

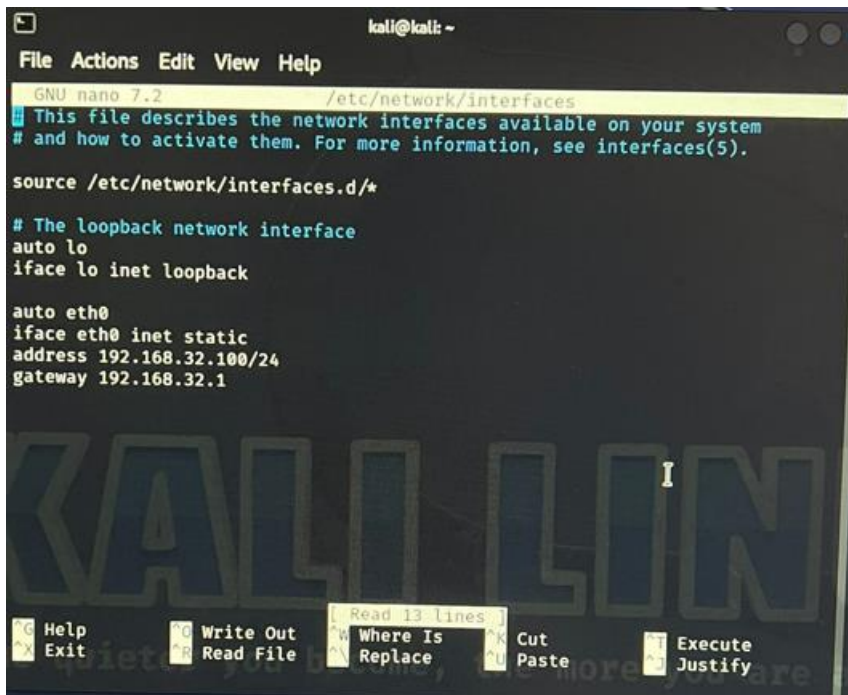
# Configurazione DNS HTTPS ed wireshark

## 1)Cambiare indirizzo ip su kali

Entro nel pannello di comandi digito sudo nano /etc/network/interfaces.d/\*

Digito la password

Ed configuro.



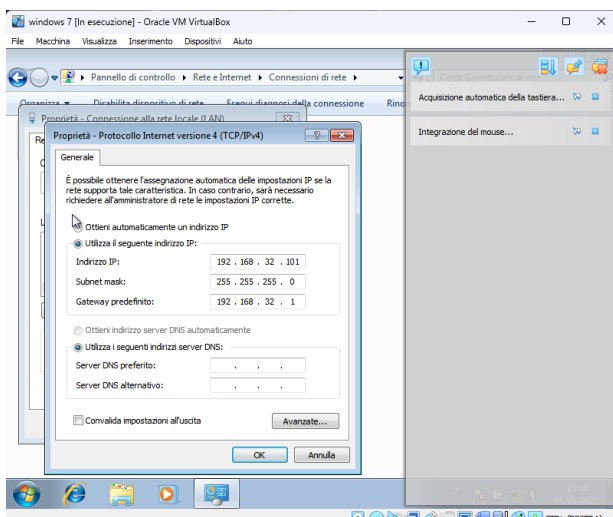
```
kali@kali: ~  
File Actions Edit View Help  
GNU nano 7.2 /etc/network/interfaces  
# This file describes the network interfaces available on your system  
# and how to activate them. For more information, see interfaces(5).  
  
source /etc/network/interfaces.d/*  
  
# The loopback network interface  
auto lo  
iface lo inet loopback  
  
auto eth0  
iface eth0 inet static  
address 192.168.32.100/24  
gateway 192.168.32.1  
  
Help Exit Write Out Read File Read 13 lines Where Is Replace Cut Paste Execute Justify
```

Premo ctrl o per salvare ed invio .

