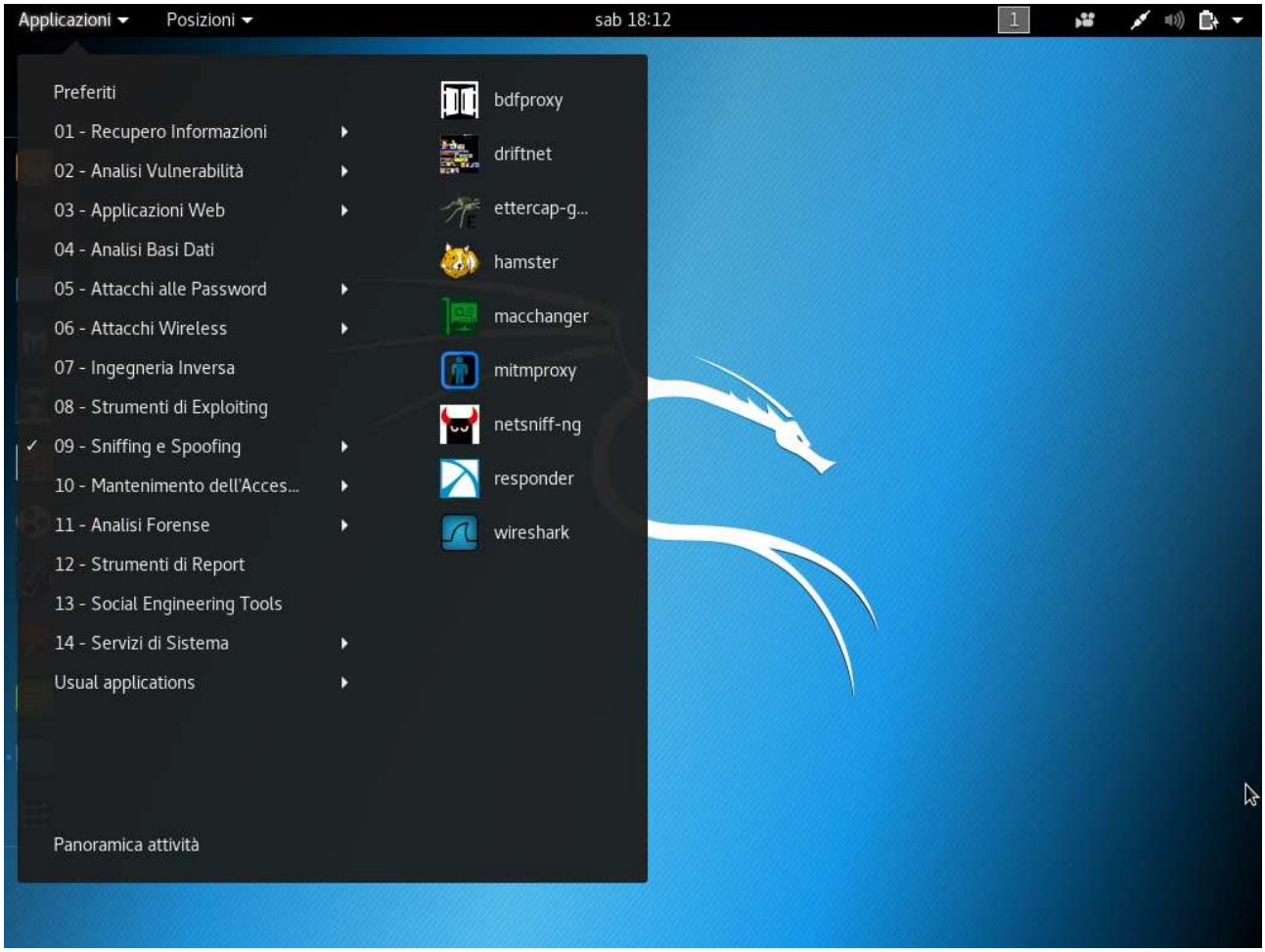
SNIFFING DI DATI DALLA RETE CON WIRESHARK

Sniffing è il termine colloquiale che si usa per indicare la cattura di dati dalla rete.

Letteralmente significa fiutare e, come un cane che fiuta per individuare una pista, noi fiutiamo la rete per individuare pacchetti.

Nella distribuzione Kali Linux, trovate Wireshark nel menu 09-Sniffing e Spoofing.

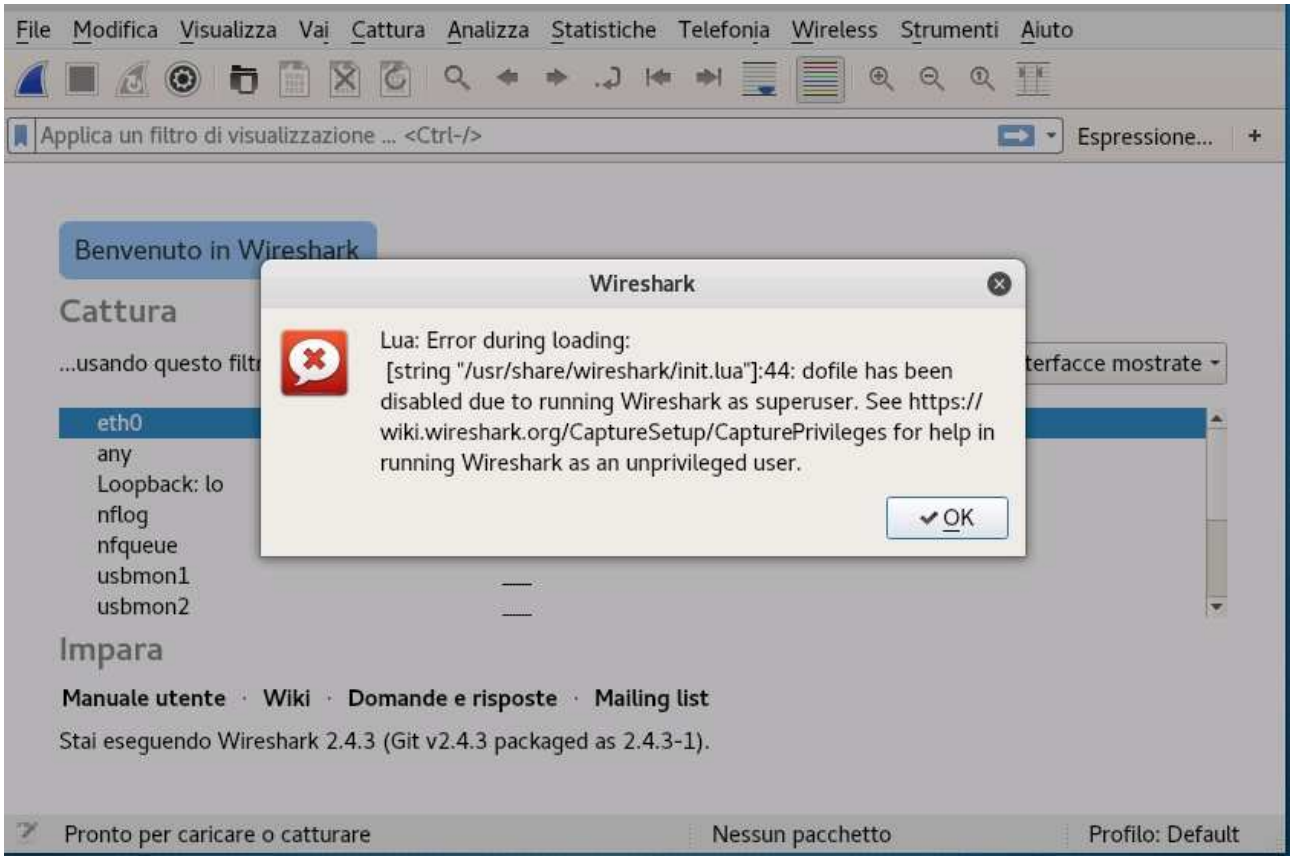


Una volta lanciato il programma, esce un avvertimento che vi ricorda che non è una buona idea catturare pacchetti con un account root per motivi di sicurezza.

