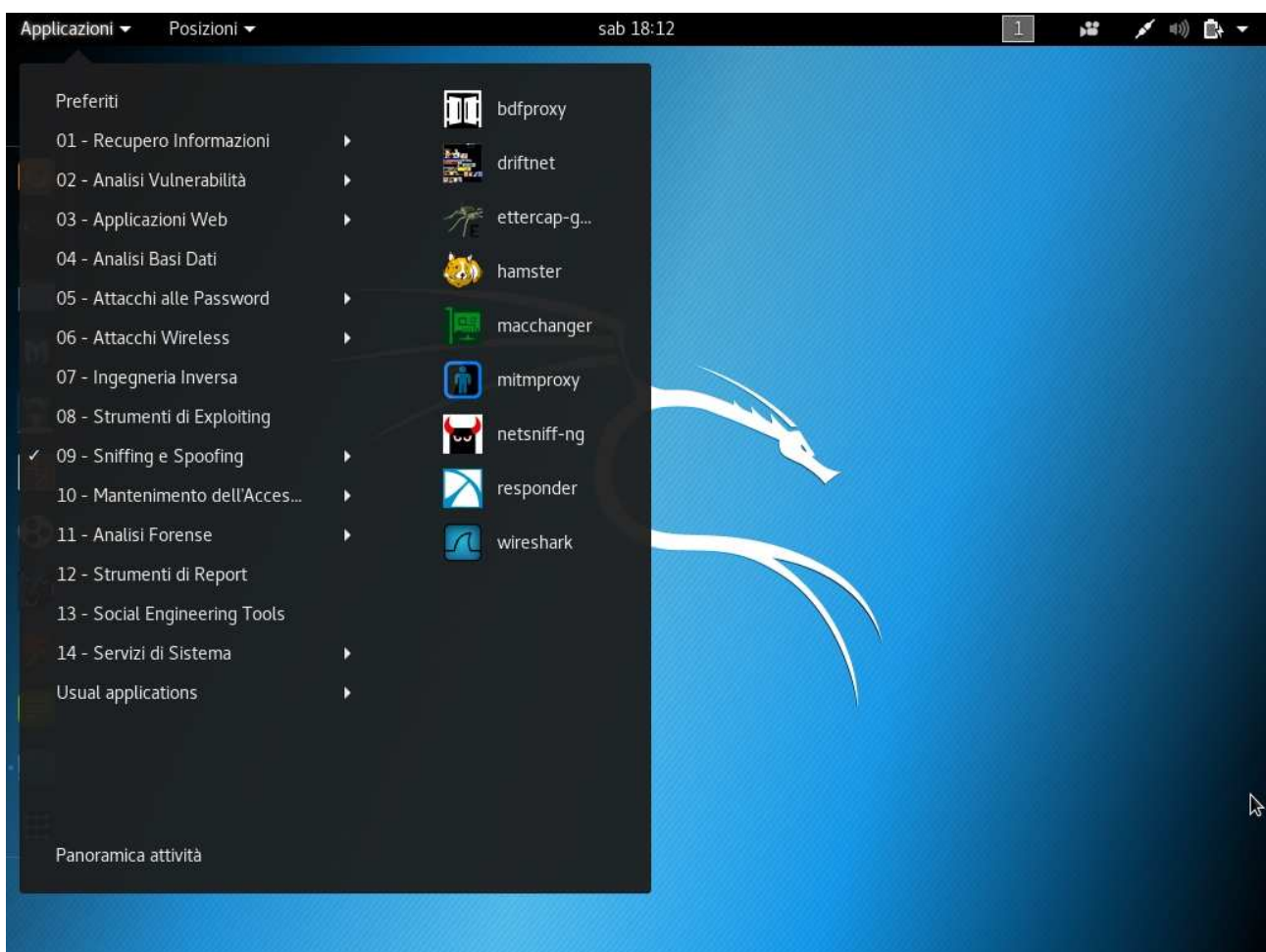


SNIFFING DI DATI DALLA RETE CON WIRESHARK

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.

