

ANALISI DELLE RETI

WIRESHARK

Wireshark è uno strumento software per catturare ed interpretare i dati che transitano su una rete.

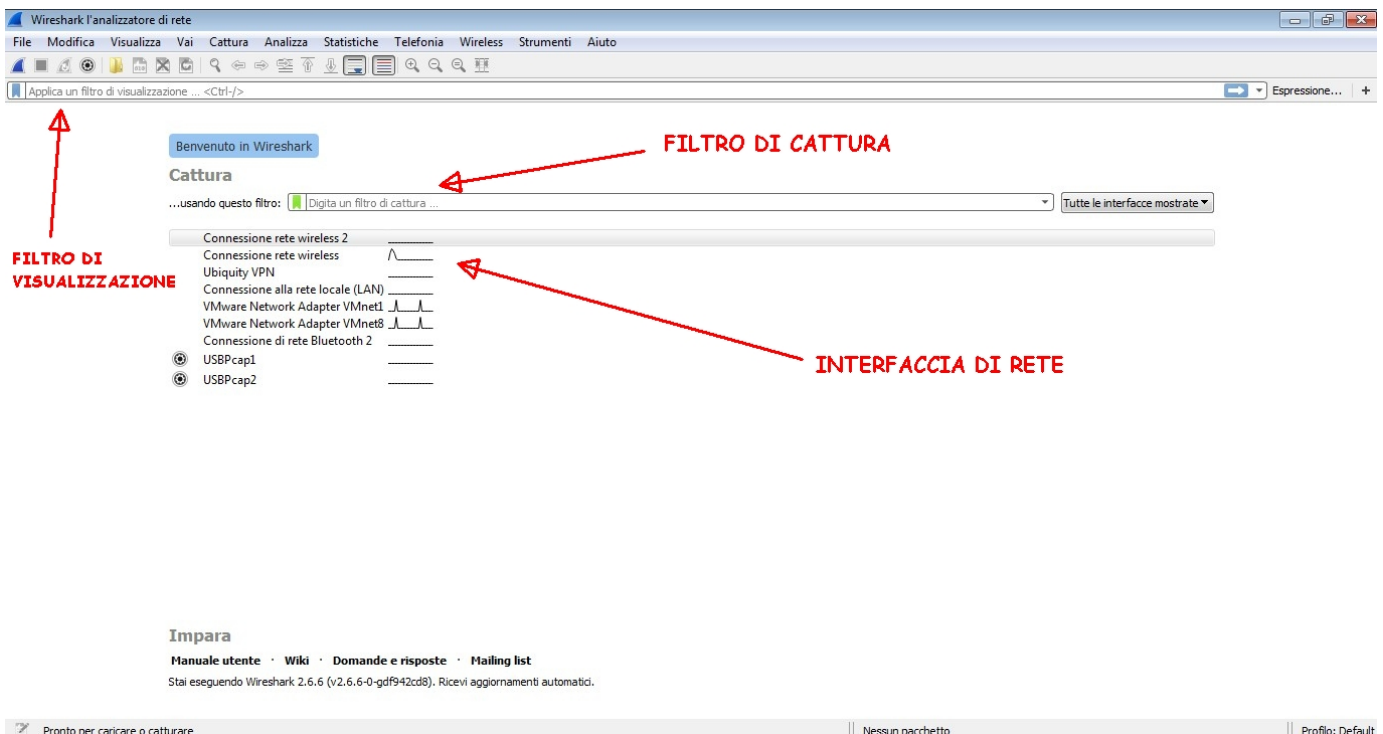


In una rete transitano dati sottoforma di *frame*, *datagrammi*, *pacchetti* o *segmenti* a seconda del loro contenuto, possiamo omunque genericamente riferirci al termine **pacchetti** di dati.

Wireshark, è un programma definito anche **analizzatore di rete**, scaricabile liberamente dal sito www.wireshark.org, e disponibile per diverse piattaforme.

Wireshark cattura i dati dall'interfaccia di rete selezionata e poi suddivide quanto catturato in pacchetti, stabilendo dove iniziano e dove finiscono. Poi interpreta e presenta i dati secondo il protocollo utilizzato. I dati catturati possono essere analizzati, subito o salvati per una successiva valutazione ed analisi ed i protocolli gestiti sono molteplici.

La G.U.I. (**G**raphical **U**ser **I**nterface) di Wireshark all'avvio si presenta in questo modo.



Tramite i 3 campi evidenziati possiamo scegliere:

- su quale interfaccia di rete catturare i pacchetti dati, ad esempio sulla wifi o sulla scheda ethernet
- quale filtro di cattura utilizzare, cioè possiamo scegliere se catturare ad esempio solo i pacchetti dati che utilizzano un protocollo TCP diretti alla porta 22.
- quale filtro di visualizzazione utilizzare, cioè possiamo scegliere cosa visualizzare anche in questo caso in base al protocollo utilizzato o in base ad altre caratteristiche che vedremo più avanti.

Senza impostare alcun filtro di cattura proviamo a cliccare sull'interfaccia di rete che attualmente stiamo utilizzando, ad esempio la connessione rete Wireless. In questo modo inizierà la cattura dei pacchetti che verranno visualizzati al centro della pagina.