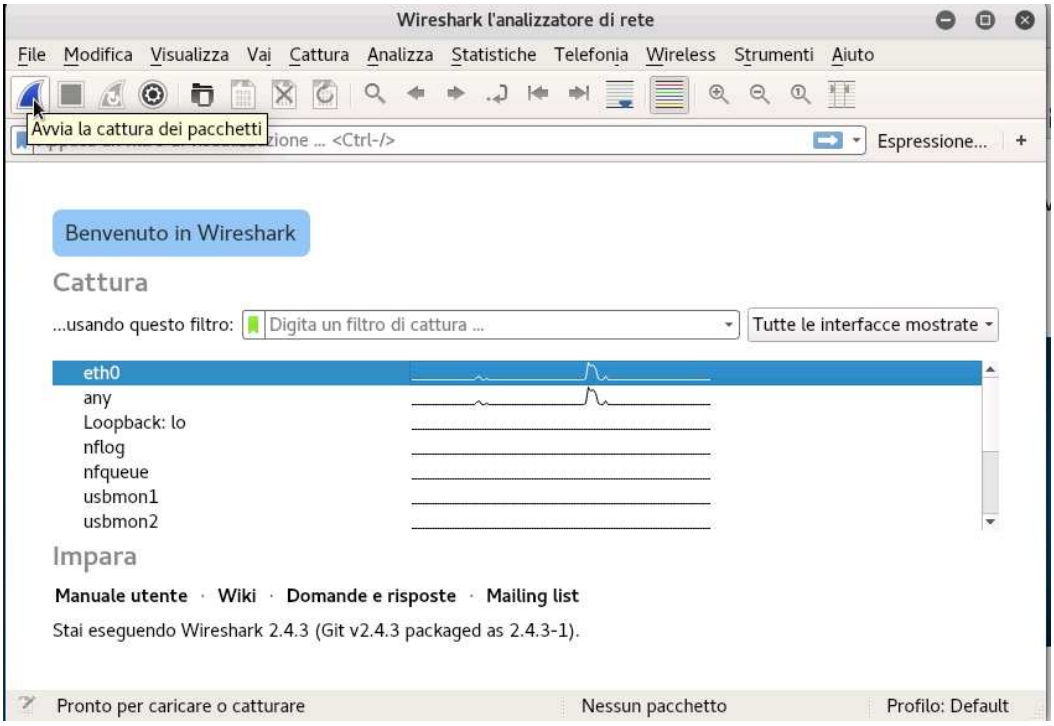
Wireshark esegue l'analisi sintattica di una grande quantità di dati non fidati ed è esposto a vulnerabilità di corruzione della memoria, che potrebbero portare all'esecuzione di codice maligno.

Se eseguiamo lo sniffing con utente con meno privilegi riduce i rischi di sicurezza.

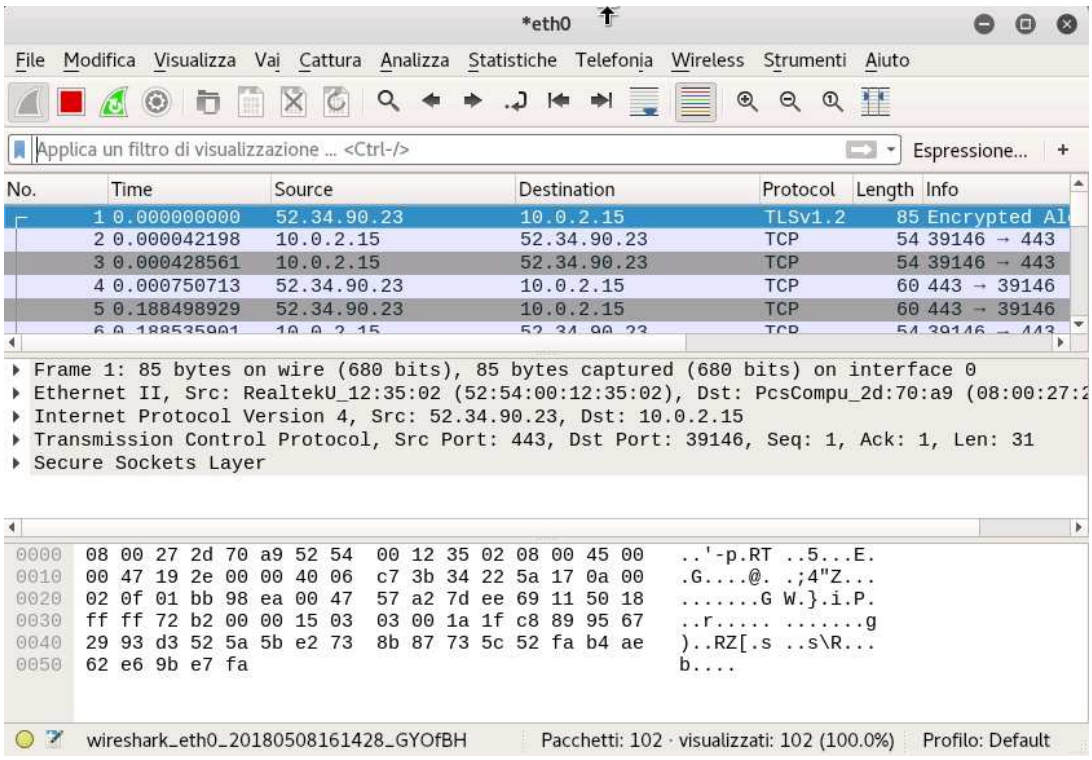
In questa lezione ignoriamo la cosa e continuiamo



Per catturare i pacchetti è sufficiente scegliere un'interfaccia (in questo caso l'eth0) e cliccare sull'icona a forma di pinna di squalo.



Come si vede nella seguente figura i pacchetti vengono catturati in pochi secondi dall'inizio dello sniffing.



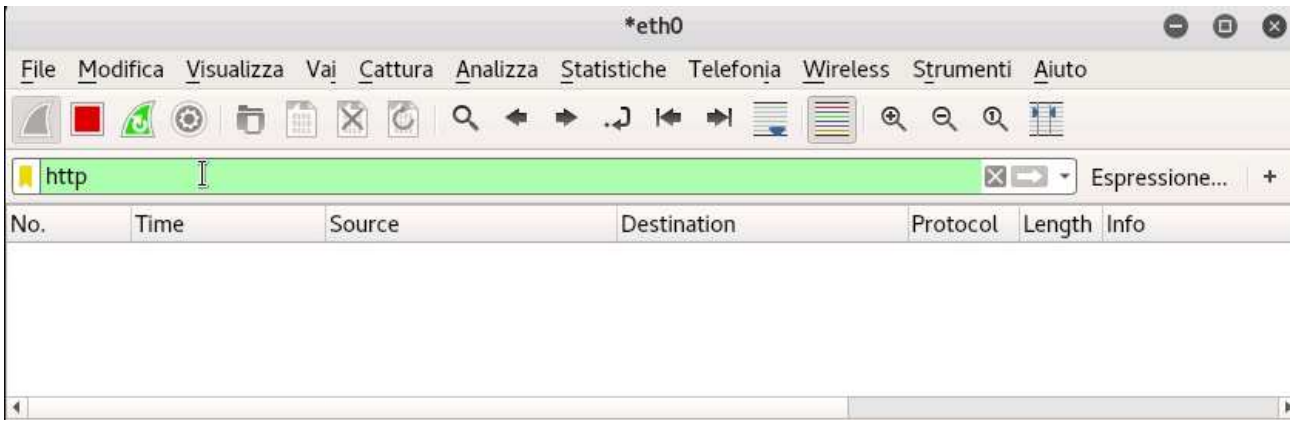
Cliccare sul bottone sulla destra della pinna, a forma di quadrato rosso, per terminare la cattura.

Cliccando su un pacchetto, nel pannello centrale si vedrà la scomposizione del pacchetto.

Qui potete espandere qualsiasi sottoalbero facendo clic sulla freccia relativa immediatamente alla sua sinistra.