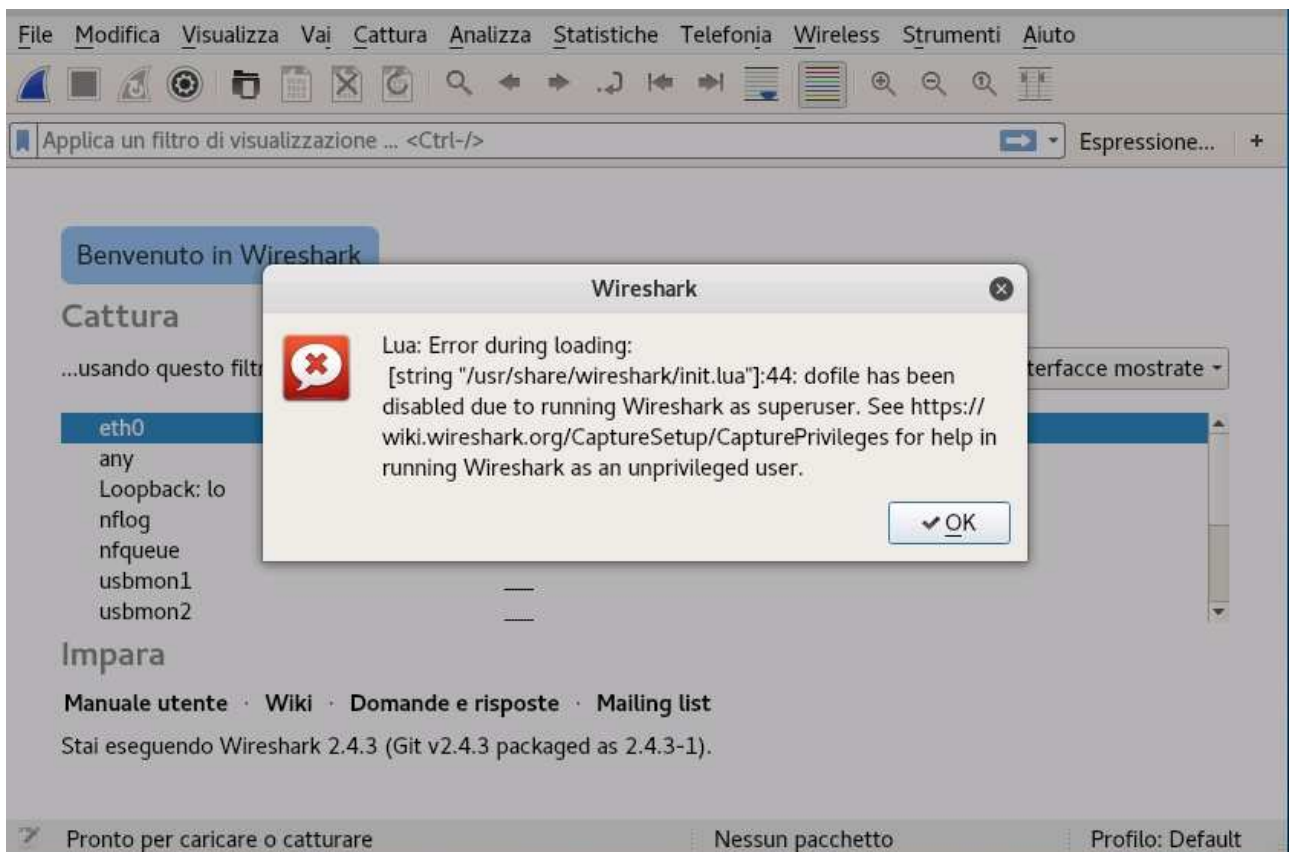


Una volta lanciato il programma, esce un avvertimento che vi ricorda che non è una buona idea catturare pacchetti con un account root per motivi di sicurezza.

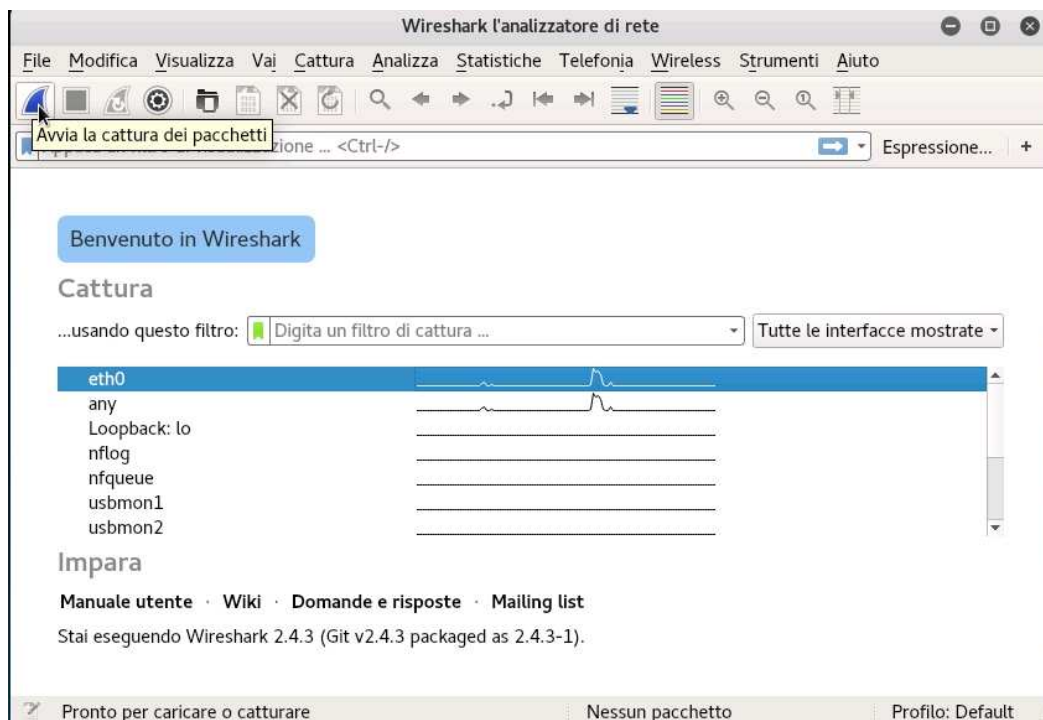
Wireshark esegue l'analisi sintattica di una grande quantità di dati non fidati ed è esposto a vulnerabilità di corruzione della memoria, che potrebbero portare all'esecuzione di codice maligno.

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

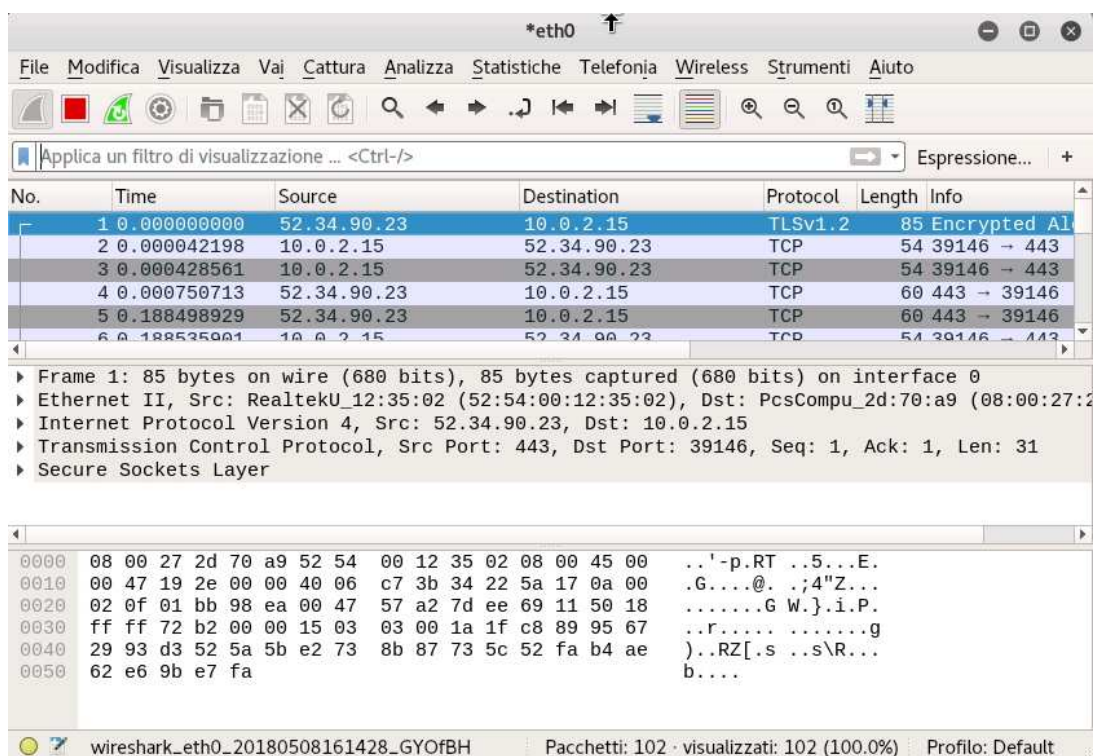
In questa lezione ignoriamo la cosa e continuiamo



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.



