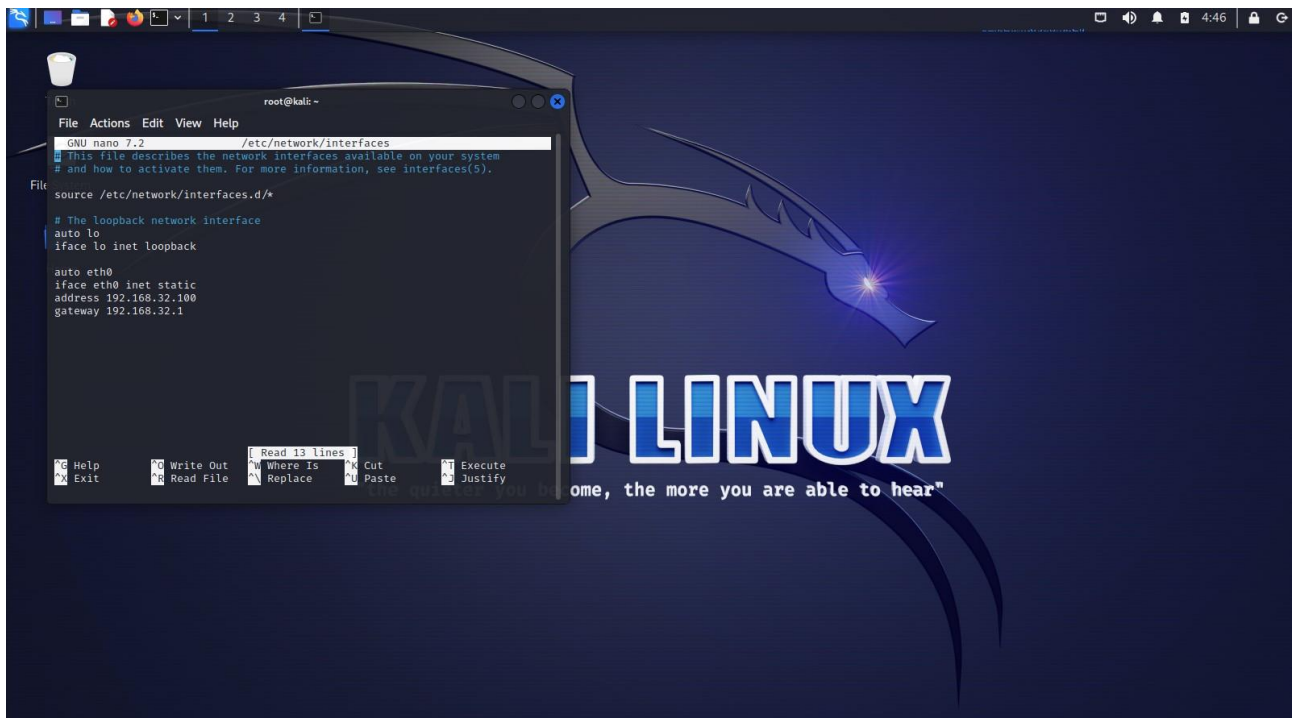
- Apri centro connessioni di rete e condivisioni
- Modifica impostazioni scheda
- Connessione alla rete locale lan (clic destro→proprietà)
- Protocollo internet versione 4 (proprietà)

E da qui si modifica l'indirizzo in 192.168.32.100 con subnet mask 255.255.255.0



Da KALI, cliccando sul terminale in alto digitiamo nella riga di comando “nano /etc/network/interfaces”