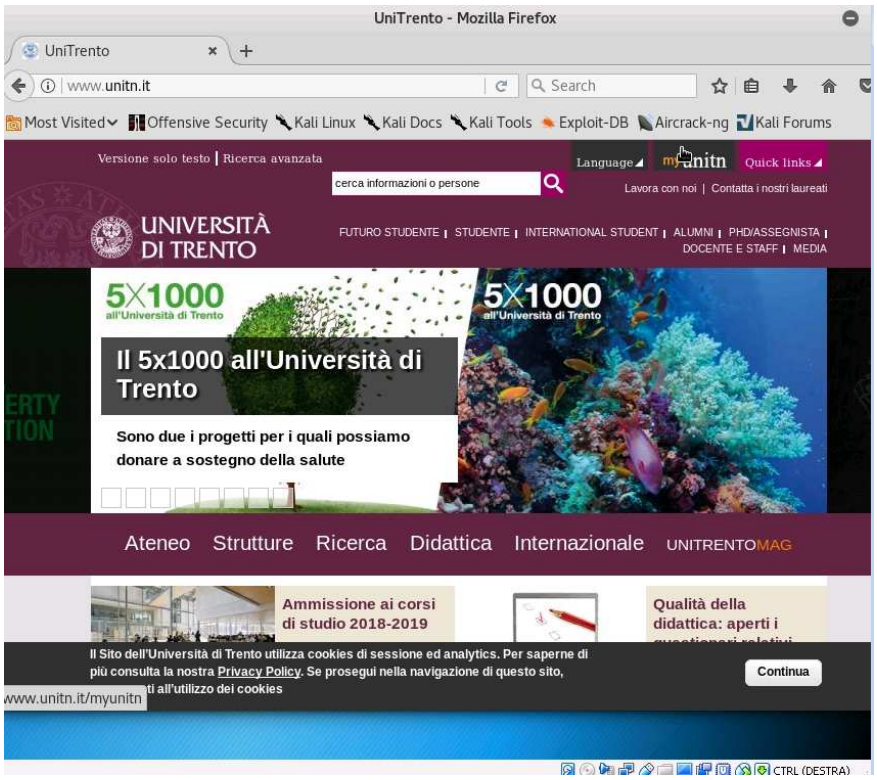
FILTRAGGIO DEL PROTOCOLLO HTTP

Scriviamo ora nelle finestra filtro questa voce: http

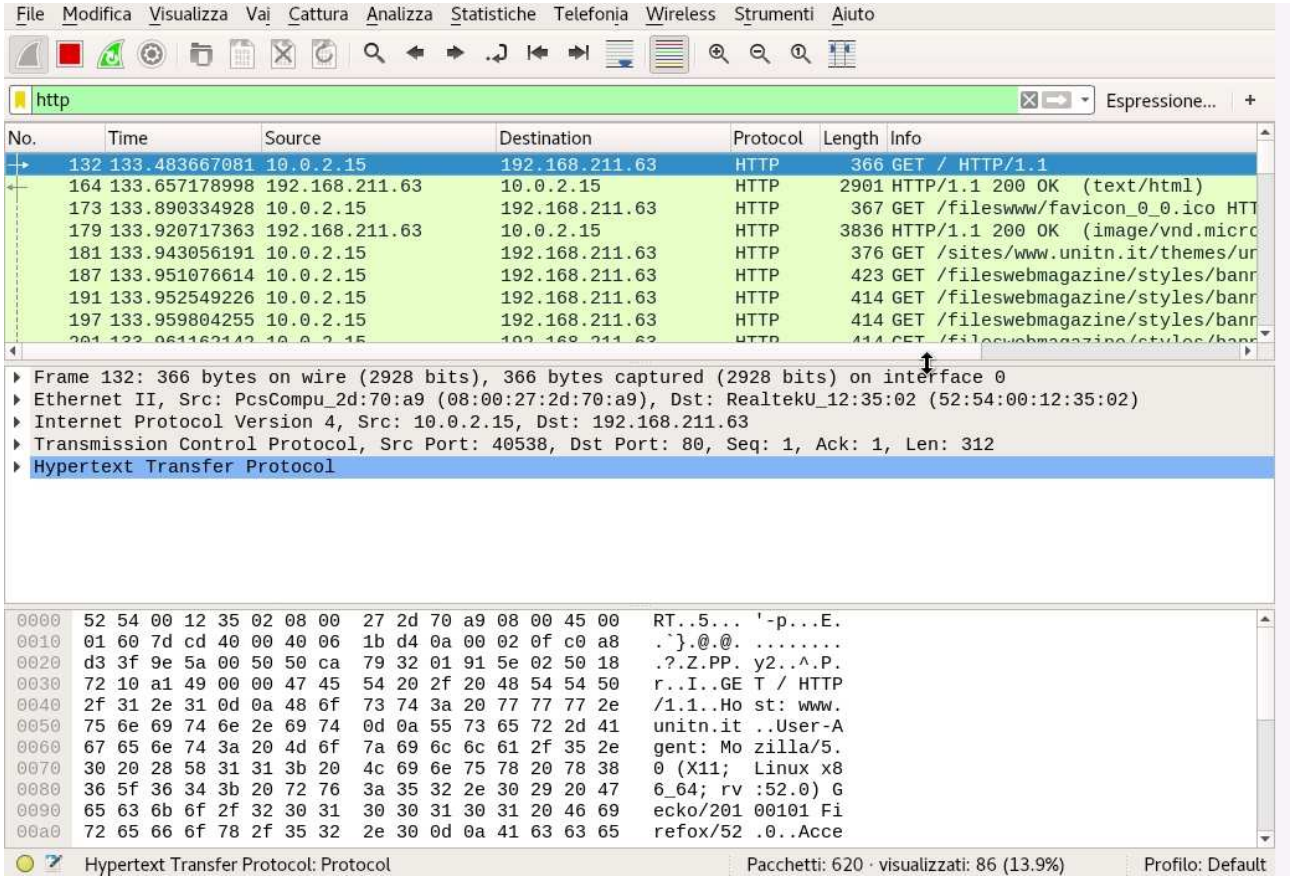


In questo modo, verranno visualizzati solo i pacchetti relativi al traffico web.

Facciamo partire la cattura del pacchetti. E nel frattempo apriamo un browser ed iniziamo a navigare. (Digitiamo, per esempi www.unitn.it sul browser)

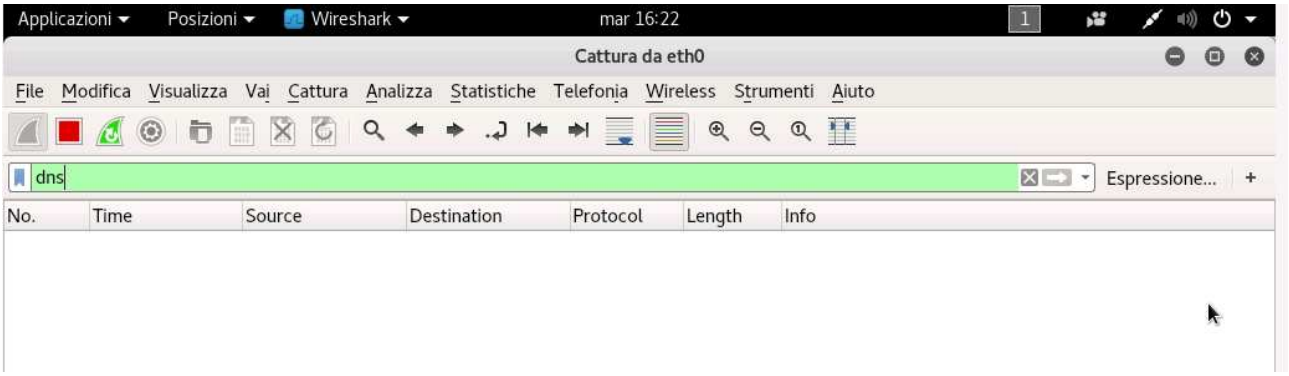


Nella finestra dei pacchetti potete ora vedere la sequenza di pacchetti inviati secondo il protocollo http.