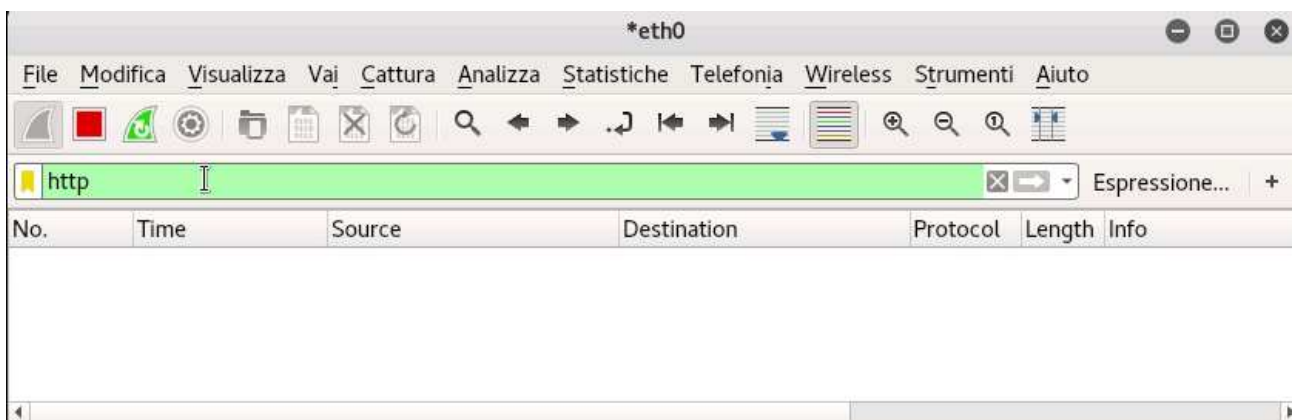
Cliccare sul bottone sulla destra della pinna, a forma di quadrato rosso, per terminare la cattura.

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.

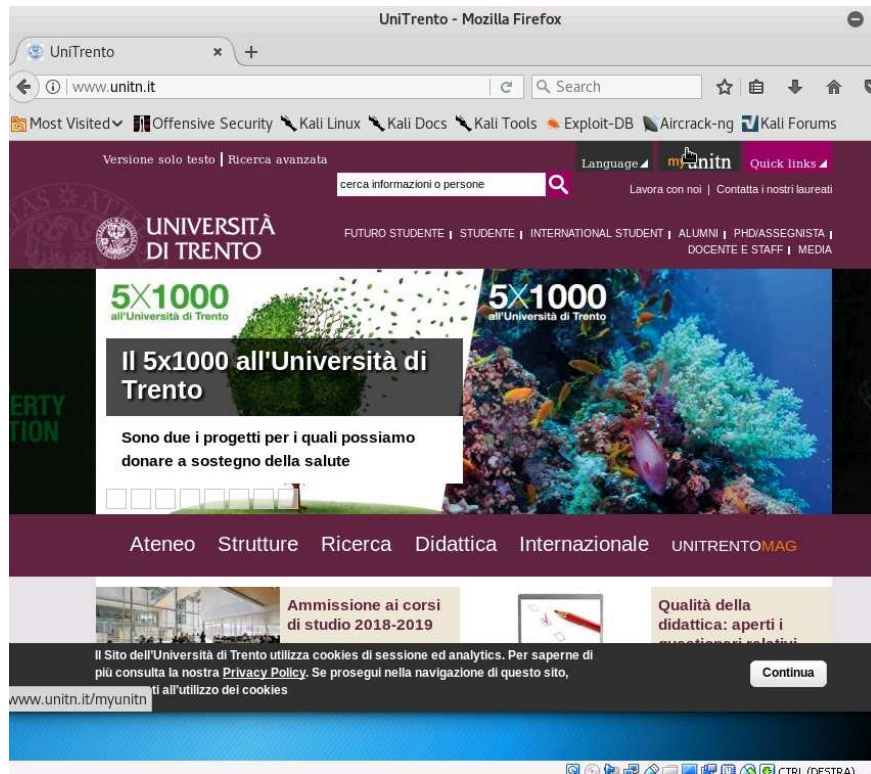
FILTRAGGIO DEL PROTOCOLLO HTTP

Scriviamo ora nella finestra filtro questa voce: http



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

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



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

The screenshot shows a network traffic analysis tool with a menu bar (File, Modifica, Visualizza, Vai, Cattura, Analizza, Statistiche, Telefonja, Wireless, Strumenti, Aiuto) and a toolbar. The main window displays a list of packets with columns: No., Time, Source, Destination, Protocol, Length, and Info. The selected packet (No. 132) is highlighted in blue. Below the list, a detailed view of the selected packet is shown, including the frame details and the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
132	133.483667081	10.0.2.15	192.168.211.63	HTTP	366	GET / HTTP/1.1
164	133.657178998	192.168.211.63	10.0.2.15	HTTP	2901	HTTP/1.1 200 OK (text/html)
173	133.890334928	10.0.2.15	192.168.211.63	HTTP	367	GET /fileswww/favicon_0_0.ico HTTP
179	133.920717363	192.168.211.63	10.0.2.15	HTTP	3836	HTTP/1.1 200 OK (image/vnd.micro
181	133.943056191	10.0.2.15	192.168.211.63	HTTP	376	GET /sites/www.unitn.it/themes/ur
187	133.951076614	10.0.2.15	192.168.211.63	HTTP	423	GET /fileswebmagazine/styles/bann
191	133.952549226	10.0.2.15	192.168.211.63	HTTP	414	GET /fileswebmagazine/styles/bann
197	133.959804255	10.0.2.15	192.168.211.63	HTTP	414	GET /fileswebmagazine/styles/bann
201	133.964162142	10.0.2.15	192.168.211.63	HTTP	414	GET /fileswebmagazine/styles/bann

Frame 132: 366 bytes on wire (2928 bits), 366 bytes captured (2928 bits) on interface 0
 Ethernet II, Src: PcsCompu_2d:70:a9 (08:00:27:2d:70:a9), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)
 Internet Protocol Version 4, Src: 10.0.2.15, Dst: 192.168.211.63
 Transmission Control Protocol, Src Port: 40538, Dst Port: 80, Seq: 1, Ack: 1, Len: 312
 Hypertext Transfer Protocol