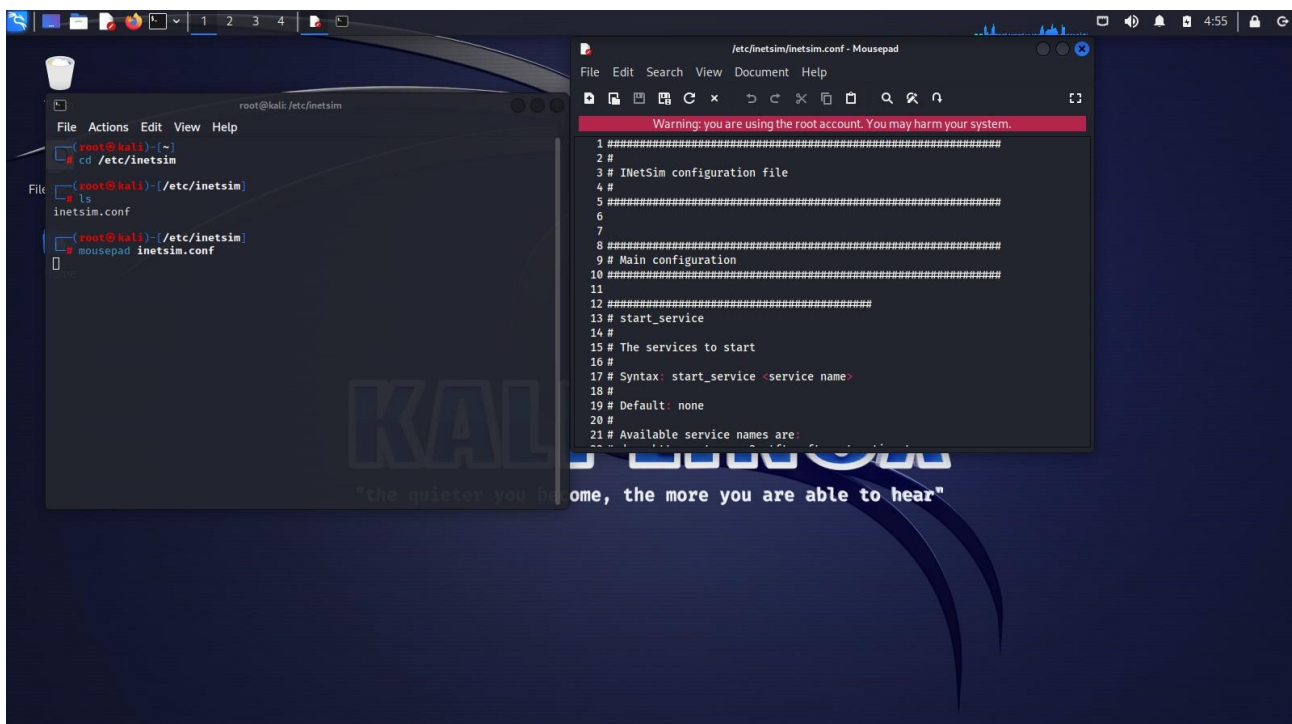
E supponendo di aver già aggiunto le righe di ip statico da precedente esercizio (auto eth0...), si modifica semplicemente l'indirizzo ip in 192.168.32.101 e il gateway in 192.168.32.1



2. Avviare server DNS e HTTPS su Kali e configurare il DNS

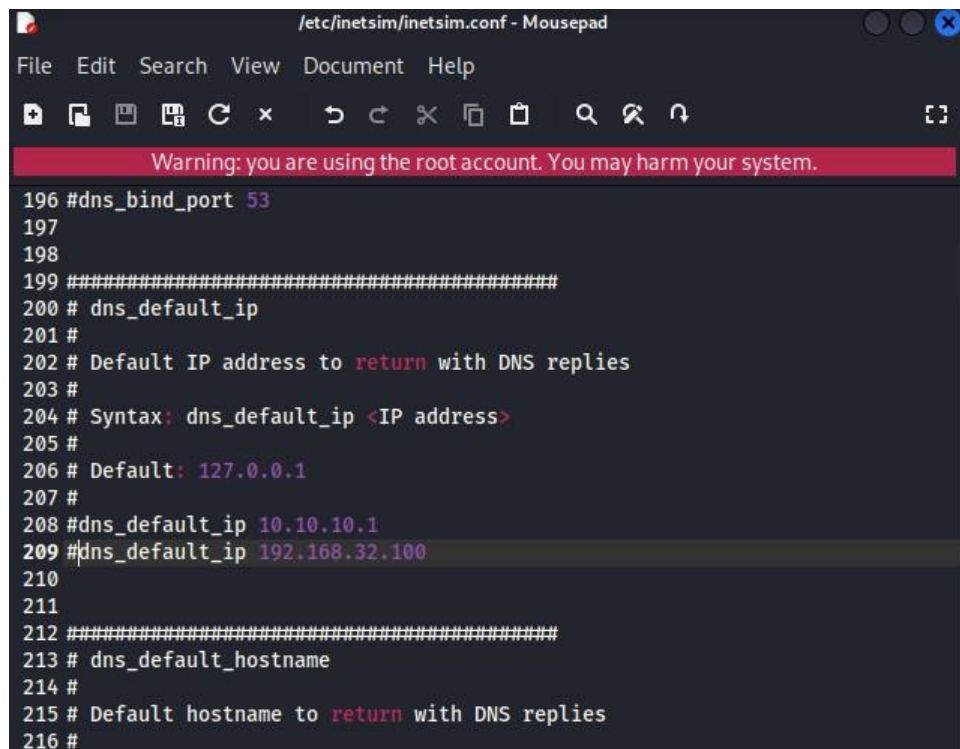
Per poter eseguire questo passaggio abbiamo bisogno di avviare “inetsim”. L’applicativo è già installato sulla macchina virtuale. Digitando nel terminale direttamente inetsim partirà la simulazione avviando i server DNS e HTTPS, ma prima abbiamo bisogno di modificare le impostazioni del file di configurazione.