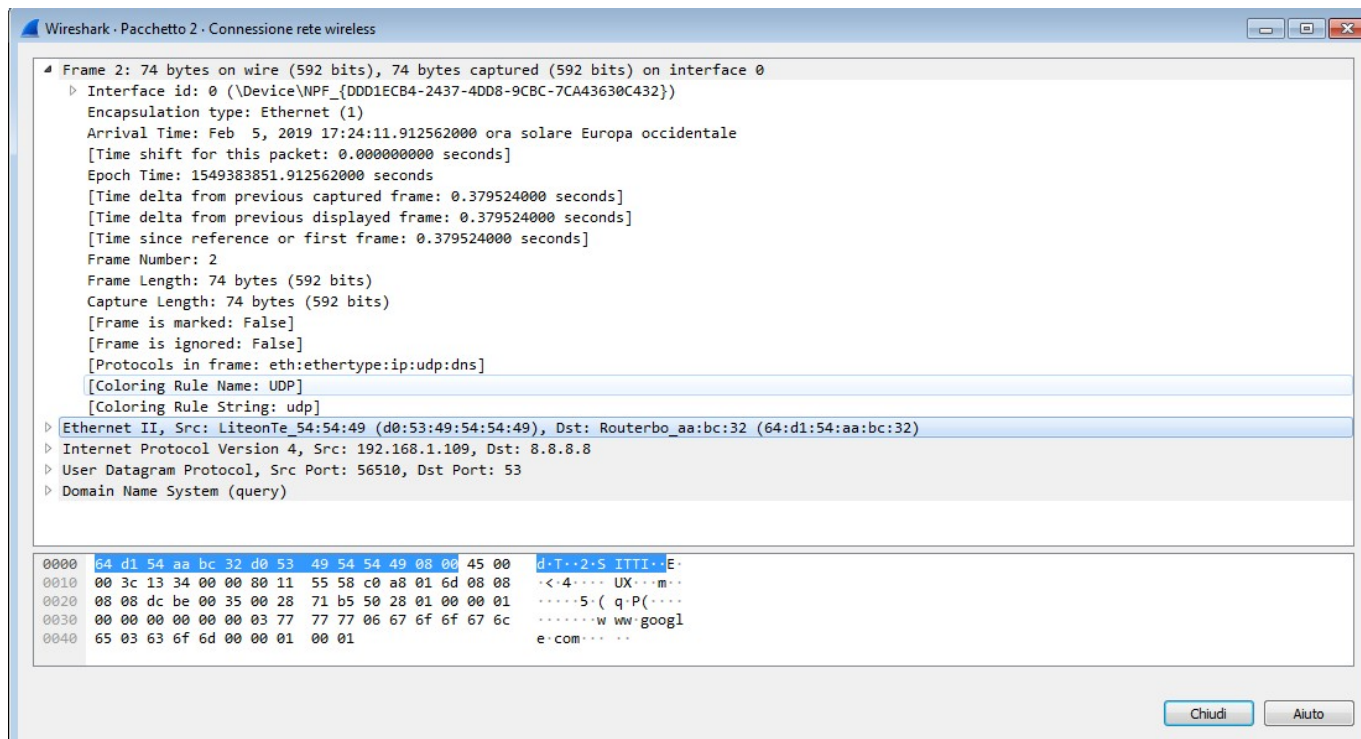
I pacchetti catturati vengono visualizzati di diverso colore a seconda del tipo di protocollo.

<i>Tempo</i>	<i>IP Sorgente</i>	<i>IP Destinazione</i>	<i>Protocollo</i>	<i>Lunghezza</i>	<i>Informazioni</i>
0.000000	192.168.1.64	230.0.0.1	UDP	92	52143 + 6666 Len=50
0.379524	192.168.1.109	8.8.8.8.	DNS	74	Standard Query 0x5028 a www.google.com