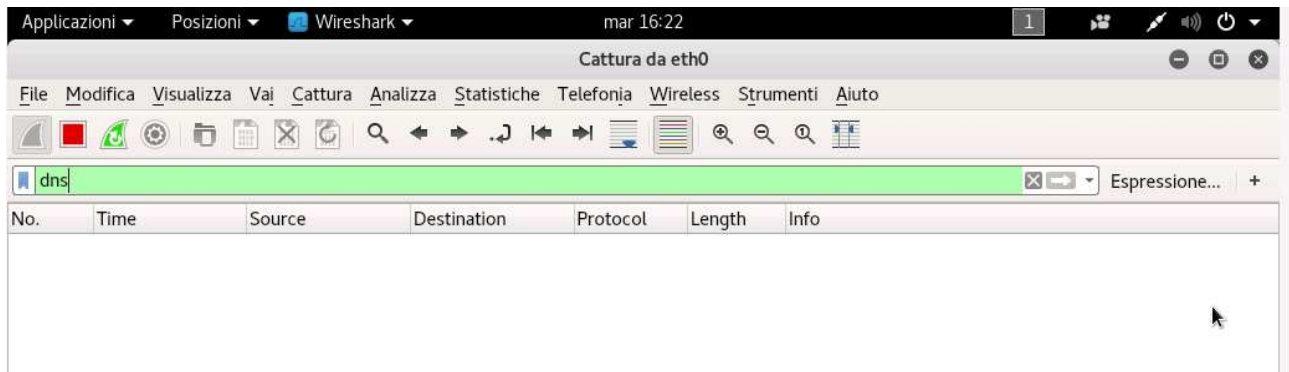
```

0000  52 54 00 12 35 02 08 00 27 2d 70 a9 08 00 45 00  RT..5... '-p...E.
0010  01 60 7d cd 40 00 40 06 1b d4 0a 00 02 0f c0 a8  .}.@.@. ....
0020  d3 3f 9e 5a 00 50 50 ca 79 32 01 91 5e 02 50 18  .?.Z.PP. y2..^P.
0030  72 10 a1 49 00 00 47 45 54 20 2f 20 48 54 54 50  r..I..GE T / HTTP
0040  2f 31 2e 31 0d 0a 48 6f 73 74 3a 20 77 77 77 2e  /1.1..Ho st: www.
0050  75 6e 69 74 6e 2e 69 74 0d 0a 55 73 65 72 2d 41  unitn.it ..User-A
0060  67 65 6e 74 3a 20 4d 6f 7a 69 6c 6c 61 2f 35 2e  gent: Mozilla/5.
0070  30 20 28 58 31 31 3b 20 4c 69 6e 75 78 20 78 38  0 (X11; Linux x8
0080  36 5f 36 34 3b 20 72 76 3a 35 32 2e 30 29 20 47  6_64; rv:52.0) G
0090  65 63 6b 6f 2f 32 30 31 30 30 31 30 31 20 46 69  ecko/201 00101 Fi
00a0  72 65 66 6f 78 2f 35 32 2e 30 0d 0a 41 63 63 65  refox/52 .0..Acce
  
```

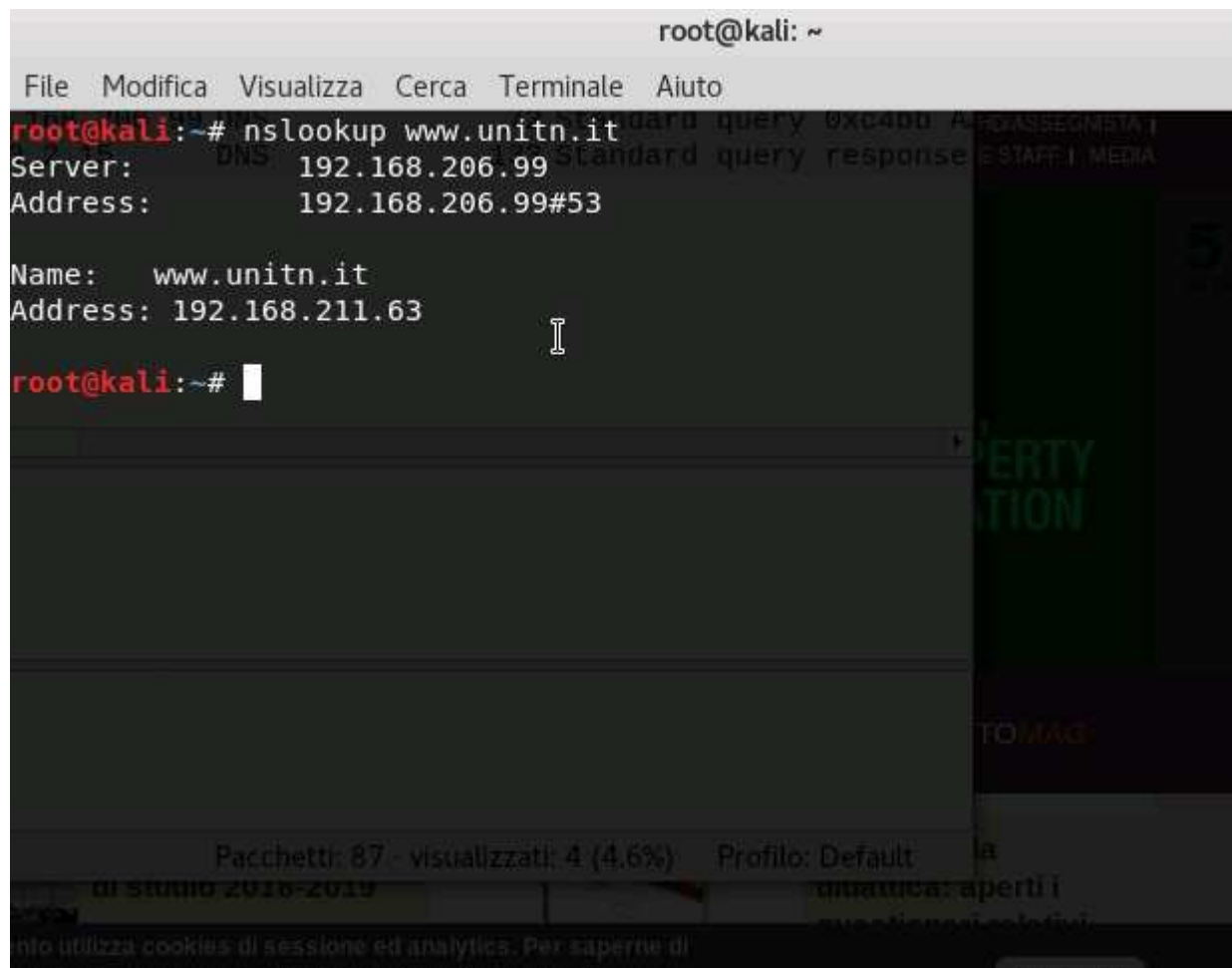
Hypertext Transfer Protocol: Protocol Pacchetti: 620 - visualizzati: 86 (13.9%) Profilo: Default

FILTRAGGIO DEL PROTOCOLLO DNS

Scriviamo ora nella 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.