Dal terminale andiamo a eseguire il comando “cd /etc/inetsim” e di seguito con “ls” possiamo vedere che la cartella contiene il file “inetsim.conf”, per aprirlo digitiamo “mousepad inetsim.conf”



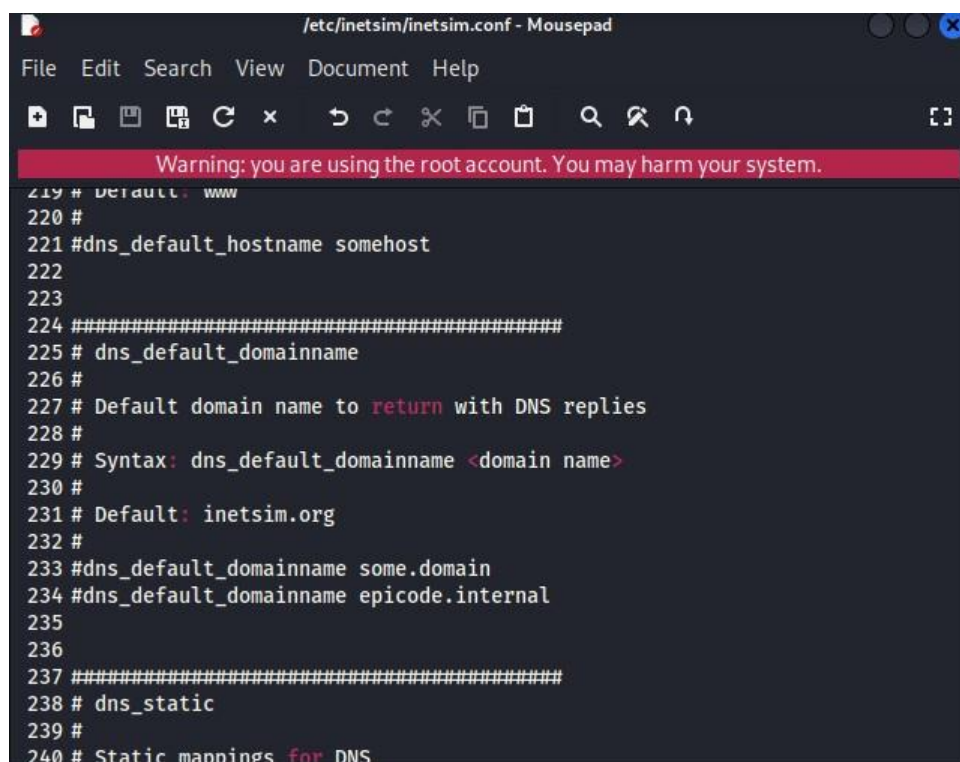
Questo comando non farà altro che aprire l'editor di testo per poter modificare le configurazioni.