Per uscire ctrl x

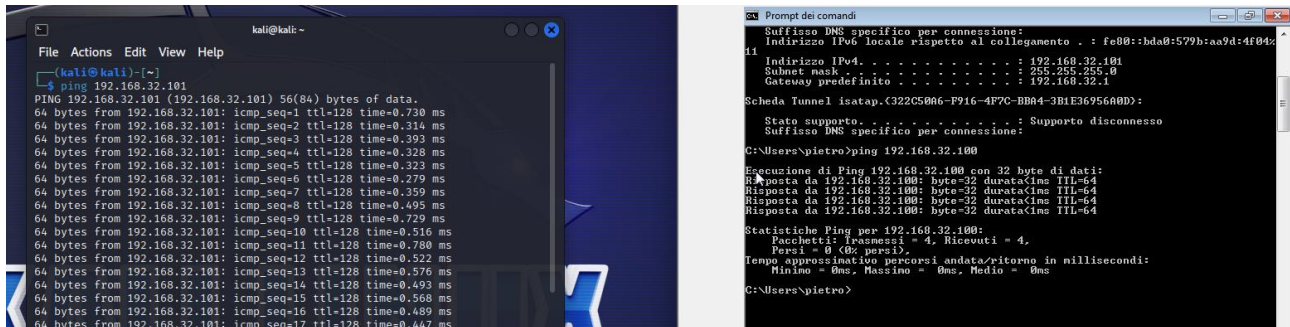
Cambio indirizzo ip di windows

Vado su pannello controllo dopo su rete e internet poi centro controllo connessioni , dopo di che su connessioni alla rete locale. proprietà; selezioni protocollo internet versione 4(tcp/ipv4) ed configuro.



Riavvio kali con sudo reboot

Ed controllo i Ping



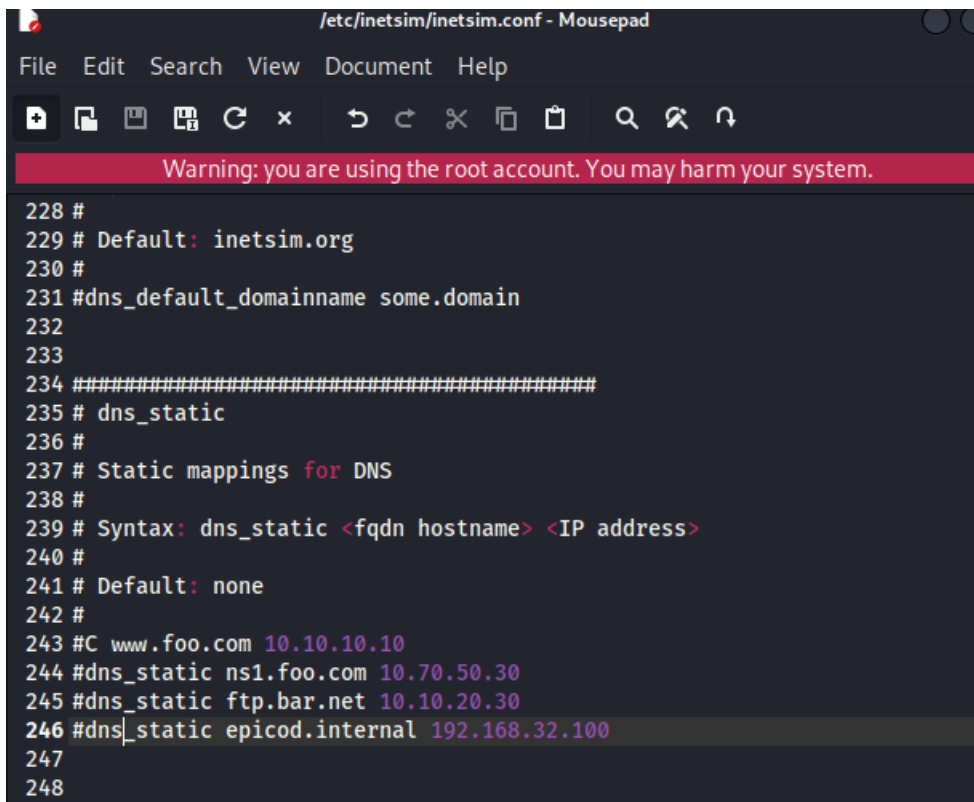
```
kali@kali: ~  
File Actions Edit View Help  
~  
(kali@kali)~  
$ ping 192.168.32.101  
PING 192.168.32.101 (192.168.32.101) 56(84) bytes of data:  
64 bytes from 192.168.32.101: icmp_seq=1 ttl=128 time=0.730 ms  
64 bytes from 192.168.32.101: icmp_seq=2 ttl=128 time=0.314 ms  
64 bytes from 192.168.32.101: icmp_seq=3 ttl=128 time=0.393 ms  
64 bytes from 192.168.32.101: icmp_seq=4 ttl=128 time=0.328 ms  
64 bytes from 192.168.32.101: icmp_seq=5 ttl=128 time=0.323 ms  
64 bytes from 192.168.32.101: icmp_seq=6 ttl=128 time=0.279 ms  
64 bytes from 192.168.32.101: icmp_seq=7 ttl=128 time=0.359 ms  
64 bytes from 192.168.32.101: icmp_seq=8 ttl=128 time=0.495 ms  
64 bytes from 192.168.32.101: icmp_seq=9 ttl=128 time=0.729 ms  
64 bytes from 192.168.32.101: icmp_seq=10 ttl=128 time=0.516 ms  
64 bytes from 192.168.32.101: icmp_seq=11 ttl=128 time=0.780 ms  