The screenshot shows the Wireshark interface with a capture on the 'dns' filter. The packet list shows four DNS-related packets:

No.	Time	Source	Destination	Protocol	Length	Info
84	51.937756372	10.0.2.15	192.168.206.99	DNS	72	Standard query 0xfc1f A www.unitn.it
85	51.950012796	192.168.206.99	10.0.2.15	DNS	88	Standard query response 0xfc1f A www.unitn.it A 192.168.206.99
86	51.950834027	10.0.2.15	192.168.206.99	DNS	72	Standard query 0xc4bb AAAA www.unitn.it
87	51.953446758	192.168.206.99	10.0.2.15	DNS	133	Standard query response 0xc4bb AAAA www.unitn.it SOA

The packet details pane for packet 84 shows:

- Frame 84: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface 0
- Ethernet II, Src: PcsCompu_2d:70:a9 (08:00:27:2d:70:a9), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)
- Internet Protocol Version 4, Src: 10.0.2.15, Dst: 192.168.206.99
- User Datagram Protocol, Src Port: 52164, Dst Port: 53
- Domain Name System (query)

The packet bytes pane shows the raw data in hexadecimal and ASCII:

```

0000  52 54 00 12 35 02 08 00 27 2d 70 a9 08 00 45 00  RT..5... '-p...E.
0010  00 3a f1 e1 00 00 40 11 ed b6 0a 00 02 0f c0 a8  .....@. ....
0020  ce 63 cb c4 00 35 00 26 9b 52 fc 1f 01 00 00 01  .c...5.& .R....
0030  00 00 00 00 00 00 03 77 77 77 05 75 6e 69 74 6e  ....w ww.unitn
0040  02 69 74 00 00 01 00 01  .it....
  
```

At the bottom, the status bar indicates: eth0: <live capture in progress> Pacchetti: 87 · visualizzati: 4 (4.6%) Profilo: Default

FILTRAGGIO DEL PROTOCOLLO ICMP

Scriviamo ora nella 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.

The image shows a Wireshark packet capture window with the filter 'icmp'. The packet list shows 28 packets, alternating between Echo (ping) requests and replies. The packet details pane shows the structure of an ICMP Echo (ping) request, including the type, code, identifier, and sequence number. The packet bytes pane shows the raw data in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
5	0.024653133	10.0.2.15	192.168.131.103	ICMP	98	Echo (ping) request id=0x0726, seq=1/256, ttl=64
6	0.048032786	192.168.131.103	10.0.2.15	ICMP	98	Echo (ping) reply id=0x0726, seq=1/256, ttl=59
9	1.026242013	10.0.2.15	192.168.131.103	ICMP	98	Echo (ping) request id=0x0726, seq=2/512, ttl=64
10	1.027506499	192.168.131.103	10.0.2.15	ICMP	98	Echo (ping) reply id=0x0726, seq=2/512, ttl=59
11	2.027914550	10.0.2.15	192.168.131.103	ICMP	98	Echo (ping) request id=0x0726, seq=3/768, ttl=64
12	2.029017279	192.168.131.103	10.0.2.15	ICMP	98	Echo (ping) reply id=0x0726, seq=3/768, ttl=59
13	3.029581186	10.0.2.15	192.168.131.103	ICMP	98	Echo (ping) request id=0x0726, seq=4/1024, ttl=64
14	3.031078092	192.168.131.103	10.0.2.15	ICMP	98	Echo (ping) reply id=0x0726, seq=4/1024, ttl=59
15	4.031697572	10.0.2.15	192.168.131.103	ICMP	98	Echo (ping) request id=0x0726, seq=5/1280, ttl=64
16	4.033081850	192.168.131.103	10.0.2.15	ICMP	98	Echo (ping) reply id=0x0726, seq=5/1280, ttl=59
17	5.033855204	10.0.2.15	192.168.131.103	ICMP	98	Echo (ping) request id=0x0726, seq=6/1536, ttl=64
18	5.035263013	192.168.131.103	10.0.2.15	ICMP	98	Echo (ping) reply id=0x0726, seq=6/1536, ttl=59
21	6.035454900	10.0.2.15	192.168.131.103	ICMP	98	Echo (ping) request id=0x0726, seq=7/1792, ttl=64
22	6.036850225	192.168.131.103	10.0.2.15	ICMP	98	Echo (ping) reply id=0x0726, seq=7/1792, ttl=59
23	7.037576540	10.0.2.15	192.168.131.103	ICMP	98	Echo (ping) request id=0x0726, seq=8/2048, ttl=64
24	7.038948362	192.168.131.103	10.0.2.15	ICMP	98	Echo (ping) reply id=0x0726, seq=8/2048, ttl=59
25	8.039700681	10.0.2.15	192.168.131.103	ICMP	98	Echo (ping) request id=0x0726, seq=9/2304, ttl=64
26	8.040970427	192.168.131.103	10.0.2.15	ICMP	98	Echo (ping) reply id=0x0726, seq=9/2304, ttl=59
27	9.040444529	10.0.2.15	192.168.131.103	ICMP	98	Echo (ping) request id=0x0726, seq=10/2560, ttl=64
28	9.041931922	192.168.131.103	10.0.2.15	ICMP	98	Echo (ping) reply id=0x0726, seq=10/2560, ttl=59

Frame 5: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interface 0
 Ethernet II, Src: PcsCompu_2d:70:a9 (08:00:27:2d:70:a9), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)
 Internet Protocol Version 4, Src: 10.0.2.15, Dst: 192.168.131.103
 Internet Control Message Protocol

0000 52 54 00 12 35 02 08 00 27 2d 70 a9 08 00 45 00 RT..5...!-p...E.
 0010 00 54 af 02 40 00 40 01 3b 88 0a 00 02 0f c0 a8 .T..@. ;.....
 0020 83 67 08 00 b8 b9 07 26 00 01 c4 b5 f1 5a 00 00 .g.....&Z..
 0030 00 00 b8 3b 0b 00 00 00 00 00 10 11 12 13 14 15 ...;.....
 0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 !"#\$\$%
 0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 &'()*+,-./012345