Andiamo quindi a cercare la riga "dns_default_ip" e ne aggiungiamo un'altra con l'indirizzo ip di KALI appena impostato 192.168.32.100



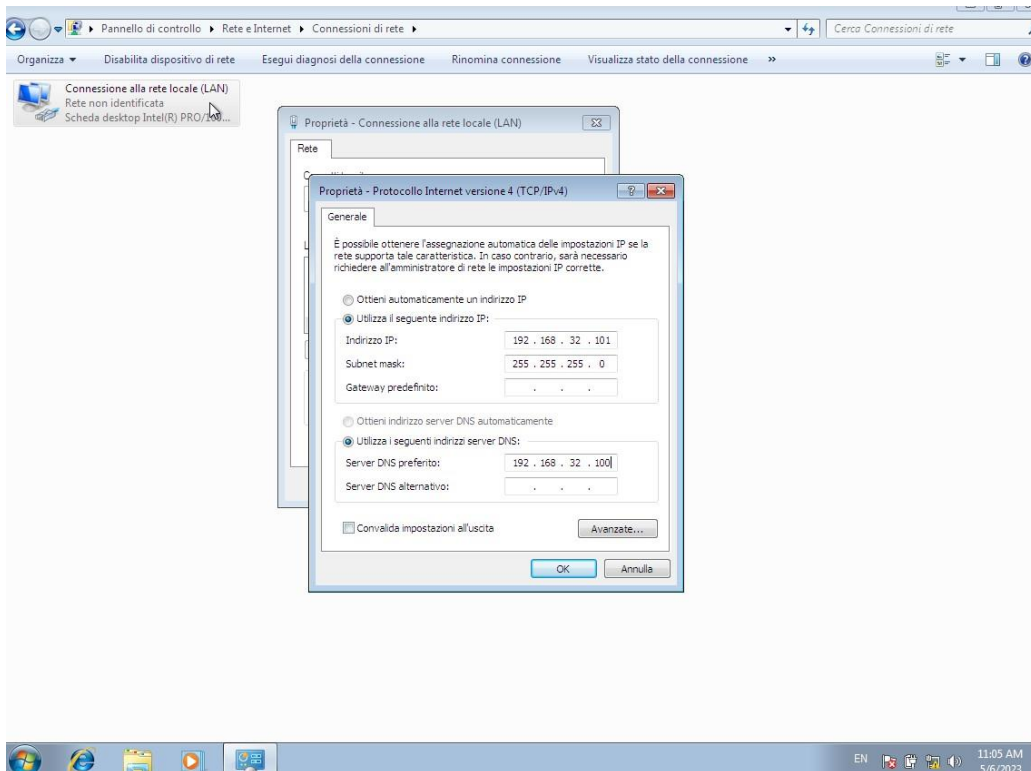
```
/etc/inetsim/inetsim.conf - Mousepad
File Edit Search View Document Help
Warning: you are using the root account. You may harm your system.
196 #dns_bind_port 53
197
198
199 #####
200 # dns_default_ip
201 #
202 # Default IP address to return with DNS replies
203 #
204 # Syntax: dns_default_ip <IP address>
205 #
206 # Default: 127.0.0.1
207 #
208 #dns_default_ip 10.10.10.1
209 #dns_default_ip 192.168.32.100
210
211
212 #####
213 # dns_default_hostname
214 #
215 # Default hostname to return with DNS replies
216 #
```

Poco più giù andiamo ad impostare anche il domain name su epicode.internal aggiungendo l'apposita riga