64 bytes from 192.168.32.101: icmp_seq=12 ttl=128 time=0.522 ms  
64 bytes from 192.168.32.101: icmp_seq=13 ttl=128 time=0.576 ms  
64 bytes from 192.168.32.101: icmp_seq=14 ttl=128 time=0.493 ms  
64 bytes from 192.168.32.101: icmp_seq=15 ttl=128 time=0.568 ms  
64 bytes from 192.168.32.101: icmp_seq=16 ttl=128 time=0.489 ms  
64 bytes from 192.168.32.101: icmp_seq=17 ttl=128 time=0.447 ms
```

```
Prompt dei comandi  
Suffisso DNS specifico per connessione:  
Indirizzo IPv6 locale rispetto al collegamento . : fe80::bda0:579b:aa9d:4f04%  
11  
Indirizzo IPv4 . . . . . : 192.168.32.101  
Subnet mask . . . . . : 255.255.255.0  
Gateway predefinito . . . . . : 192.168.32.1  
Scheda Tunnel isatap.{322C50A6-F916-4F7C-BBA4-3B1E36956A0D}:  
Stato supporto. . . . . : Supporto disconnesso  
Suffisso DNS specifico per connessione:  
C:\Users\pietro>ping 192.168.32.100  
Esecuzione di Ping 192.168.32.100 con 32 byte di dati:  
Risposta da 192.168.32.100: byte=32 durata<ms TTL=64  
Risposta da 192.168.32.100: byte=32 durata<ms TTL=64  
Risposta da 192.168.32.100: byte=32 durata<ms TTL=64  
Statistiche Ping per 192.168.32.100:  
Pacchetti: Inviati = 4, Ricevuti = 4,  
Persi = 0 (0% persi),  
Tempo approssimativo percorsi andata/ritorno in millisecondi:  
Minimo = 0ms, Massimo = 0ms, Medio = 0ms  
C:\Users\pietro>
```