FILTRAGGIO DEL PROTOCOLLO SSH

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

Apriamo un terminale e scriviamo:

```
ssh retiavanzatex.disi.unitn.it
```

```
root@kali:~# ssh retiavanzatex.disi.unitn.it
The authenticity of host 'retiavanzatex.disi.unitn.it (192.168.131.103)' can't be
established.
ECDSA key fingerprint is SHA256:SEAF9dCwsYb0Zm/zmag4lnx020JYRPlcnlgCz01fcwE.
Are you sure you want to continue connecting (yes/no)?
root@kali:~# clear
root@kali:~# ssh retiavanzatex.disi.unitn.it
The authenticity of host 'retiavanzatex.disi.unitn.it (192.168.131.103)' can't be
established.
ECDSA key fingerprint is SHA256:SEAF9dCwsYb0Zm/zmag4lnx020JYRPlcnlgCz01fcwE.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'retiavanzatex.disi.unitn.it' (ECDSA) to the list of
known hosts.
root@retiavanzatex.disi.unitn.it's password:
Permission denied, please try again.
root@retiavanzatex.disi.unitn.it's password:
```

No.	Time	Source	Destination	Protocol	Length	Info
8	0.012918508	10.0.2.15	192.168.131.103	SSHv2	86	Client: Protocol (SSH-2.0-OpenSSH_7.6p1 Debi
10	0.043464062	192.168.131.103	10.0.2.15	SSHv2	95	Server: Protocol (SSH-2.0-OpenSSH_7.2p2 U
12	0.043786860	10.0.2.15	192.168.131.103	SSHv2	1414	Client: Key Exchange Init
14	0.045502681	192.168.131.103	10.0.2.15	SSHv2	1030	Server: Key Exchange Init
15	0.047633282	10.0.2.15	192.168.131.103	SSHv2	102	Client: Elliptic Curve Diffie-Hellman Key
17	0.070620014	192.168.131.103	10.0.2.15	SSHv2	418	Server: Elliptic Curve Diffie-Hellman Key
19	7.361660174	10.0.2.15	192.168.131.103	SSHv2	70	Client: New Keys
21	7.362530846	10.0.2.15	192.168.131.103	SSHv2	98	Client: Encrypted packet (len=44)
23	7.400738151	192.168.131.103	10.0.2.15	SSHv2	98	Server: Encrypted packet (len=44)
25	7.400878780	10.0.2.15	192.168.131.103	SSHv2	114	Client: Encrypted packet (len=60)
27	7.407343016	192.168.131.103	10.0.2.15	SSHv2	106	Server: Encrypted packet (len=52)
29	26.283108595	10.0.2.15	192.168.131.103	SSHv2	138	Client: Encrypted packet (len=84)
31	28.516875504	192.168.131.103	10.0.2.15	SSHv2	106	Server: Encrypted packet (len=52)

Frame 8: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0	
Ethernet II, Src: PcsCompu_2d:70:a9 (08:00:27:2d:70:a9), Dst: RealtekU_12:35:02 (52:54:00:12:35:02)	
Internet Protocol Version 4, Src: 10.0.2.15, Dst: 192.168.131.103	
Transmission Control Protocol, Src Port: 52172, Dst Port: 22, Seq: 1, Ack: 1, Len: 32	
0000	52 54 00 12 35 02 08 00 27 2d 70 a9 08 00 45 00 RT..5... '-p...E.
0010	00 48 03 e3 40 00 40 06 e6 ae 0a 00 02 0f c0 a8 .H...@.
0020	83 67 cb cc 00 16 6d d8 89 c9 06 0c d4 02 50 18 .g...m.P.
0030	72 10 50 59 00 00 53 53 48 2d 32 2e 30 2d 4f 70 r.PY..SS H-2.0-Op
0040	65 6e 53 53 48 5f 37 2e 36 70 31 20 44 65 62 69 enSSH_7. 6p1 Debi

wireshark_eth0_20180508164500_zWU2oJ Pacchetti: 36 · visualizzati: 13 (36.1%) Profilo: Default

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.

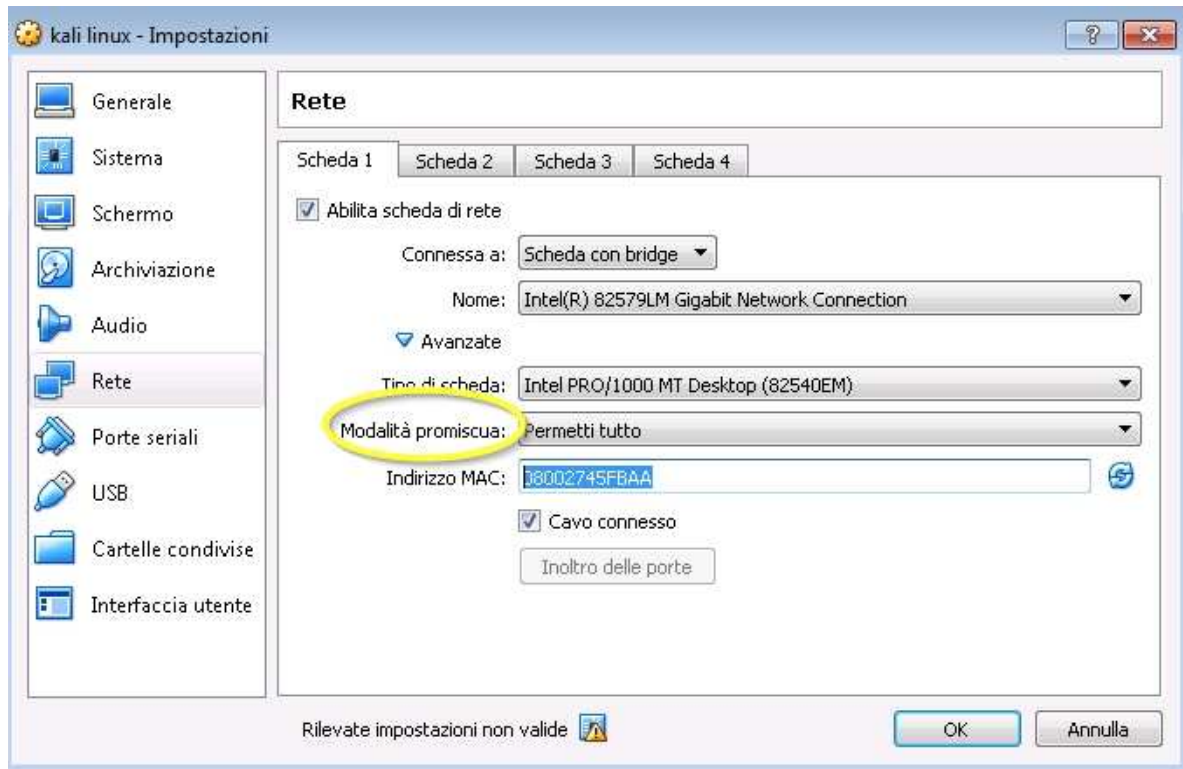
La modalità promiscua permette di avere visibilità su tutti i pacchetti che raggiungono la vostra interfaccia di rete.