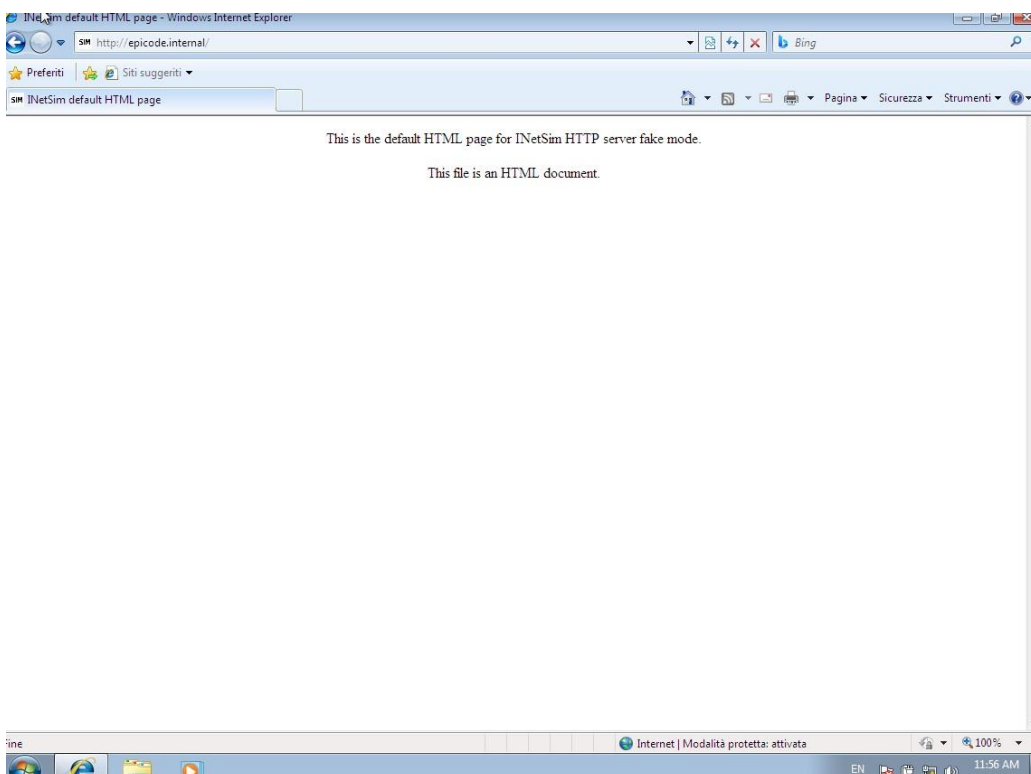


```
/etc/inetsim/inetsim.conf - Mousepad
File Edit Search View Document Help
Warning: you are using the root account. You may harm your system.
219 # Default: www
220 #
221 #dns_default_hostname somehost
222
223
224 #####
225 # dns_default_domainname
226 #
227 # Default domain name to return with DNS replies
228 #
229 # Syntax: dns_default_domainname <domain name>
230 #
231 # Default: inetsim.org
232 #
233 #dns_default_domainname some.domain
234 #dns_default_domainname epicode.internal
235
236
237 #####
238 # dns_static
239 #
240 # Static mappings for DNS
```

Fatta questa operazione abbiamo bisogno che WINDOWS utilizzi questo DNS per risolvere il nome dell'host, quindi dalla finestra delle impostazioni della scheda, sempre sul protocollo IPv4, si aggiunge l'ip di KALI sui server DNS



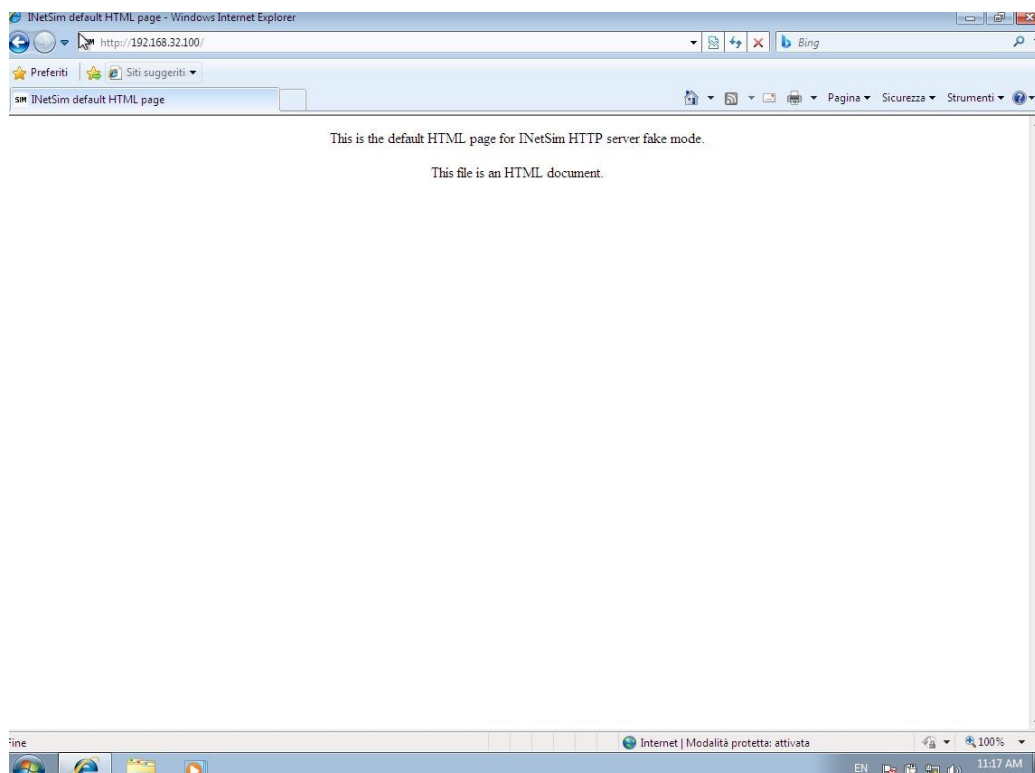
Dal web browser di WINDOWS possiamo verificare che il sito è raggiungibile tramite il dominio scelto