## 2)Setto la DNS

Inserendo a kali in nel pannello di comando sudo nano /etc/inetsim/inetsim.conf

Inserisco

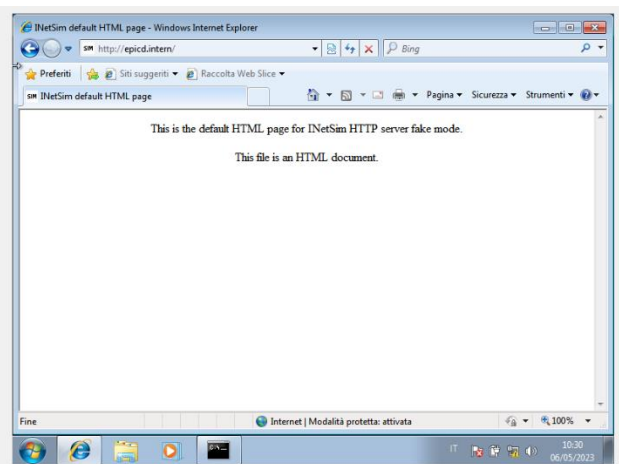


```
/etc/inetsim/inetsim.conf - Mousepad  
File Edit Search View Document Help  
Warning: you are using the root account. You may harm your system.  
228 #  
229 # Default: inetsim.org  
230 #  
231 #dns_default_domainname some.domain  
232  
233  
234 #####  
235 # dns_static  
236 #  
237 # Static mappings for DNS  
238 #  
239 # Syntax: dns_static <fqdn hostname> <IP address>  
240 #  
241 # Default: none  
242 #  
243 #C www.foo.com 10.10.10.10  
244 #dns_static ns1.foo.com 10.70.50.30  
245 #dns_static ftp.bar.net 10.10.20.30  
246 #dns_static epicod.internal 192.168.32.100  
247  
248
```