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)

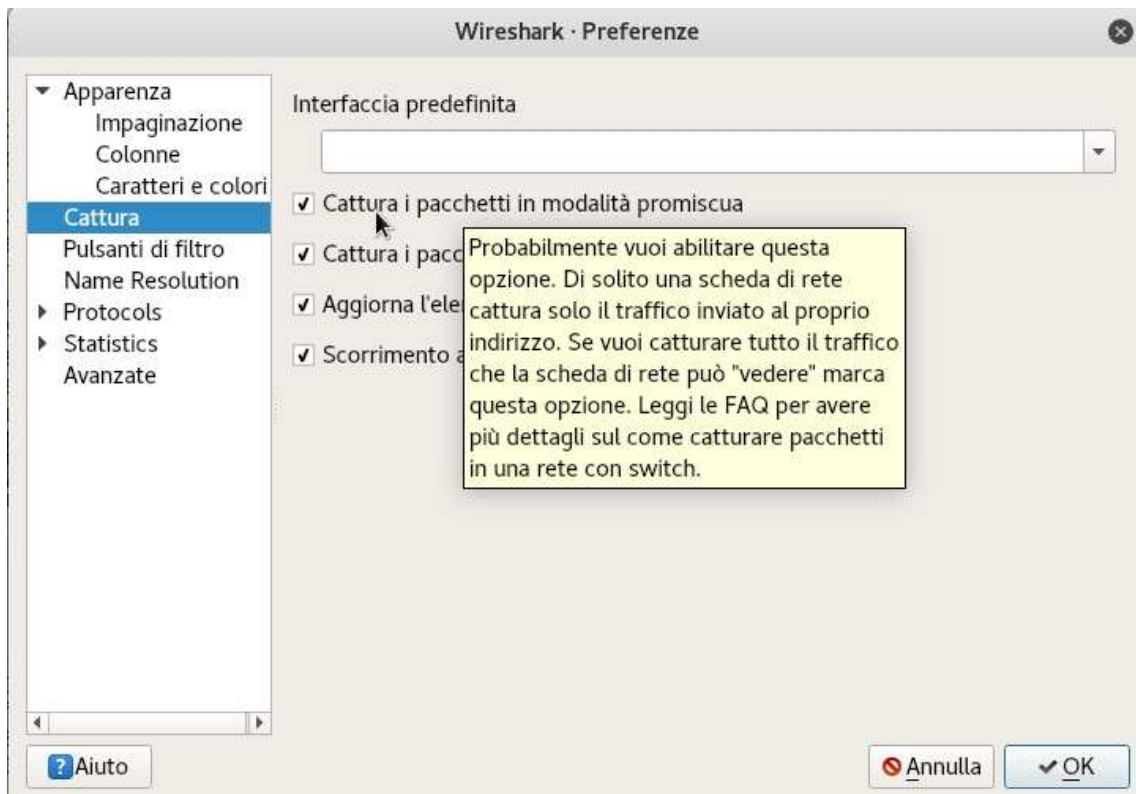
usciamo dalla macchina virtuale e andiamo nelle impostazioni della scheda di rete della macchina virtuale

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 1.3.6.1.4.1.1347.43.5
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 1.3.6.1.4.1.1347.43.5
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

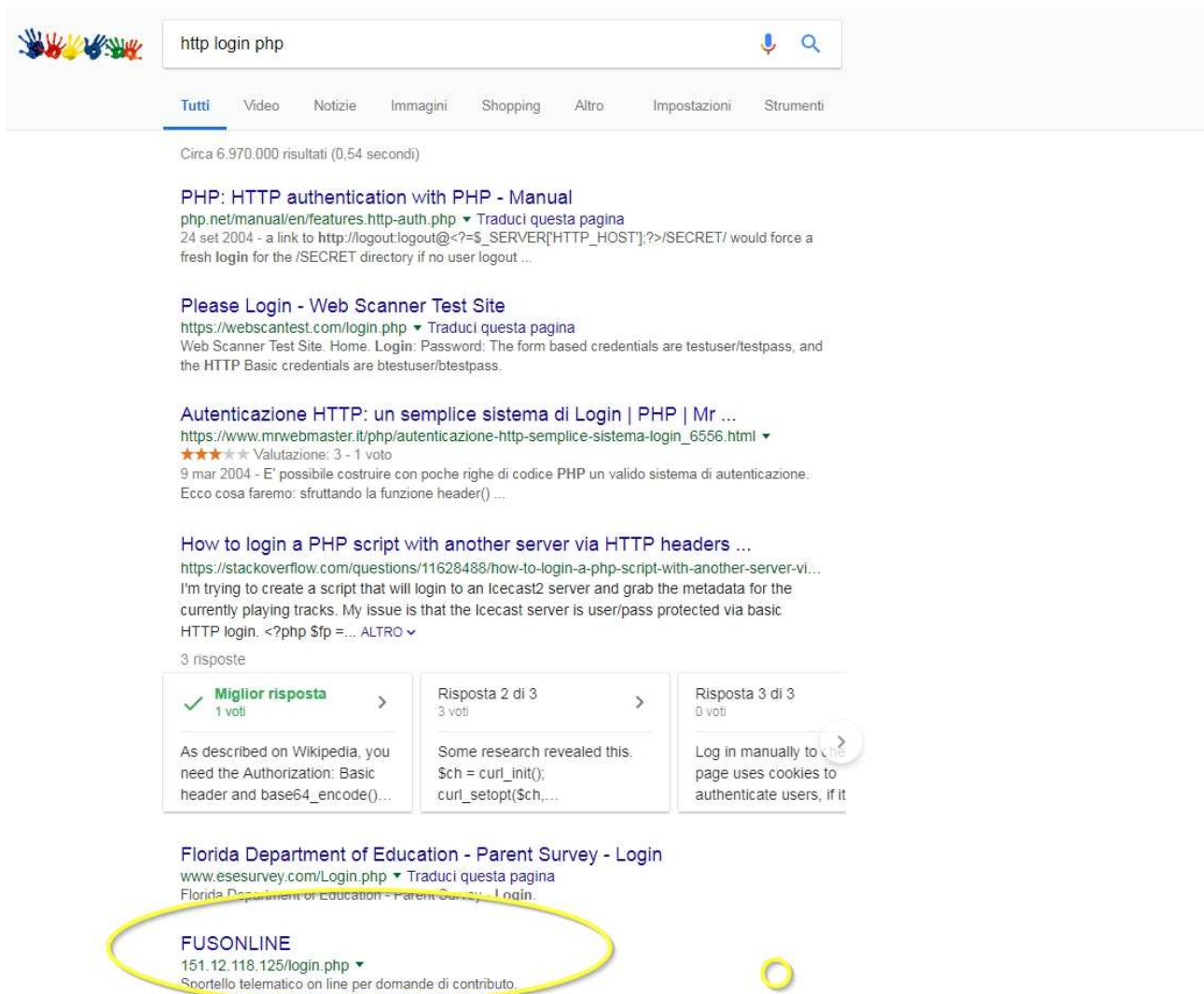
Vediamo qualche esempio....