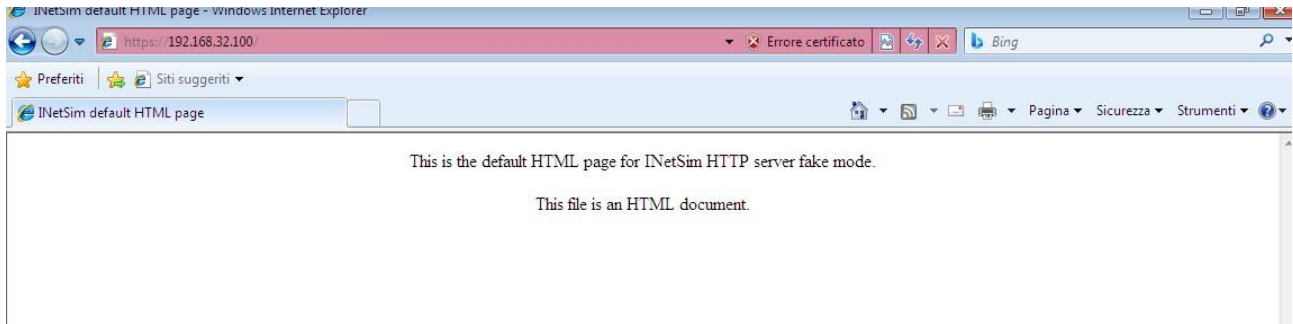
A questo punto si può far partire la simulazione dal terminale KALI semplicemente digitando “inetsim” che avvierà tutti i servizi.

```
root@kali: ~  
File Actions Edit View Help  
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 165018) ==  
Session ID: 165018  
Listening on: 0.0.0.0  
Real Date/Time: 2023-05-06 05:14:39  
Fake Date/Time: 2023-05-06 05:14:39 (Delta: 0 seconds)  
Forking services ...  
* dns_53_tcp_udp - started (PID 165038)  
* irc_6667_tcp - started (PID 165048)  
* syslog_514_udp - started (PID 165052)  
* ident_113_tcp - started (PID 165051)  
* time_37_udp - started (PID 165054)  
* ntp_123_udp - started (PID 165049)  
* finger_79_tcp - started (PID 165050)  
* time_37_tcp - started (PID 165053)  
* daytime_13_udp - started (PID 165056)  
* daytime_13_tcp - started (PID 165055)  
* echo_7_tcp - started (PID 165057)  
* discard_9_tcp - started (PID 165059)  
* discard_9_udp - started (PID 165060)  
* echo_7_udp - started (PID 165058)  
* tftp_69_udp - started (PID 165047)  
* smtp_25_tcp - started (PID 165041)  
* quotd_17_udp - started (PID 165062)  
* pop3s_995_tcp - started (PID 165044)  
* chargen_19_tcp - started (PID 165063)  
* smtps_465_tcp - started (PID 165042)  
* ftp_21_tcp - started (PID 165045)  
* dummy_1_tcp - started (PID 165065)  
* quotd_17_tcp - started (PID 165061)  
* chargen_19_udp - started (PID 165064)  
* dummy_1_udp - started (PID 165066)  
* ftps_990_tcp - started (PID 165046)  
* https_443_tcp - started (PID 165040)  
* pop3_110_tcp - started (PID 165043)  
* http_80_tcp - started (PID 165039)  
done.  
Simulation running.
```

Per verificare che il server http/https sia funzionante, da WINDOWS utilizzando il web browser internet explorer possiamo raggiungere il sito fake inetsim digitando nella barra degli indirizzi l’indirizzo IP o il dominio di KALI.