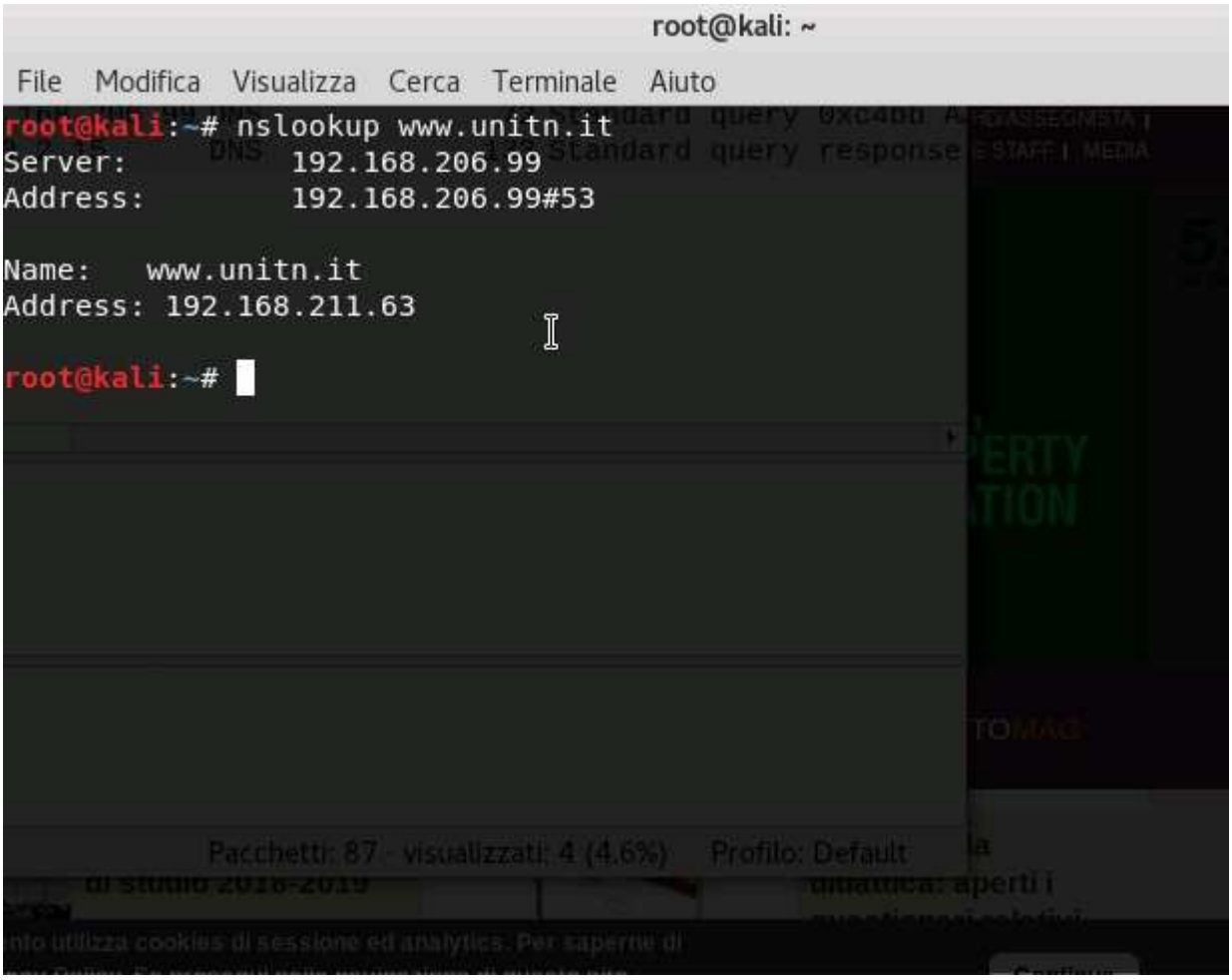


FILTRAGGIO DEL PROTOCOLLO DNS

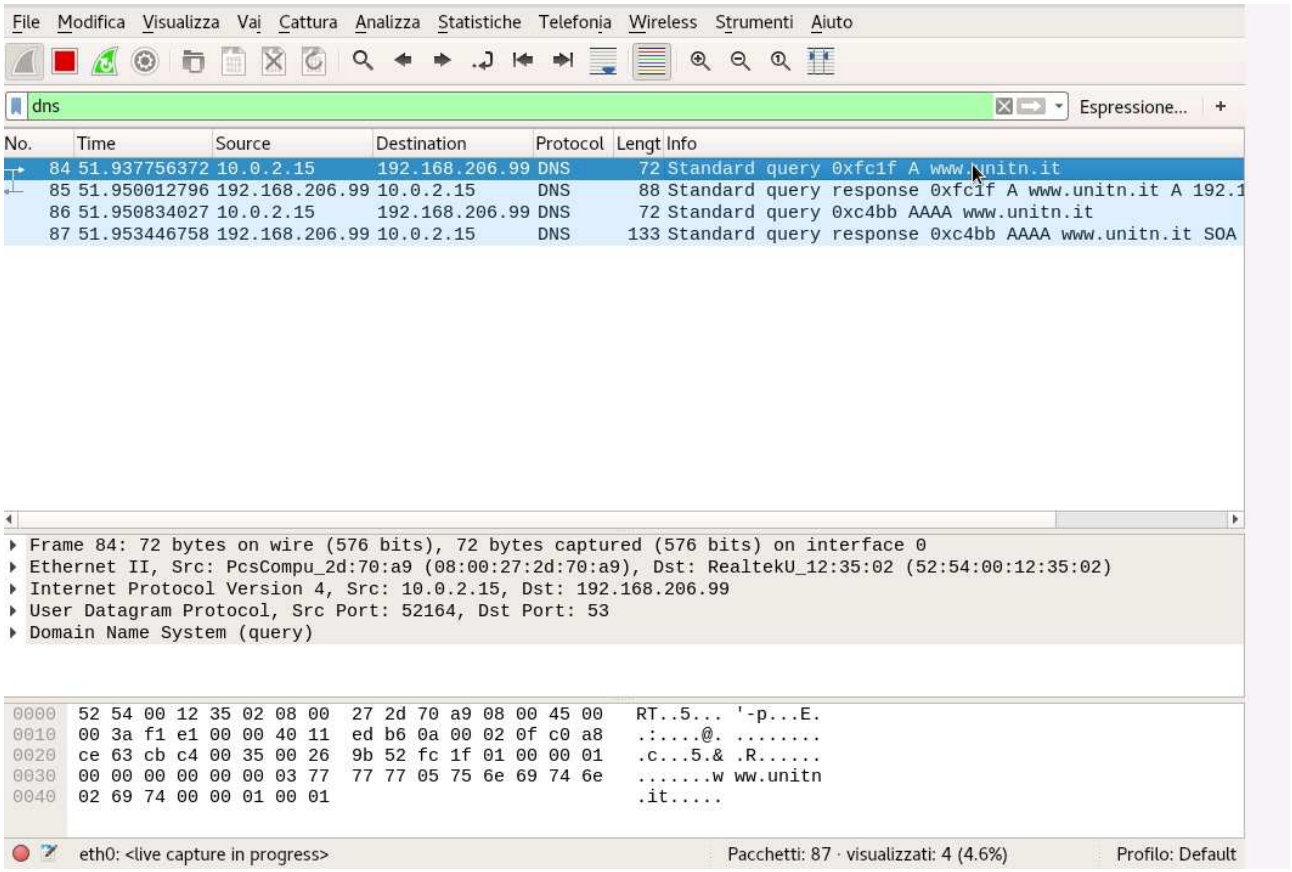
Scriviamo ora nelle finestra filtro questa voce: dns



Apriamo un terminale e scriviamo: **`nslookup`** **`www.unitn.it`**



Nella finestra dei pacchetti potete ora vedere la sequenza di pacchetti inviati secondo il protocollo dns.



FILTRAGGIO DEL PROTOCOLLO ICMP

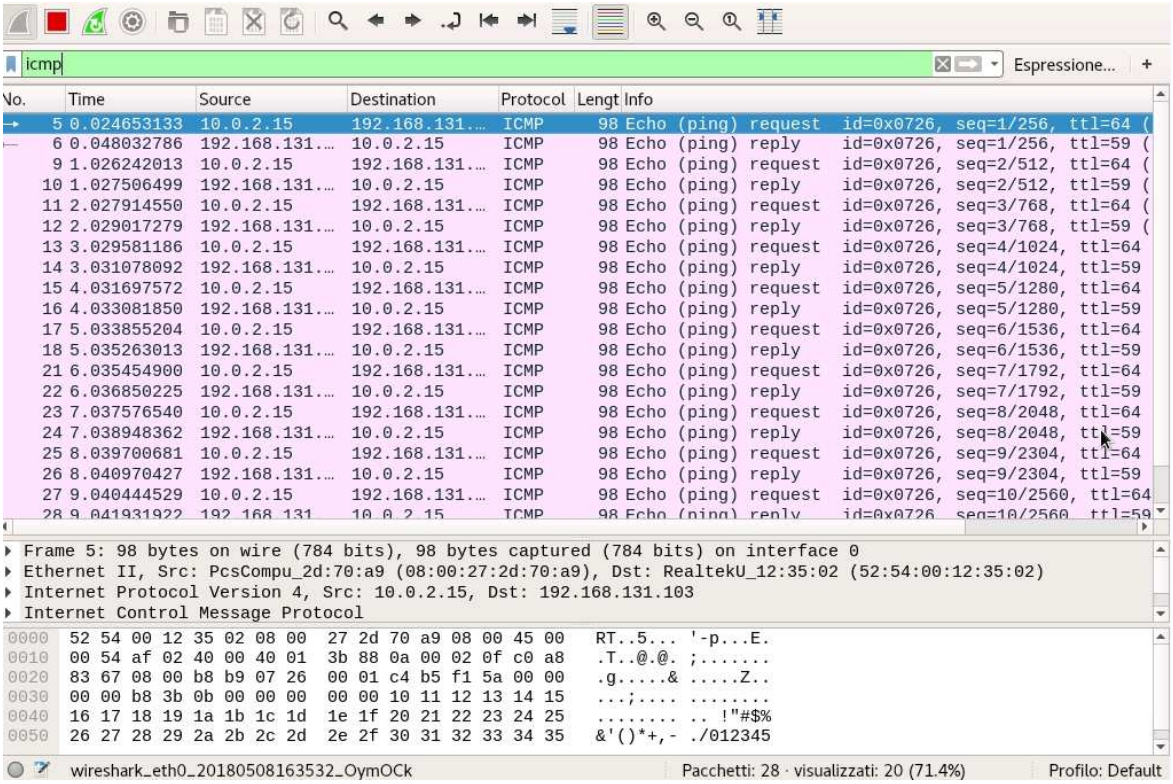
Scriviamo ora nelle finestra filtro questa voce: icmp e facciamo partire la cattura.