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



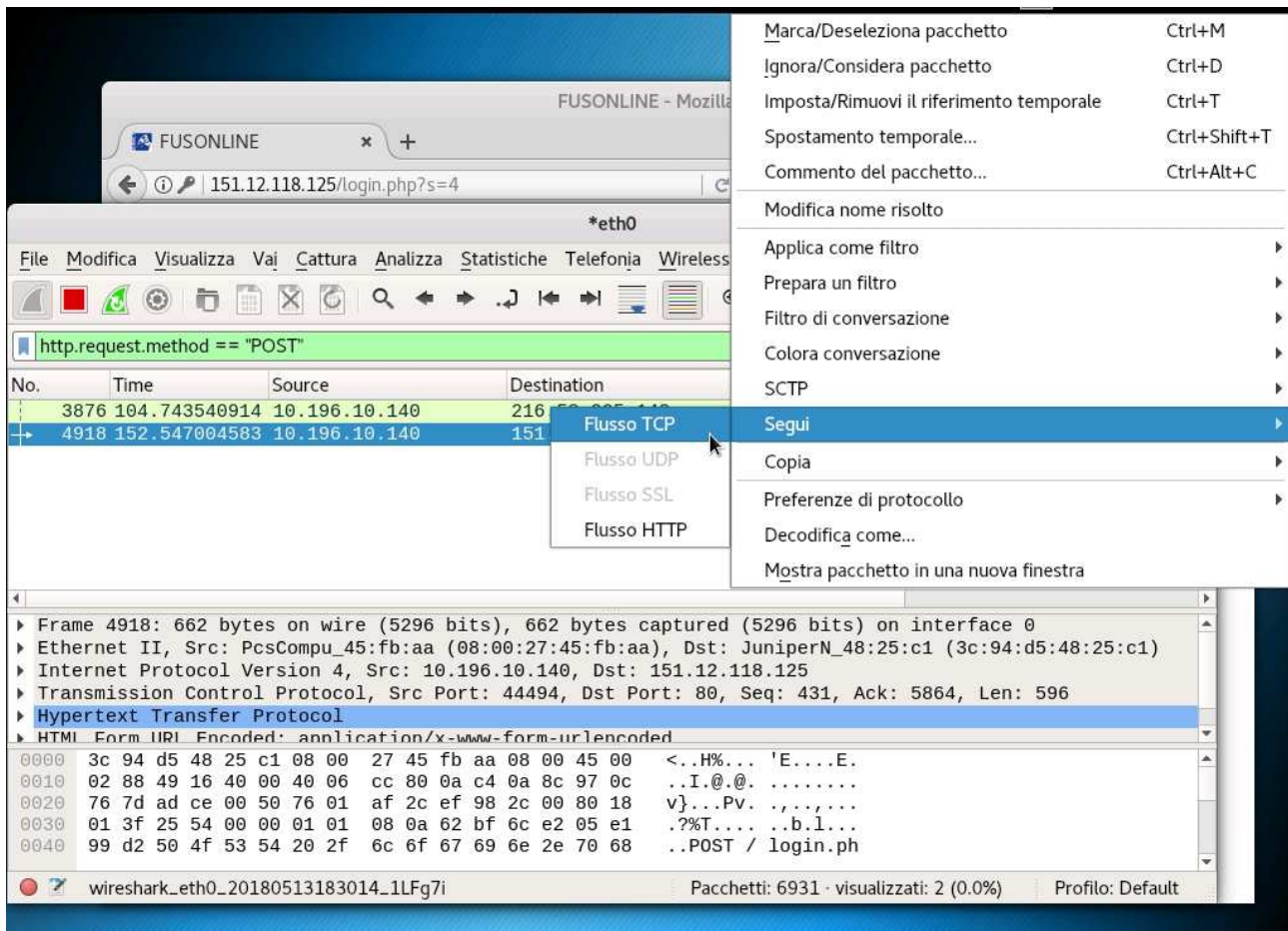
Apriamo un browser e cerchiamo qualche sito con login non sicure.

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 scegliamo 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=nonlasoHT HTTP/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:

FUSONLINE

151.12.118.125/login.php ▼

Sportello telematico on line per domande di contributo.

Making a login form using PHP - HTML Form Guide

form.guide/php-form/php-login-form.html ▼ Traduci questa pagina

Shows how to create a membership based web site using PHP. This is the second step where you create the login form.

Dichiarazione ai sensi dell'art.16, comma 1, del D.P.R. del 27 gennaio ...

193.206.192.119/fgas_v2/login.php ▼

Dichiarazione ai sensi dell'art.16, comma 1, del D.P.R. del 27 gennaio 2012, n.43 - Homer, Login. Nome utente. Password. Auto login fino a quando non effettuo ...

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

Caricamenti massivi
Login

SINAnet

ISPRA
Istituto Superiore per la Protezione e la Ricerca Ambientale

Rete del sistema Informativo Nazionale Ambientale

Dichiarazione ai sensi dell'art.16, comma 1, del D.P.R. del 27 gennaio 2012, n.43

[Home](#) / [Login](#)

Nome utente: test@unitn.it

Password: ●●●

☐ Auto login fino a quando non effettuo il logout

☐ Salva il mio nome utente

☒ Richiedimi sempre username e password

[Login](#)

[Recupera credenziali](#) [Registrazione](#)

Il pacchetto interessante è il secondo

*eth0

File Modifica Visualizza Vai Cattura Analizza Statistiche Telefonia Wireless Strumenti Aiuto

http.request.method == "POST"

No.	Time	Source	Destination	Protocol	Length	Info
1829	29.007867705	10.196.10.140	151.12.118.125	HTTP	654	[TCP Previous segment not captured]
19150	322.217911771	10.196.10.140	193.206.192.119	HTTP	608	POST /fgas_v2/login.php HTTP/1.1

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=samplouser; 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
```