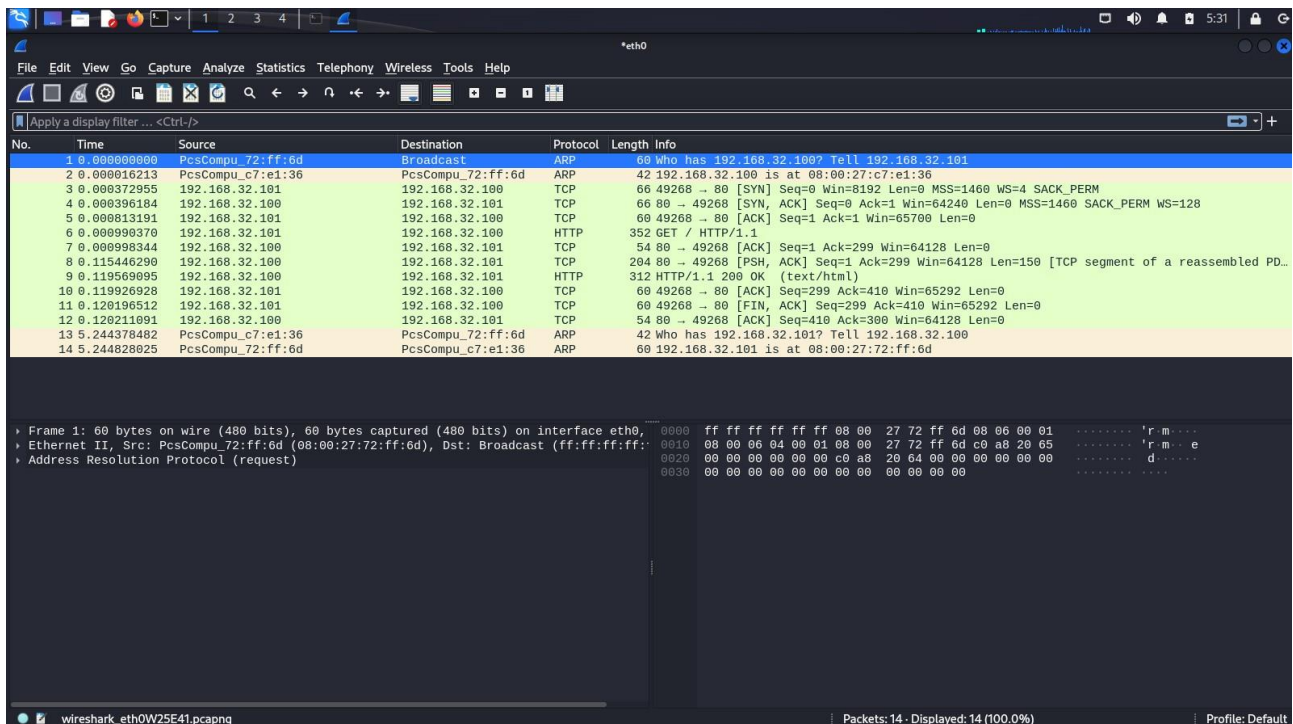


Effettuando la connessione https dar  ovviamente errore per una questione di certificati, ma ignorando l’alert e proseguendo sar  comunque possibile raggiungere il sito.



3. WIRESHARK

Avviamo Wireshark direttamente dagli strumenti di sniffing & spoofing di KALI, facciamo partire la cattura mentre da WINDOWS ci colleghiamo al sito fake. I risultati sono i seguenti:



Da questa schermata   possibile vedere lo scambio di risposte tra client e server che comunicano attraverso il protocollo di trasporto TCP instaurato tramite il “three-way handshake” (SYN/ACK).   inoltre possibile osservare il metodo utilizzato GET e la risposta 200 OK del server.

Evidenziando le diverse righe possiamo visualizzare gli indirizzi MAC di origine e destinazione e come questi cambiano nei vari livelli di comunicazione di rete.