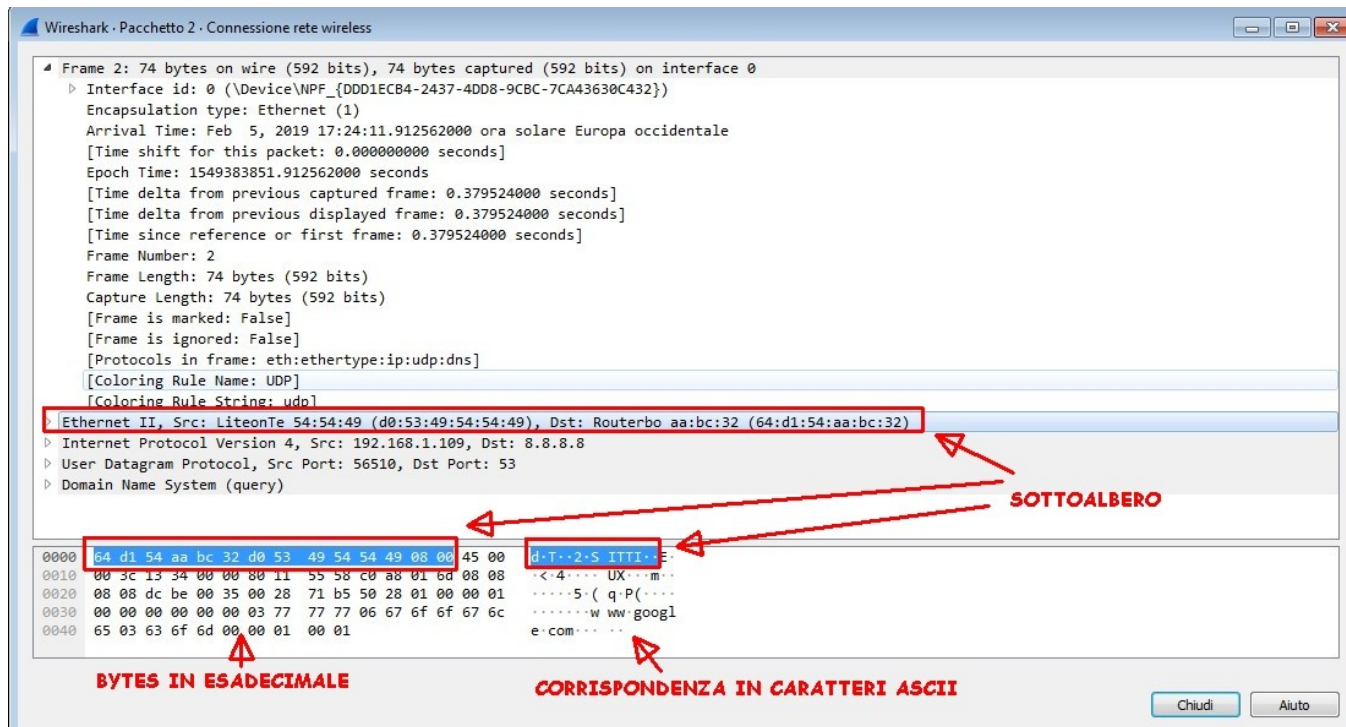
The screenshot displays the Wireshark network protocol analyzer interface. The top menu bar includes File, Modifica, Visualizza, Vai, Cattura, Analisi, Statistiche, Telefono, Wireless, Strumenti, and Aiuto. The toolbar contains various icons for file operations, capture, and analysis. The main window is divided into three panes:

- Packet List:** Shows a list of captured packets. Packet 1 is a UDP packet from 192.168.1.64 to 230.0.0.1, length 92. Packet 2 is a DNS query from 192.168.1.109 to 8.8.8.8. Packet 3 is a DNS response from 8.8.8.8 to 192.168.1.109. Packet 4 is a TCP packet from 192.168.1.109 to 230.0.0.1, length 100. Packet 5 is a TCP packet from 192.168.1.109 to 230.0.0.1, length 100. Packet 6 is a TCP packet from 192.168.1.109 to 230.0.0.1, length 100. Packet 7 is a TCP packet from 192.168.1.109 to 230.0.0.1, length 100. Packet 8 is a TCP packet from 192.168.1.109 to 230.0.0.1, length 100. Packet 9 is a TCP packet from 192.168.1.109 to 230.0.0.1, length 100. Packet 10 is a TCP packet from 192.168.1.109 to 230.0.0.1, length 100. Packet 11 is a TCP packet from 192.168.1.109 to 230.0.0.1, length 100. Packet 12 is a TCP packet from 192.168.1.109 to 230.0.0.1, length 100. Packet 13 is a TCP packet from 192.168.1.109 to 230.0.0.1, length 100. Packet 14 is a TCP packet from 192.168.1.109 to 230.0.0.1, length 100. Packet 15 is a TCP packet from 192.168.1.109 to 230.0.0.1, length 100. Packet 16 is a TCP packet from 192.168.1.109 to 230.0.0.1, length 100. Packet 17 is a TCP packet from 192.168.1.109 to 230.0.0.1, length 100. Packet 18 is a TCP packet from 192.168.1.109 to 230.0.0.1, length 100. Packet 19 is a TCP packet from 192.168.1.109 to 230.0.0.1, length 100. Packet 20 is a TCP packet from 192.168.1.109 to 230.0.0.1, length 100. Packet 21 is a TCP packet from 192.168.1.109 to 230.0.0.1, length 100.
- Packet Details:** Shows the details of the selected packet (Packet 1). It includes the Ethernet II header, Internet Protocol Version 4 header, and User Datagram Protocol header.
- Packet Bytes:** Shows the raw data of the selected packet in hexadecimal and ASCII format.

Nelle 3 sezioni sopra evidenziate troviamo una sintesi di ogni pacchetto, cioè il suo contenuto con dettagli che contengono altre informazioni visualizzabili cliccando sopra ad ogni riga, ed in basso il contenuto in byte. Con la visuale generale è difficile visualizzare ogni informazione, per questo cliccando su uno specifico pacchetto (ad esempio il secondo che è una richiesta DNS al server per conoscere l'indirizzo del sito www.google.com) si apre una finestra con i dettagli del pacchetto.