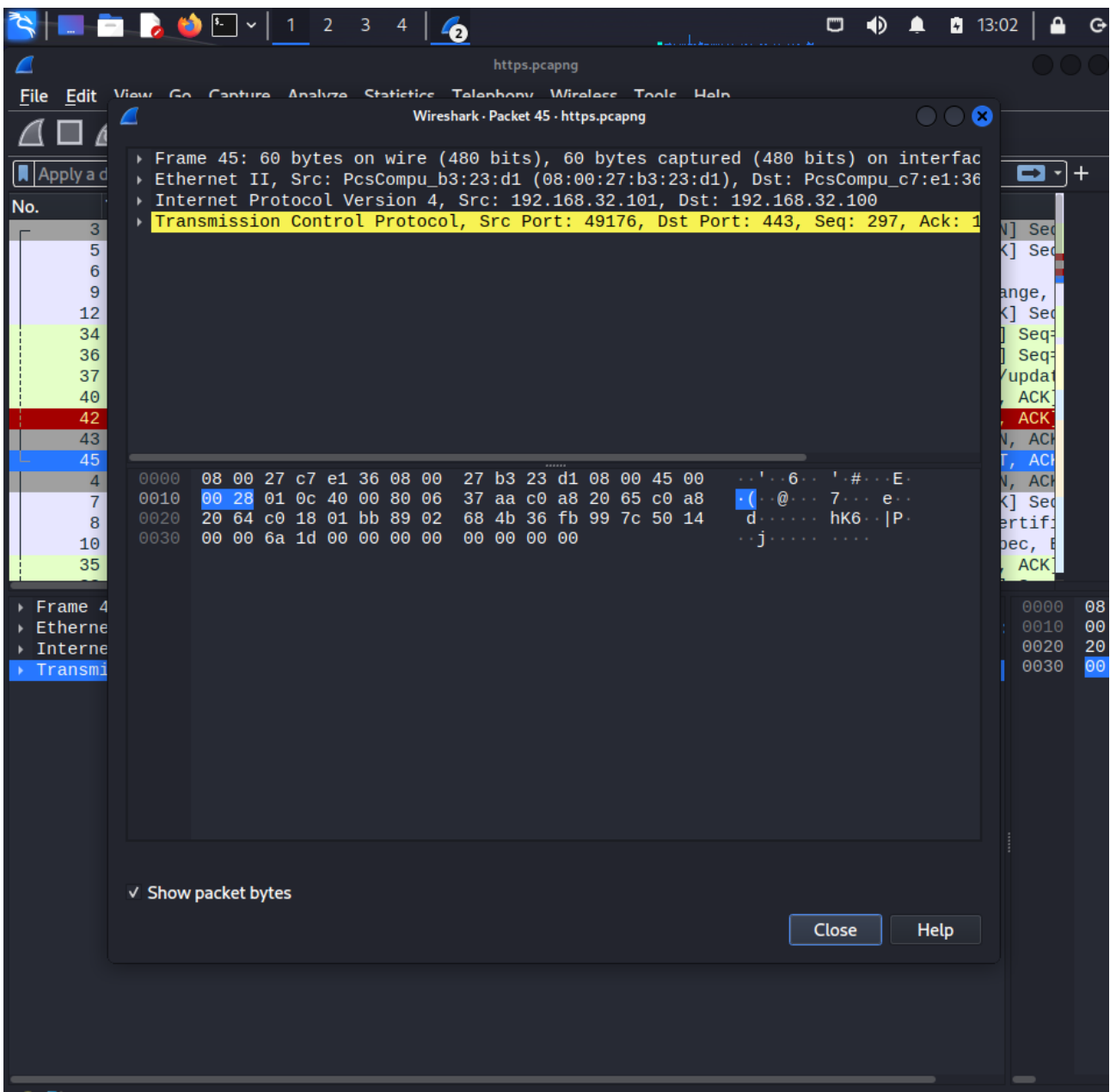
Avvio la simulazione con sudo inetsim

```
kali@kali:~$ sudo inetsim
Sorry, this program must be started as root!

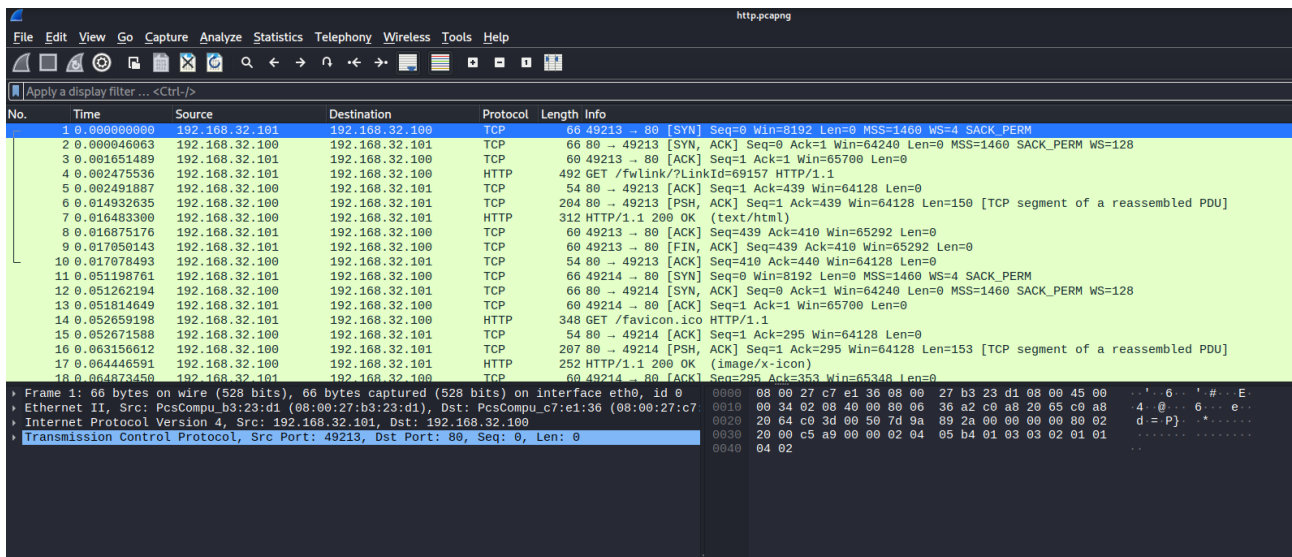
(kali@kali:~$)
# sudo inetsim
INetsim 1.2.2 (2020-05-10) by Matthias Eckert & Thomas Hungenberg
Main logfile '/var/log/inetsim/main.log' does not exist. Trying to create it.
...
Main logfile '/var/log/inetsim/main.log' successfully created.
Sub logfile '/var/log/inetsim/service.log' does not exist. Trying to create it.
...
Sub logfile '/var/log/inetsim/service.log' successfully created.
Debug logfile '/var/log/inetsim/debug.log' does not exist. Trying to create it.
...
Debug logfile '/var/log/inetsim/debug.log' successfully created.
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 'www.foo.com' in configuration file '/etc/inetsim/inetsim.conf' line 243
Configuration file parsed successfully.
== INetsim main process started (PID 51548) ==
Session ID: 51548
Listening on: 192.168.32.100
Real Date/Time: 2023-05-06 04:27:43
Fake Date/Time: 2023-05-06 04:27:43 (Delta: 0 seconds)
```