Apply a display filter ... <Ctrl-/>						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	PcsCompu_72:ff:6d	Broadcast	ARP	60	Who has 192.168
2	0.000016213	PcsCompu_c7:e1:36	PcsCompu_72:ff:6d	ARP	42	192.168.32.100
3	0.000372955	192.168.32.101	192.168.32.100	TCP	66	49268 → 80 [SYN
4	0.000396184	192.168.32.100	192.168.32.101	TCP	66	80 → 49268 [SYN
5	0.000813191	192.168.32.101	192.168.32.100	TCP	60	49268 → 80 [ACK
6	0.000990370	192.168.32.101	192.168.32.100	HTTP	352	GET / HTTP/1.1
7	0.000998344	192.168.32.100	192.168.32.101	TCP	54	80 → 49268 [ACK
8	0.115446290	192.168.32.100	192.168.32.101	TCP	204	80 → 49268 [PSH
9	0.119569095	192.168.32.100	192.168.32.101	HTTP	312	HTTP/1.1 200 OK
10	0.119926928	192.168.32.101	192.168.32.100	TCP	60	49268 → 80 [ACK
11	0.120196512	192.168.32.101	192.168.32.100	TCP	60	49268 → 80 [FIN
12	0.120211091	192.168.32.100	192.168.32.101	TCP	54	80 → 49268 [ACK
13	5.244378482	PcsCompu_c7:e1:36	PcsCompu_72:ff:6d	ARP	42	Who has 192.168
14	5.244828025	PcsCompu_72:ff:6d	PcsCompu_c7:e1:36	ARP	60	192.168.32.101

▶ Frame 3: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface eth0, id 0
 ▾ Ethernet II, Src: PcsCompu_72:ff:6d (08:00:27:72:ff:6d), Dst: PcsCompu_c7:e1:36 (08:00:27:c7:e1:36)
 ▶ Destination: PcsCompu_c7:e1:36 (08:00:27:c7:e1:36)
 ▶ Source: PcsCompu_72:ff:6d (08:00:27:72:ff:6d)
 Type: IPv4 (0x0800)
 ▶ Internet Protocol Version 4, Src: 192.168.32.101, Dst: 192.168.32.100
 ▶ Transmission Control Protocol, Src Port: 49268, Dst Port: 80, Seq: 0, Len: 0

Apply a display filter ... <Ctrl-/>						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	PcsCompu_72:ff:6d	Broadcast	ARP	60	Who has 192.168
2	0.000016213	PcsCompu_c7:e1:36	PcsCompu_72:ff:6d	ARP	42	192.168.32.100
3	0.000372955	192.168.32.101	192.168.32.100	TCP	66	49268 → 80 [SYN
4	0.000396184	192.168.32.100	192.168.32.101	TCP	66	80 → 49268 [SYN
5	0.000813191	192.168.32.101	192.168.32.100	TCP	60	49268 → 80 [ACK
6	0.000990370	192.168.32.101	192.168.32.100	HTTP	352	GET / HTTP/1.1
7	0.000998344	192.168.32.100	192.168.32.101	TCP	54	80 → 49268 [ACK
8	0.115446290	192.168.32.100	192.168.32.101	TCP	204	80 → 49268 [PSH
9	0.119569095	192.168.32.100	192.168.32.101	HTTP	312	HTTP/1.1 200 OK
10	0.119926928	192.168.32.101	192.168.32.100	TCP	60	49268 → 80 [ACK
11	0.120196512	192.168.32.101	192.168.32.100	TCP	60	49268 → 80 [FIN
12	0.120211091	192.168.32.100	192.168.32.101	TCP	54	80 → 49268 [ACK
13	5.244378482	PcsCompu_c7:e1:36	PcsCompu_72:ff:6d	ARP	42	Who has 192.168
14	5.244828025	PcsCompu_72:ff:6d	PcsCompu_c7:e1:36	ARP	60	192.168.32.101

▶ Frame 4: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface eth0, id 0
 ▾ Ethernet II, Src: PcsCompu_c7:e1:36 (08:00:27:c7:e1:36), Dst: PcsCompu_72:ff:6d (08:00:27:72:ff:6d)
 ▶ Destination: PcsCompu_72:ff:6d (08:00:27:72:ff:6d)
 ▶ Source: PcsCompu_c7:e1:36 (08:00:27:c7:e1:36)
 Type: IPv4 (0x0800)
 ▶ Internet Protocol Version 4, Src: 192.168.32.100, Dst: 192.168.32.101
 ▶ Transmission Control Protocol, Src Port: 80, Dst Port: 49268, Seq: 0, Ack: 1, Len: 0

Ripetendo la procedura di cattura, ma cercando questa volta da WINDOWS di raggiungere il sito HTTPS, wireshark evidenzierà come ci sia un fallimento nell'instaurazione di una connessione sicura TCP, a causa della mancanza dei certificati di sicurezza.