Apriamo un terminale e inviamo 10 pacchetti ping al nostro server retiavanzatex.disi.unitn.it scriviamo:

```
ping -c10 retiavanzatex.disi.unitn.it
```

```
root@kali:~# ping -c10 retiavanzatex.disi.unitn.it
PING retiavanzatex.disi.unitn.it (192.168.131.103) 56(84) bytes of data.
64 bytes from retiavanzatex.disi.unitn.it (192.168.131.103): icmp_seq=1 ttl=59 t
ime=23.3 ms
64 bytes from retiavanzatex.disi.unitn.it (192.168.131.103): icmp_seq=2 ttl=59 t
ime=1.28 ms
64 bytes from retiavanzatex.disi.unitn.it (192.168.131.103): icmp_seq=3 ttl=59 t
ime=1.12 ms
64 bytes from retiavanzatex.disi.unitn.it (192.168.131.103): icmp_seq=4 ttl=59 t
ime=1.53 ms
64 bytes from retiavanzatex.disi.unitn.it (192.168.131.103): icmp_seq=5 ttl=59 t
ime=1.41 ms
64 bytes from retiavanzatex.disi.unitn.it (192.168.131.103): icmp_seq=6 ttl=59 t
ime=1.44 ms
64 bytes from retiavanzatex.disi.unitn.it (192.168.131.103): icmp_seq=7 ttl=59 t
ime=1.42 ms
64 bytes from retiavanzatex.disi.unitn.it (192.168.131.103): icmp_seq=8 ttl=59 t
ime=1.40 ms
64 bytes from retiavanzatex.disi.unitn.it (192.168.131.103): icmp_seq=9 ttl=59 t
ime=1.30 ms
64 bytes from retiavanzatex.disi.unitn.it (192.168.131.103): icmp_seq=10 ttl=59
time=1.51 ms

--- retiavanzatex.disi.unitn.it ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9015ms
rtt min/avg/max/mdev = 1.128/3.584/23.397/6.605 ms
root@kali:~#
```

Nella finestra dei pacchetti potete ora vedere la sequenza di pacchetti inviati secondo il protocollo ping, le echo request e le echo reply.

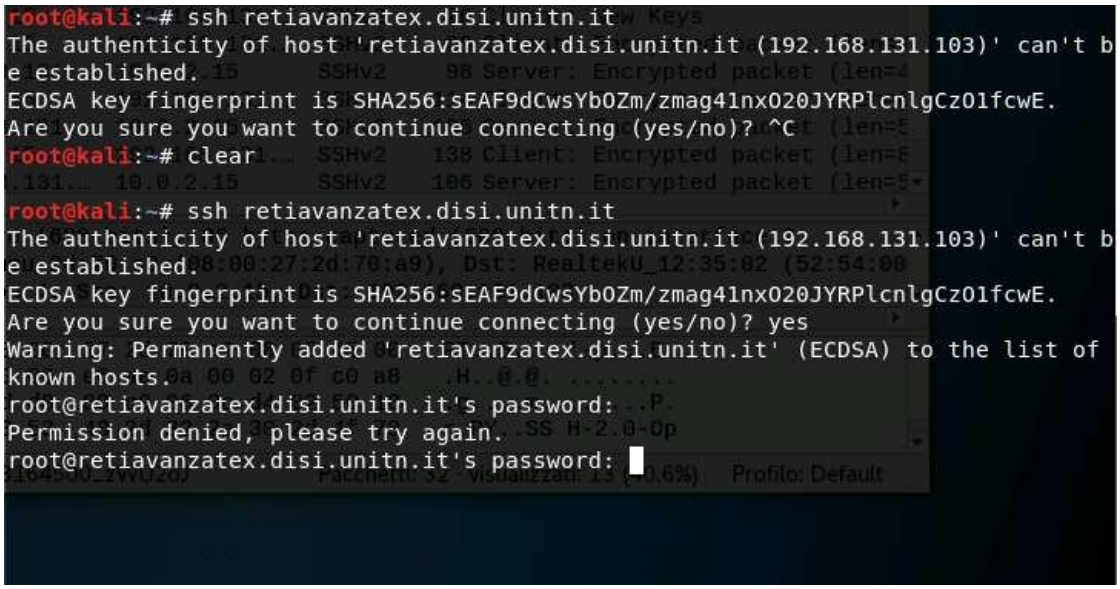


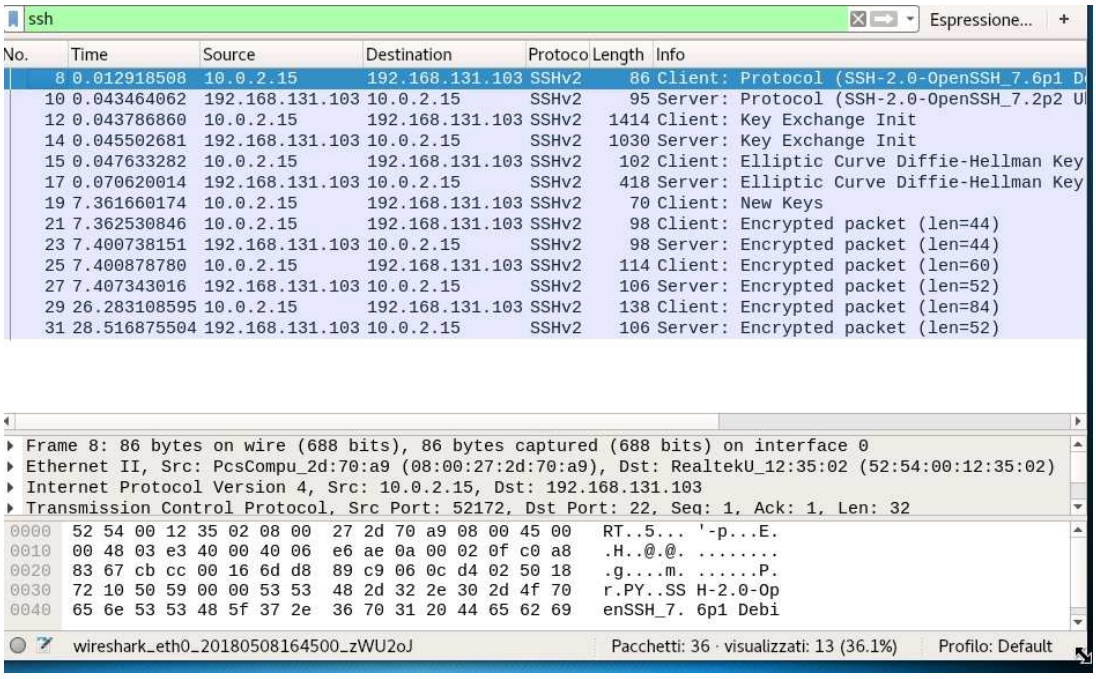
FILTRAGGIO DEL PROTOCOLLO SSH

Scriviamo ora nelle finestra filtro questa voce: **`ssh`** e facciamo partire la cattura.

Apriamo un terminale e scriviamo:

`ssh retiavanzatex.disi.unitn.it`





MODALITA' PROMISCUA

Normalmente potete vedere solo il traffico di rete che ha origine dalla vostra macchina, che è destinato alla vostra macchina, o il traffico broadcast in quanto una macchina si interessa solo dei pacchetti che gli sono pertinenti.

Quando la scheda di rete riceve un pacchetto che non ha il suo indirizzo, il pacchetto viene lasciato andare.