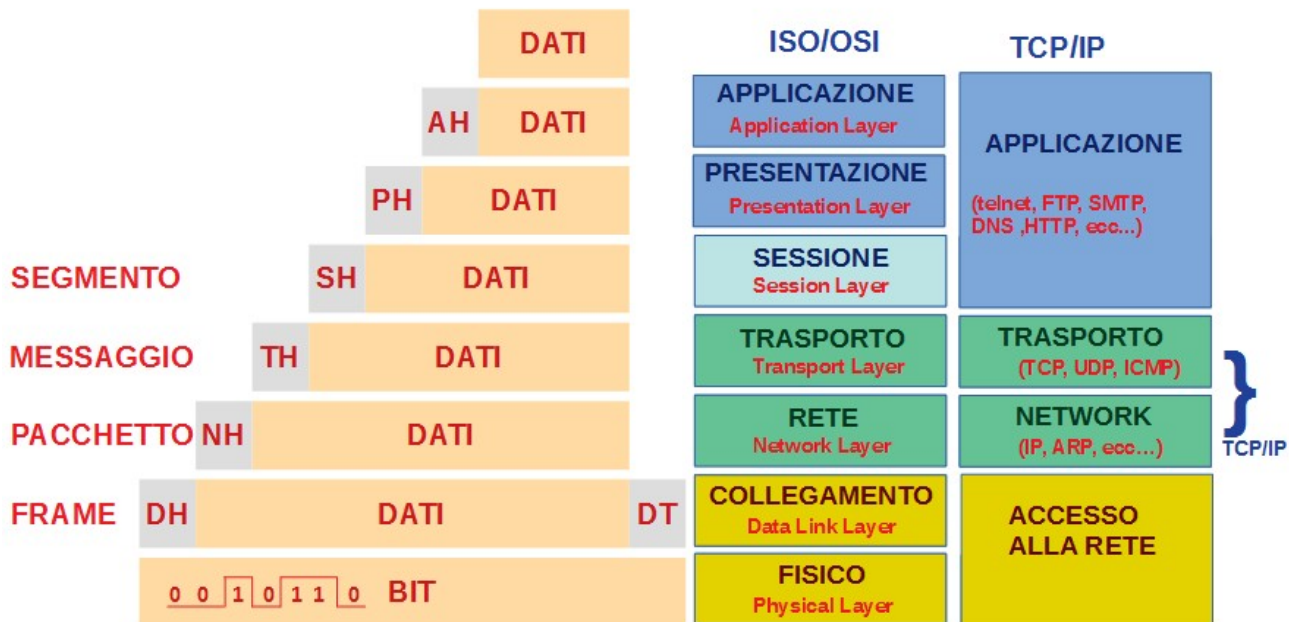
Questo sistema può servire ad esempio per confrontare due pacchetti visualizzandoli contemporaneamente. All'interno della visualizzazione dei dettagli del pacchetto, troviamo una struttura ad albero, ed ogni sezione viene definita **“sottoalbero” (subtree)**. Cliccando sulla freccia a sinistra il sottoalbero si espande o si chiude. Cliccando su un sottoalbero, vengono inoltre **evidenziati i bytes interessati da quella sezione**.



Cominciamo a fare una distinzione con i termini che stiamo utilizzando, se parliamo di **pacchetto**, intendiamo l'insieme di dati in uscita dal livello 3 (*livello Rete*) del modello ISO-OSI, se invece utilizziamo il termine **frame**, intendiamo l'insieme di dati in uscita dal livello 2 (*livello Collegamento*) dello stesso modello.

Non dobbiamo dimenticare infatti che nel modello ISO-OSI viene effettuato un **imbustamento multiplo**, cioè ai pacchetti in entrata ad ogni livello vengono aggiunti dei dati (intestazioni e code dette anche header e tail).

IMBUSTAMENTO MULTIPLO



Chiusa questa parentesi torniamo ad analizzare i dettagli del pacchetto, analizziamo ad esempio il contenuto della prima riga.

```

Frame 2 74 bytes on wire (592 bits) 74 bytes captured (592 bits) on interface 0
  Interface id: 0 (\Device\NPF_{DDD1ECB4-2437-4DD8-9CBC-7CA43630C432})
    Encapsulation type: Ethernet (1)
    Arrival Time: Feb  5, 2019 17:24:11.912562000 ora solare Europa occidentale
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1549383851.912562000 seconds
    [Time delta from previous captured frame: 0.379524000 seconds]
    [Time delta from previous displayed frame: 0.379524000 seconds]
    [Time since reference or first frame: 0.379524000 seconds]
    Frame Number: 2
    Frame Length: 74 bytes (592 bits)
    Capture Length: 74 bytes (592 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:udp:dns]
    [Coloring Rule Name: UDP]
    [Coloring Rule String: udp]
  > Ethernet II, Src: LiteonTe_54:54:49 (d0:53:49:54:54:49), Dst: Routerbo_aa:bc:32 (64:d1:54:aa:bc:32)
  > Internet Protocol Version 4, Src: 192.168.1.109, Dst: 8.8.8.8
  > User Datagram Protocol, Src Port: 56510, Dst Port: 53
  > Domain Name System (query)

```

Possiamo notare oltre al numero del pacchetto ed all'interfaccia di acquisizione, la quantità di dati sulla linea e la quantità di dati catturati. In questo caso coincidono, ma mettendo dei filtri può accadere che non vengano memorizzati tutti i dati che passano sulla linea (questo se si vuole utilizzare meno spazio sul PC).