### 3)Procedo con Wireshark



Porta 433 per cui https noto che dei pacchetti sono criptati



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.32.101	192.168.32.100	TCP	60	49213 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM
2	0.000046063	192.168.32.100	192.168.32.101	TCP	60	80 → 49213 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
3	0.001651489	192.168.32.101	192.168.32.100	TCP	60	49213 → 80 [ACK] Seq=1 Ack=1 Win=65700 Len=0
4	0.002475536	192.168.32.101	192.168.32.100	HTTP	492	GET /fwLink/?LinkId=69157 HTTP/1.1
5	0.002491887	192.168.32.100	192.168.32.101	TCP	54	80 → 49213 [ACK] Seq=1 Ack=439 Win=64128 Len=0
6	0.014932635	192.168.32.100	192.168.32.101	TCP	204	80 → 49213 [PSH, ACK] Seq=1 Ack=439 Win=64128 Len=150 [TCP segment of a reassembled PDU]
7	0.016483300	192.168.32.100	192.168.32.101	HTTP	312	HTTP/1.1 200 OK (text/html)
8	0.016875176	192.168.32.101	192.168.32.100	TCP	60	49213 → 80 [ACK] Seq=439 Ack=410 Win=65292 Len=0
9	0.017050143	192.168.32.101	192.168.32.100	TCP	60	49213 → 80 [FIN, ACK] Seq=439 Ack=410 Win=65292 Len=0
10	0.017078493	192.168.32.100	192.168.32.101	TCP	54	80 → 49213 [ACK] Seq=410 Ack=440 Win=64128 Len=0
11	0.051198761	192.168.32.101	192.168.32.100	TCP	60	49214 → 80 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM
12	0.051262194	192.168.32.100	192.168.32.101	TCP	60	80 → 49214 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SACK_PERM WS=128
13	0.051814649	192.168.32.101	192.168.32.100	TCP	60	49214 → 80 [ACK] Seq=1 Ack=1 Win=65700 Len=0
14	0.052659198	192.168.32.101	192.168.32.100	HTTP	348	GET /favicon.ico HTTP/1.1
15	0.052671588	192.168.32.100	192.168.32.101	TCP	54	80 → 49214 [ACK] Seq=1 Ack=295 Win=64128 Len=0
16	0.063156612	192.168.32.100	192.168.32.101	TCP	207	80 → 49214 [PSH, ACK] Seq=1 Ack=295 Win=64128 Len=153 [TCP segment of a reassembled PDU]
17	0.064446591	192.168.32.100	192.168.32.101	HTTP	252	HTTP/1.1 200 OK (image/x-icon)
18	0.064873450	192.168.32.101	192.168.32.100	TCP	60	49214 → 80 [ACK] Seq=295 Ack=353 Win=65348 Len=0

Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface eth0, id 0  
Ethernet II, Src: PcsCompu\_b3:23:d1 (08:00:27:b3:23:d1), Dst: PcsCompu\_c7:e1:36 (08:00:27:c7:e1:36)  
Internet Protocol Version 4, Src: 192.168.32.101, Dst: 192.168.32.100  
Transmission Control Protocol, Src Port: 49213, Dst Port: 80, Seq: 0, Len: 0

Porta 80 e tutti i pacchetti sono leggibili .

**Http e https** sono due varianti dello stesso protocollo di telecomunicazione, ovvero quell'insieme di regole che definisce i metodi comunicativi tra due o più entità. Essi sono molto utilizzati durante la navigazione web medianti i più conosciuti browser, come Google Chrome, Microsoft Edge e Safari.

La principale differenza tra **http** (l'acronimo di **HyperText Transfer Protocol**) ed **https** (la cui **s** finale sta per **Secure**) risiede appunto nella maggiore sicurezza che quest'ultima variante di protocollo offre rispetto alla prima. Infatti, se nell'http lo scambio di risorse tra client e server avviene "in chiaro" (ovvero l'informazione che viene diffusa può essere letta da chi decide di intromettersi senza diritto nello scambio dei dati), nell'https la comunicazione è protetta grazie all'impiego di determinati certificati (come quello **SSL**, acronimo di **Secure Socket Layer**) .

## HTTPS = HTTP + SSL

Per proteggere le informazioni potenzialmente sensibili dalla perdita, i siti web utilizzano i certificati SSL per creare una connessione sicura tra server web e browser, proteggendo la trasmissione di richieste e risposte HTTP.

L'uso di un certificato SSL è la differenza fondamentale tra HTTP e HTTPS.

HTTPS crittografa il trasporto dei dati in modo che non sia visibile agli hacker o ad altri che monitorano la connessione. Ciò garantisce l'integrità dei dati e impedisce che le informazioni vengano modificate, danneggiate o rubate durante la trasmissione.

I protocolli SSL/TLS autenticano inoltre gli utenti per proteggere le informazioni e garantire che non vengano rivelate a utenti non autorizzati.