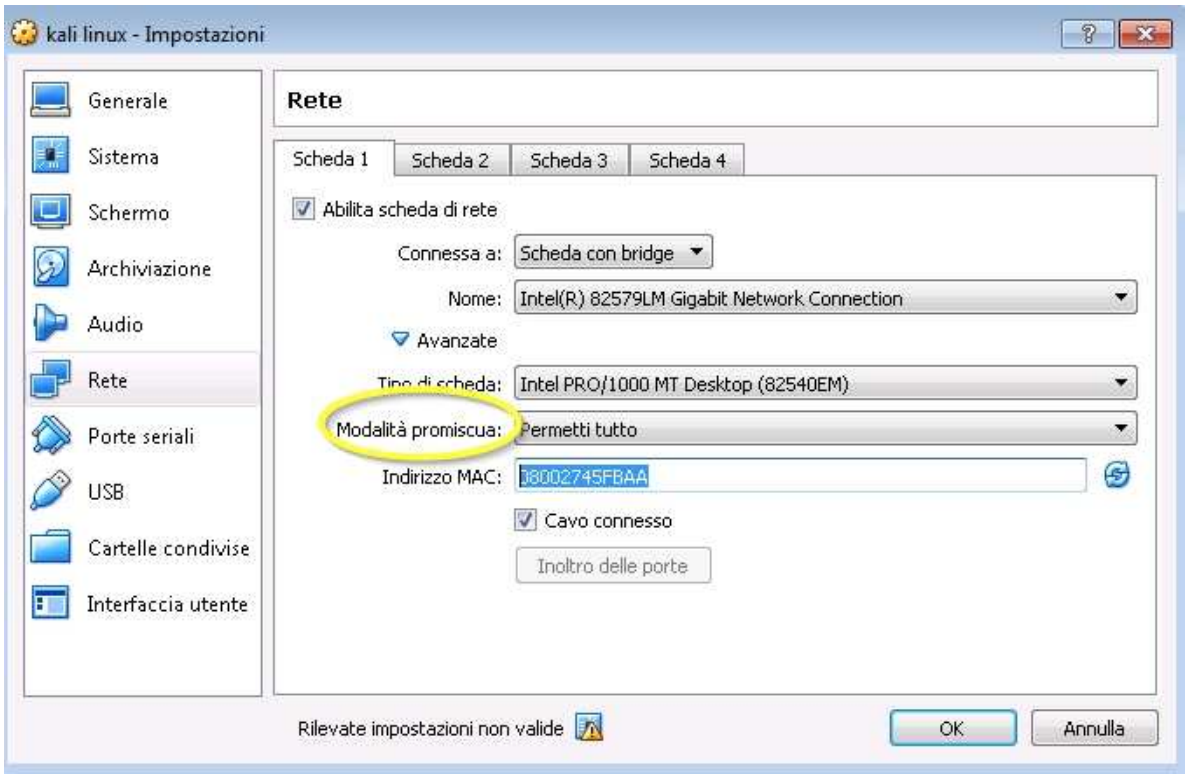
La modalità promiscua permette di avere visibilità su tutti i pacchetti che raggiungono la vostra interfaccia di rete.

Quando è abilitata questa modalità, la scheda di rete accetta tutti i pacchetti consentendone la cattura da parte dello sniffer.

La parte che segue non potete farla (in teoria) in aula didattica in quanto il server dhcp non vi rilascia un indirizzo ip nella modalità bridge (in pratica si potrebbe fare, ma è meglio non divulgarlo)

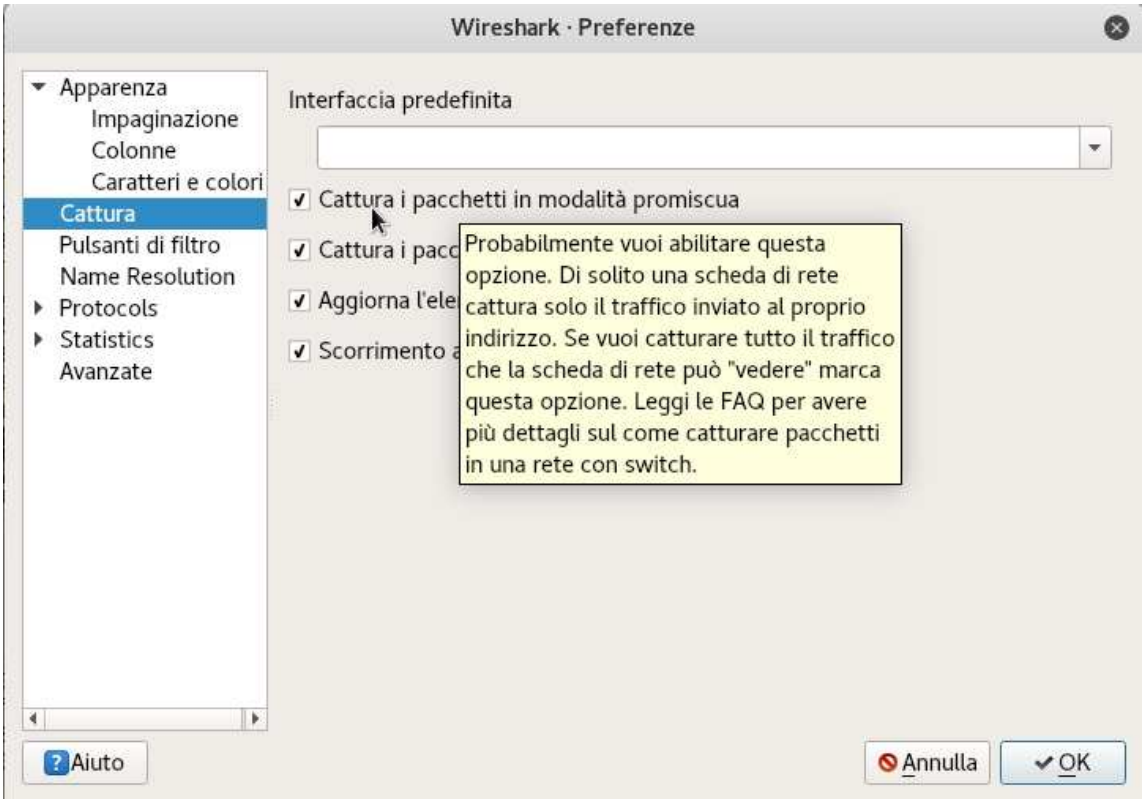
usciamo dalla macchina virtuale e andiamo nelle impostazioni della scheda di rete della machhina virtuale

cambio il mac address, con uno che so che è registrato sul dhcp server:
metto: 08002745FBAA