Nel pannello **PACKET BYTES**, in basso troviamo il contenuto in bytes del pacchetto catturato.

0000	64 d1 54 aa bc 32 d0 53 49 54 54 49 08 00 45 00	d·T··2·S ITTI··E·
0010	00 3c 13 34 00 00 80 11 55 58 c0 a8 01 6d 08 08	·<·4···· UX···m··
0020	08 08 dc be 00 35 00 28 71 b5 50 28 01 00 00 01	·····5·(q·P(····
0030	00 00 00 00 00 00 03 77 77 77 06 67 6f 6f 67 6c	·······w ww·googl
0040	65 03 63 6f 6d 00 00 01 00 01	e·com····

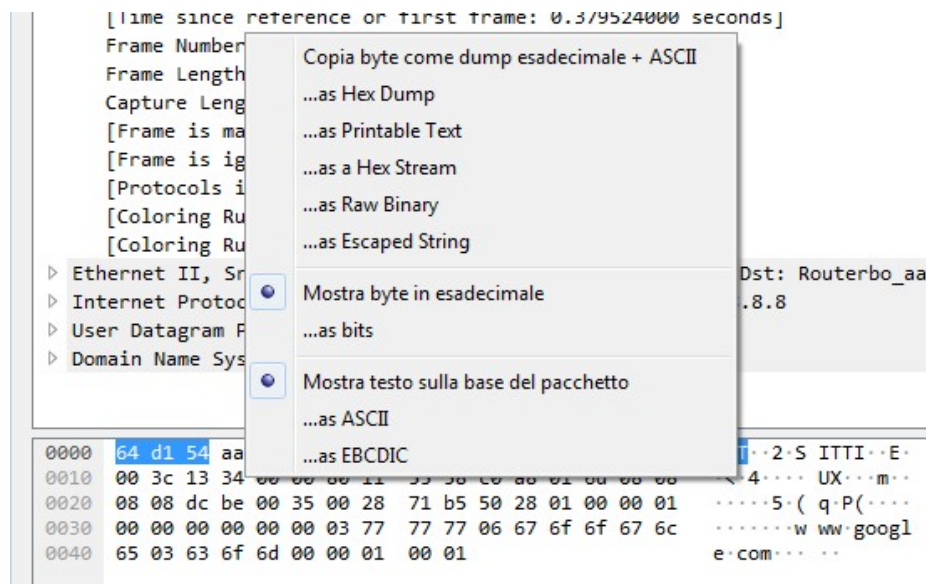
OFFSET BYTES **CONTENUTO IN HEX** **DATI IN ASCII**

La prima colonna è un indice in esadecimale che rappresenta l'indirizzo di ogni byte. La prima riga inizia da 0000 e ci sono in tutto 16 bytes, pertanto la seconda riga inizierà in 10 (esadecimale).

Nella seconda colonna troviamo il contenuto in byte del pacchetto dati.

Nella terza colonna troviamo lo stesso contenuto, visualizzato però in caratteri ASCII, se un valore esadecimale, non corrisponde ad alcun carattere ASCII, viene visualizzato un punto “.”

Un clic con il pulsante destro sul pannello, apre una finestra consente di visualizzare il contenuto in bytes o in bit, e la terza colonna in ASCII (**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange, *codice standard americano per lo scambio di informazioni*) o in EBCDIC (**E**xtended **B**inary **C**oded **D**ecimal **I**nterchange **C**ode, *sistema di codifica ad 8 bit utilizzato dalla IBM*).



Come detto prima, evidenziando una riga nel **PACKET DETAILS**, si evidenziano bytes corrispondenti nel **PACKET BYTES**.

Caratteristica quest'ultima molto utile per analizzare eventuali problemi nella **dissezione**.

Per dissezione si intende l'operazione che Wireshark effettua (*mediante i **dissettori**, propri strumenti software*) per effettuare l'analisi sintattica di un protocollo al fine di decodificarlo e presentarlo nell'interfaccia.

I dissettori sono la parte fondamentale di Wireshark.

FILTRI

Altra parte importante di Wireshark, sono i filtri. Mediante questo strumento riusciamo a ridurre la quantità di dati catturati, concentrando l'attenzione su ciò che è di nostro interesse.

Su Wireshark ci sono due tipi di filtri, i filtri di **cattura** (*capture filters*) e quelli di **visualizzazione** (*display filters*).

I primi agiscono sulla cattura dei pacchetti trascurando i pacchetti che non soddisfano i criteri stabiliti nel filtro.

I secondi agiscono solo sulla visualizzazione, pertanto i pacchetti vengono catturati ma visualizzati secondo i criteri indicati.