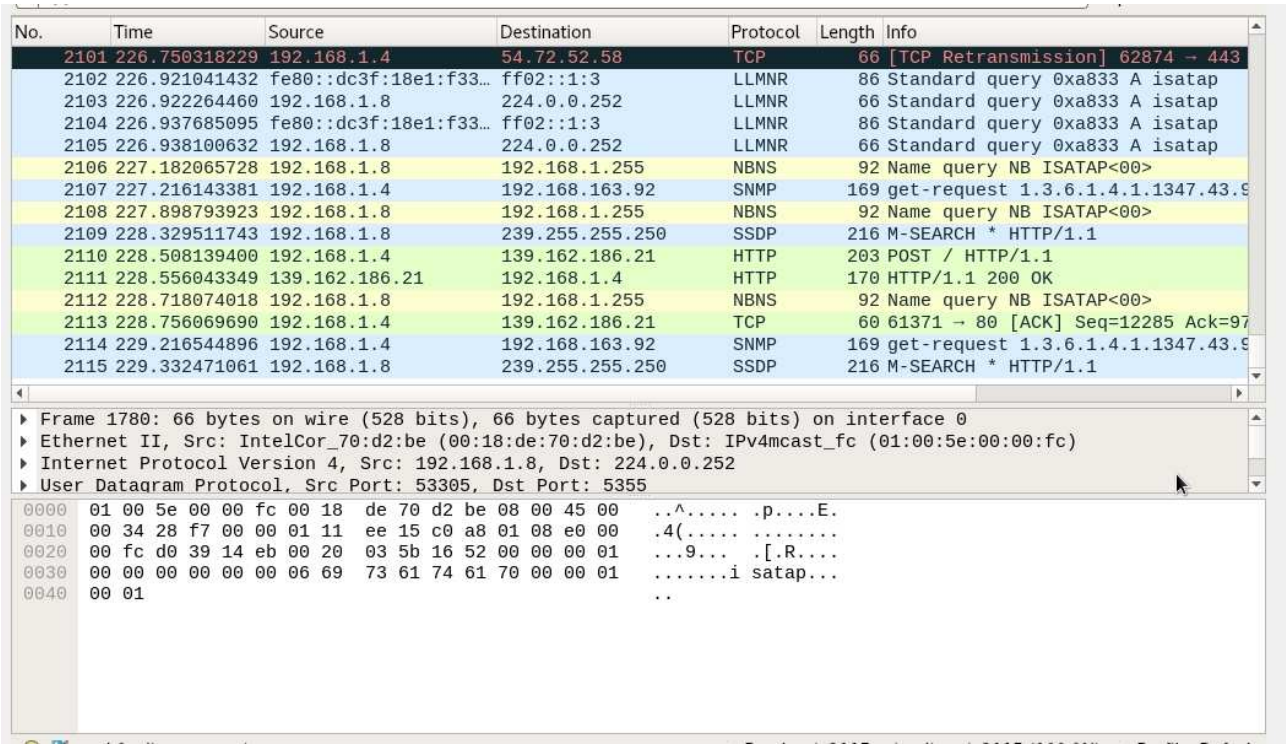


Nella voce Modalità promiscua scelgo Permetti tutto

Faccio partire la macchina virtuale e Wireshark e controllo nelle preferenze che sia abilitata la modalità promiscua



Faccio partire la cattura dei pacchetti e vedete tutto il traffico della rete.



No.	Time	Source	Destination	Protocol	Length	Info
2101	226.750318229	192.168.1.4	54.72.52.58	TCP	66	[TCP Retransmission] 62874 → 443
2102	226.921041432	fe80::dc3f:18e1:f33…	ff02::1:3	LLMNR	86	Standard query 0xa833 A isatap
2103	226.922264460	192.168.1.8	224.0.0.252	LLMNR	66	Standard query 0xa833 A isatap
2104	226.937685095	fe80::dc3f:18e1:f33…	ff02::1:3	LLMNR	86	Standard query 0xa833 A isatap
2105	226.938100632	192.168.1.8	224.0.0.252	LLMNR	66	Standard query 0xa833 A isatap
2106	227.182065728	192.168.1.8	192.168.1.255	NBNS	92	Name query NB ISATAP<00>
2107	227.216143381	192.168.1.4	192.168.163.92	SNMP	169	get-request 1.3.6.1.4.1.1347.43.9
2108	227.898793923	192.168.1.8	192.168.1.255	NBNS	92	Name query NB ISATAP<00>
2109	228.329511743	192.168.1.8	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1
2110	228.508139400	192.168.1.4	139.162.186.21	HTTP	203	POST / HTTP/1.1
2111	228.556043349	139.162.186.21	192.168.1.4	HTTP	170	HTTP/1.1 200 OK
2112	228.718074018	192.168.1.8	192.168.1.255	NBNS	92	Name query NB ISATAP<00>
2113	228.756069690	192.168.1.4	139.162.186.21	TCP	60	61371 → 80 [ACK] Seq=12285 Ack=97
2114	229.216544896	192.168.1.4	192.168.163.92	SNMP	169	get-request 1.3.6.1.4.1.1347.43.9
2115	229.332471061	192.168.1.8	239.255.255.250	SSDP	216	M-SEARCH * HTTP/1.1

```
▸ Frame 1780: 66 bytes on wire (528 bits), 66 bytes captured (528 bits) on interface 0
▸ Ethernet II, Src: IntelCor_70:d2:be (00:18:de:70:d2:be), Dst: IPv4mcast_fc (01:00:5e:00:00:fc)
▸ Internet Protocol Version 4, Src: 192.168.1.8, Dst: 224.0.0.252
▸ User Datagram Protocol, Src Port: 53305, Dst Port: 5355

0000  01 00 5e 00 00 fc 00 18  de 70 d2 be 08 00 45 00   ..^..... .p....E.
0010  00 34 28 f7 00 00 01 11  ee 15 c0 a8 01 08 e0 00   .4(..... ........
0020  00 fc d0 39 14 eb 00 20  03 5b 16 52 00 00 00 01   ...9...  .[.R....
0030  00 00 00 00 00 00 06 69  73 61 74 61 70 00 00 01   .......i satap...
0040  00 01                                             ..
```

Ora che vediamo tutto il traffico possiamo, per esempio intercettare le password di login fatte sui siti non sicuri…..

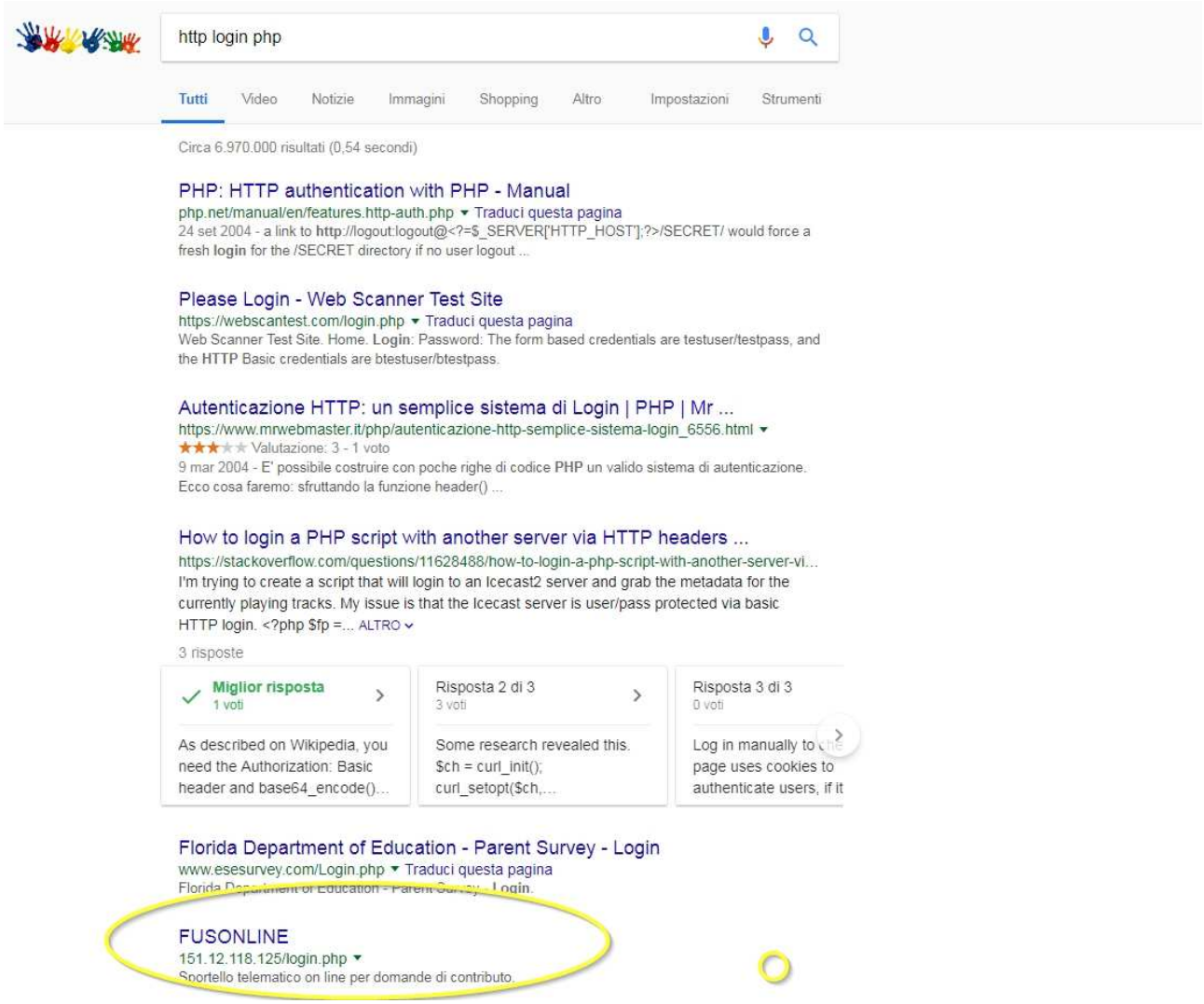
Questa parte potete provarla anche in NAT

Vediamo qualche esempio….

Impostiamo il filtro di cattura su **`http.request.method == "POST"`** e facciamo partire l'ascolto dei pacchetti.

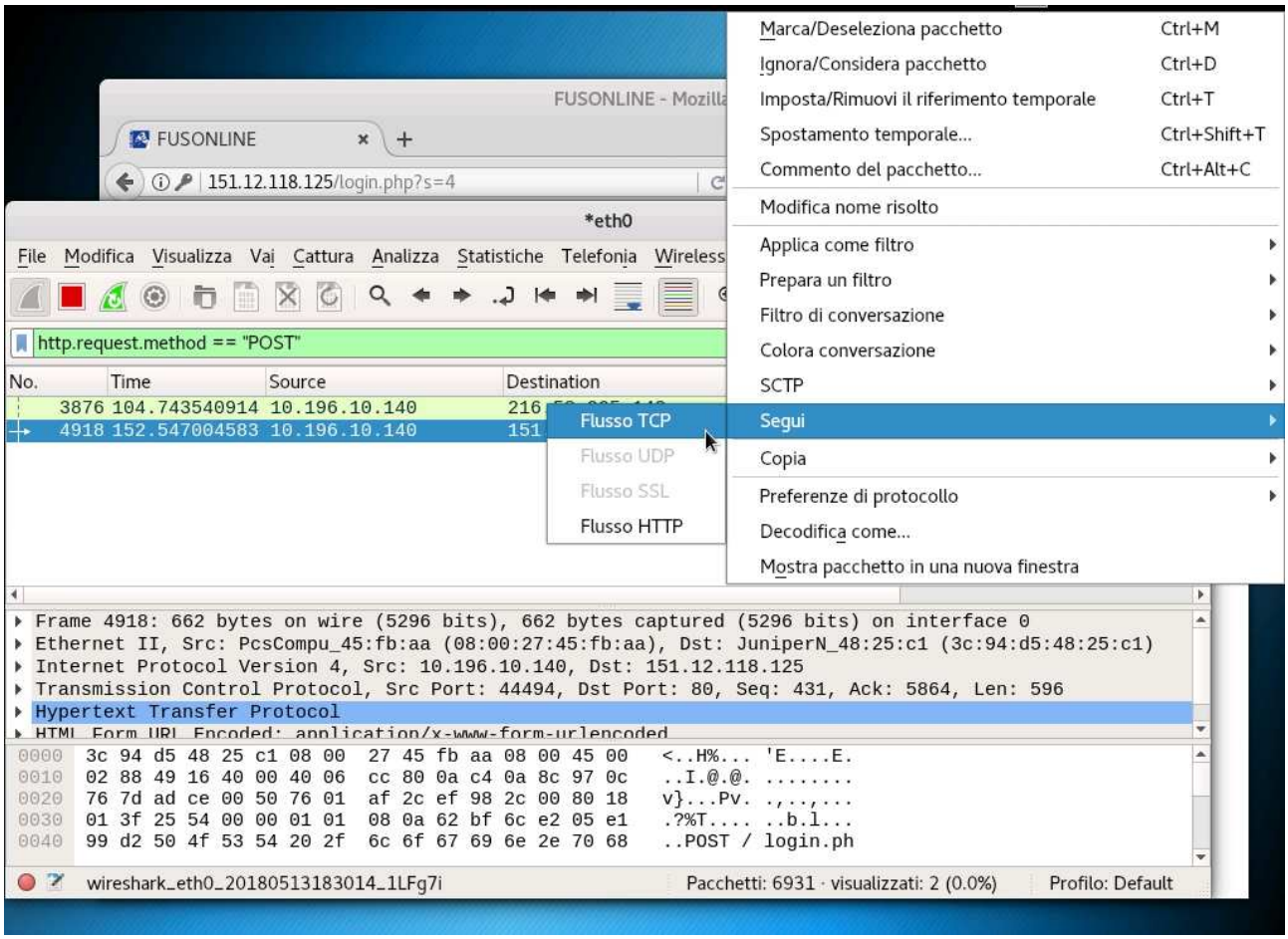
Apriamo un browser e cerchiamo qualche sito con login non sicure.

digitiamo in google: **`http login php`**



Apro il sito evidenziato in giallo, e digitiamo nome utente e password e diamo invio

Torniamo a wireshark, vediamo che ha catturato un pacchetto interessante, selezioniamolo col tasto destro del mouse segliamo segui – flusso tcp



si aprirà una finestra con i dettagli del post appena eseguito,
scorriamo verso la metà e troveremo la login e la password

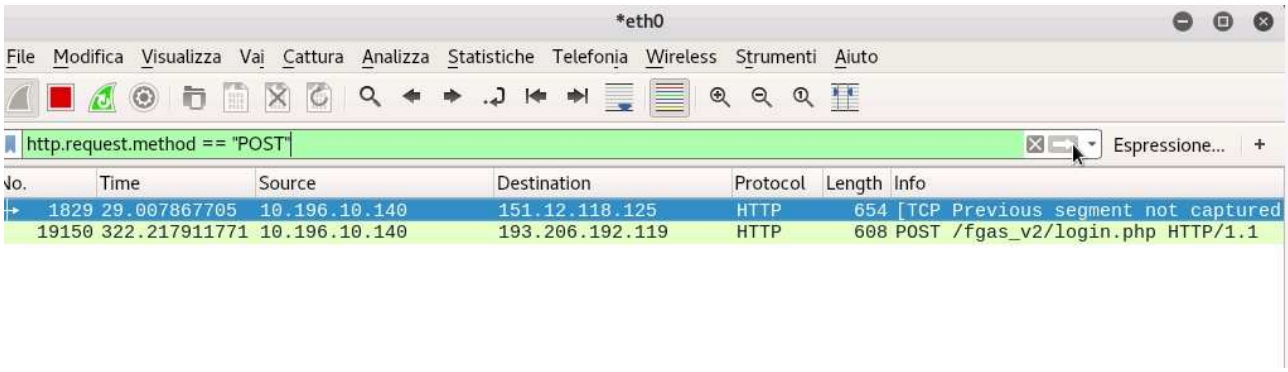
```
                                                        </div>
        </body>
</html>
POST /login.php?s=4 HTTP/1.1
Host: 151.12.118.125
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/
52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://151.12.118.125/login.php
Cookie: _ga=GA1.1.1644076960.1526225759; _gid=GA1.1.58568545.1526225759;
PHPSESSID=7mjorj7dr0gtn300g3kji9bpr0; _gat=1
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 38

email=test%40unitn.it&password=nonlasoHTTP/1.1 200 OK
Date: Sun, 13 May 2018 16:37:57 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
```

Proviamo con un altro sito

Due righe sotto , nella ricerca precedente in google troviamo questo sito:

facciamo partire l'ascolto dei pacchetti e inseriamo login e password:

Il pacchetto interessante è il secondo



Anche in questo caso troviamo la password in chiaro

```
POST /fgas_v2/login.php HTTP/1.1
Host: 193.206.192.119
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:52.0) Gecko/20100101 Firefox/
52.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://193.206.192.119/fgas_v2/login.php
Cookie: PHPSESSID=qqobjbjc9c0u7fnovrndvv2uk6
Connection: keep-alive
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
Content-Length: 43

username=test%40unitn.it&password=boh&type-HTTP/1.1 200 OK
Date: Sun, 13 May 2018 16:27:20 GMT
Server: Apache/2.2.22 (Ubuntu)
X-Powered-By: PHP/5.3.10-1ubuntu3.11
Expires: Mon, 26 Jul 1997 05:00:00 GMT
Cache-Control: private, no-store, no-cache, must-revalidate
```

In altri casi troviamo le password criptate

```
HTTP/1.1 302 Found
Date: Mon, 10 Nov 2014 23:52:21 GMT
Server: Apache/2.2.15 (CentOS)
X-Powered-By: PHP/5.3.3
P3P: CP="NOI ADM DEV PSAi COM NAV OUR OTRo STP IND DEM"
Set-Cookie: non=non; expires=Thu, 07-Nov-2024 23:52:21 GMT; path=/
Set-Cookie: password=e4b7c855be6e3d4307b8d6ba4cd4ab91; expires=Thu, 07-Nov-2024 23:52:21 GMT; path=/
Set-Cookie: scifuser=sampleuser; expires=Thu, 07-Nov-2024 23:52:21 GMT; path=/
Location: loggedin.php
Content-Length: 0
Connection: close
Content-Type: text/html; charset=UTF-8
```

Le credenziali sono quelle segnate in rosso. La password non è mostrata in chiaro), ma è un valore hash.
Per decryptare la password, possiamo provare un bruteforce attack con hascat come visto nella lezione precedente

`hashcat -m 0 -a 0 hash.txt dizionario.txt`