Altra differenza tra i due, è la sintassi utilizzata, nel caso dei filtri di cattura si utilizza una sintassi di basso livello denominata **Berkeley Packet Filter BPF**, mentre invece i filtri di visualizzazione utilizzano una sintassi logica più comune ed utilizzata anche da altri programmi.

FILTRI DI CATTURA

Ricordiamo che l'utilizzo dei filtri di cattura riduce di molto la quantità di pacchetti catturati e memorizzati, e ciò influisce anche su ciò che possiamo visualizzare tramite i filtri di visualizzazione (*se un pacchetto non è stato catturato non può essere ovviamente visualizzato*).

Lasciando vuoto il campo dei filtri di cattura Wireshark acquisirà tutto, opzione questa che a volte può creare problemi causati dalla quantità di dati (*vedi ad esempio l'utilizzo di Wireshark in remoto*).

Gli elementi che costituiscono un filtro di cattura sono; **protocollo, direzione e tipo**.

Ad esempio, ***tcp dst port 22***, indica che si vogliono catturare i pacchetti trasmessi secondo il protocollo TCP, con destinazione la porta 22.

In questo caso il protocollo è il **TCP**, la direzione è **dst** (destinazione) ed il tipo è **porta 22**.

La direzione può essere di tipo **src** o **dst**, il primo src per catturare da uno specifico indirizzo di origine, il secondo dst per catturare un pacchetto indirizzato ad uno specifico indirizzo di destinazione.

Oltre a definire una destinazione, si possono usare dei modificatori come **src or dst** oppure **src and dst**.

I tipi possibili sono: **host, port, net, portrange**. Se il tipo viene omissso si considera il primo host.

Direzione e protocollo possono essere omissi, per cercare un tipo nell'origine o nella destinazione in tutti i protocolli. Per esempio ***dst host 192.168.0.100*** mostrerebbe solo il traffico che va a questo indirizzo IP.

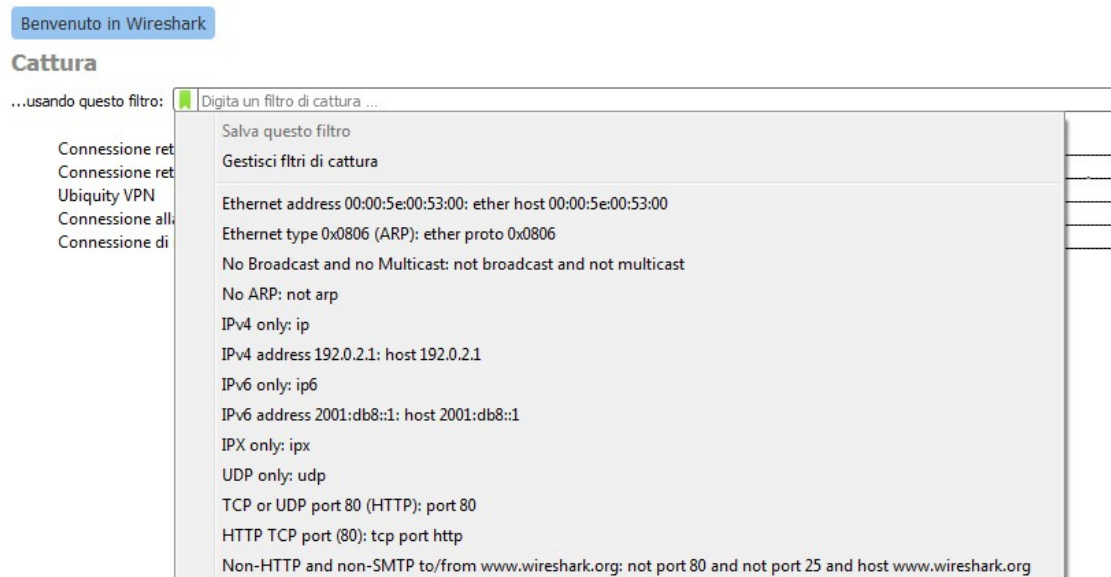
Se si omette l'elemento dst verrebbe mostrato il traffico in ingresso ed in uscita dall'indirizzo IP indicato.

Il programma ci impedisce di inserire dei filtri con la sintassi sbagliata, pertanto riusciamo ad accorgerci subito di eventuali errori.

Di seguito una serie di esempi, tratti dal manuale ufficiale di wireshark <https://wiki.wireshark.org/>

FILTRO	SIGNIFICATO
host 172.18.5.4	Cattura il traffico in entrata ed uscita dall'indirizzo 172.18.5.4
net 192.168.0.0/24 oppure net 192.168.0.0 mask 255.255.255.0	Cattura tutto il traffico della rete indicata, cioè tutto il traffico da e per gli indirizzi compresi tra 192.168.0.1 a 192.168.0.255
src net 192.168.0.0/24 oppure src net 192.168.0.0 mask 255.255.255.0	Cattura il traffico proveniente dalla rete indicata, cioè tutto il traffico proveniente dagli indirizzi compresi tra 192.168.0.1 a 192.168.0.255
dst net 192.168.0.0/24 oppure dst net 192.168.0.0 mask 255.255.255.0	Cattura il traffico destinato alla rete indicata, cioè tutto il traffico diretto agli indirizzi compresi tra 192.168.0.1 a 192.168.0.255
port 53	Cattura tutto il traffico indirizzato alla porta 53
Host www.example.com and not (port 80 or port 25) oppure host www.example.com and not port 80 and not port 25	Cattura tutto il traffico diretto all'indirizzo www.example.com escluso quello SMTP (porta 25) e HTTP (porta 80)
port not 53 and not arp	Cattura tutti i pacchetti indirizzati alla porta 53 escludendo quelli ARP
(tcp[0:2] > 1500 and tcp[0:2] < 1550) or (tcp[2:2] > 1500 and tcp[2:2] < 1550) oppure tcp portrange 1501-1549	Cattura i pacchetti TCP diretti a due gruppi di porte quelle superiori a 1500 e quelle inferiori a 1550
ip	Cattura solo il traffico IPV4
ether	Cattura i protocolli ethernet
tcp	Cattura i protocolli TCP
ip6	Cattura i protocolli IPV6
arp	Cattura i protocolli ARP
Tcp port 80	Cattura i protocolli TCP diretti alla porta 80
ether broadcast	Cattura il traffico broadcast (indirizzato a tutti i dispositivi della rete) su rete ethernet

Cliccando comunque sulla bandierina verde accanto alla barra dei filtri di catture, Wireshark ci mostra dei filtri già pronti e ci consente di gestire i filtri modificandoli o aggiungendone altri.



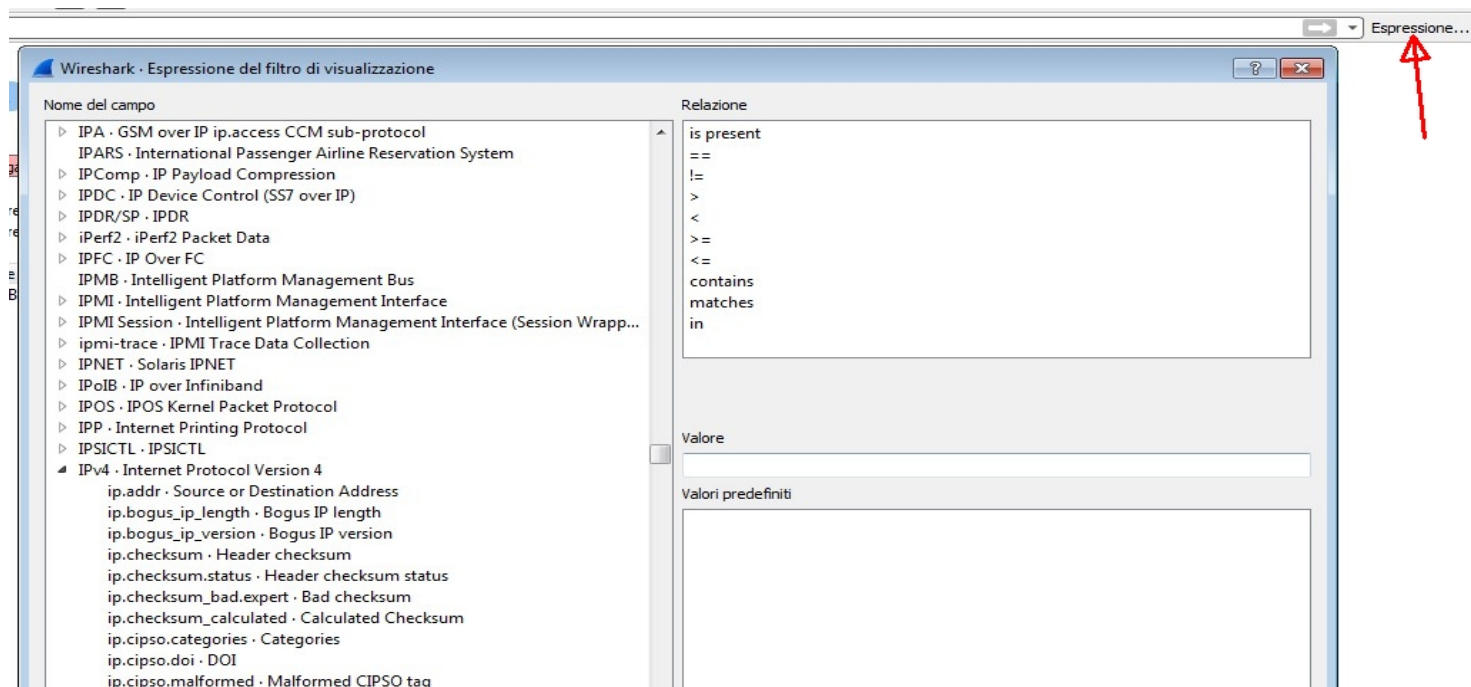
FILTRI DI VISUALIZZAZIONE

La sintassi del filtro di visualizzazione si basa su espressioni che restituiscono i valori TRUE o FALSE.

Queste espressioni utilizzano operatori di confronti che si possono combinare con operatori BOOLEANI, per esprimere criteri complessivi ed ottenere i risultati desiderati.

Operatore	In linguaggio C	Descrizione
eq	==	Uguale a
ne	!=	Diverso da
gt	>	Maggiore di
lt	<	Minore di
ge	>=	Maggiore o uguale di
le	<=	Minore o uguale di
Contains		Verifica se il campo contiene un dato valore
Matches		Esamina un campo sulla base di un'espressione regolare in stile PERL
and	&&	Restituisce TRUE se entrambi le espressioni sono vere
or		Restituisce TRUE se una o entrambe le espressioni è vera
xor	^^	Restituisce TRUE solo se una delle due espressioni è vera
not	!	Nega l'espressione seguente
	[]	Accede ad un sottostringa della stringa, es. dns.resp.name[1..4] accede i primi 4 caratteri del nome di risposta DNS
	()	Raggruppa le espressioni

Wireshark ci offre un grande aiuto per comporre i filtri di visualizzazione, cliccando sulla scritta "espressione" in alto a destra vicino la riga di comando del filtro di visualizzazione, si apre una finestra con una miriade di filtri possibili e personalizzabili. Ogni tipo di filtro ha una struttura ad albero e cliccando sulle frecce si aprono altre sottotipologie.

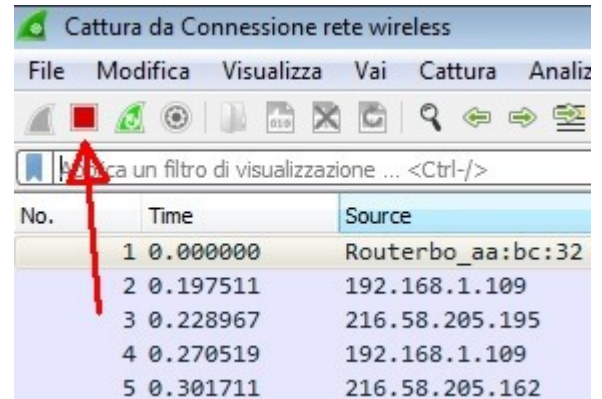


CATTURA DI PACCHETTI DALLA RETE LOCALE

Proviamo ora ad effettuare una semplice cattura di pacchetti dalla rete locale, per poi analizzare quanto acquisito. Dopo aver lanciato Wireshark ed aver selezionato l'interfaccia di rete attiva, l'acquisizione ha inizio. Per avere più pacchetti da analizzare, digitiamo nel browser l'indirizzo di un sito ad esempio www.ebay.it, ed avviamo un breve navigazione tra le sue pagine.

Non essendoci filtri attivi, con questa semplice operazione verranno catturati diversi pacchetti dati.

Conviene pertanto bloccare la cattura dei pacchetti con l'apposito pulsante di STOP.

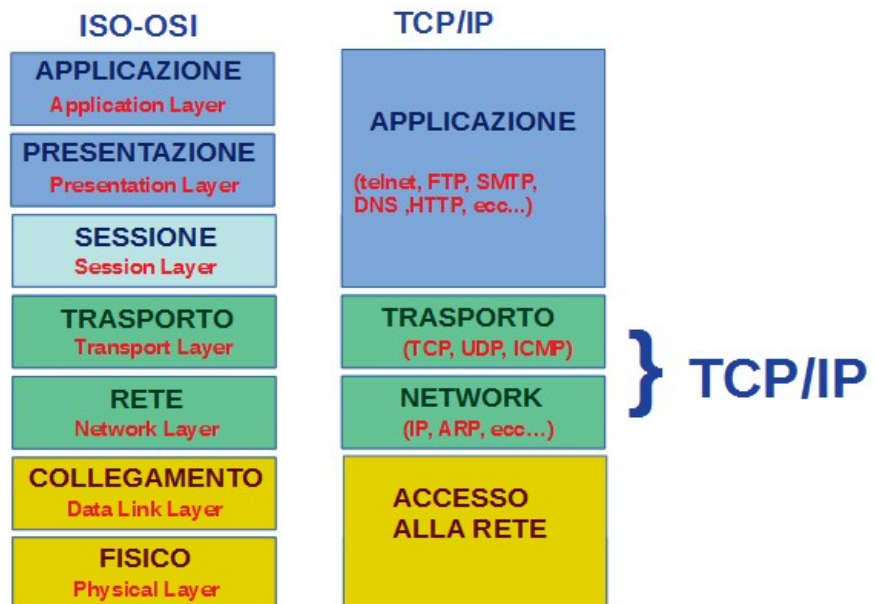


Una delle caratteristiche più interessanti di Wireshark, sta proprio nel funzionamento dei dissettori, e cioè quegli strumenti software in grado di decodificare un pacchetto dati e presentarlo nell'interfaccia.

Nel fare questa operazione infatti, i pacchetti verranno mostrati secondo l'ordine del modello TCP/IP.

Internet infatti si basa su questo protocollo che è differente dallo standard ISO/OSI per il numero di livelli e per la distribuzione dei vari protocolli in ogni strato.

In questa immagine possiamo distinguere i 4 livelli del modello TCP/IP, con qualche esempio di protocollo appartenente ad ogni livello.



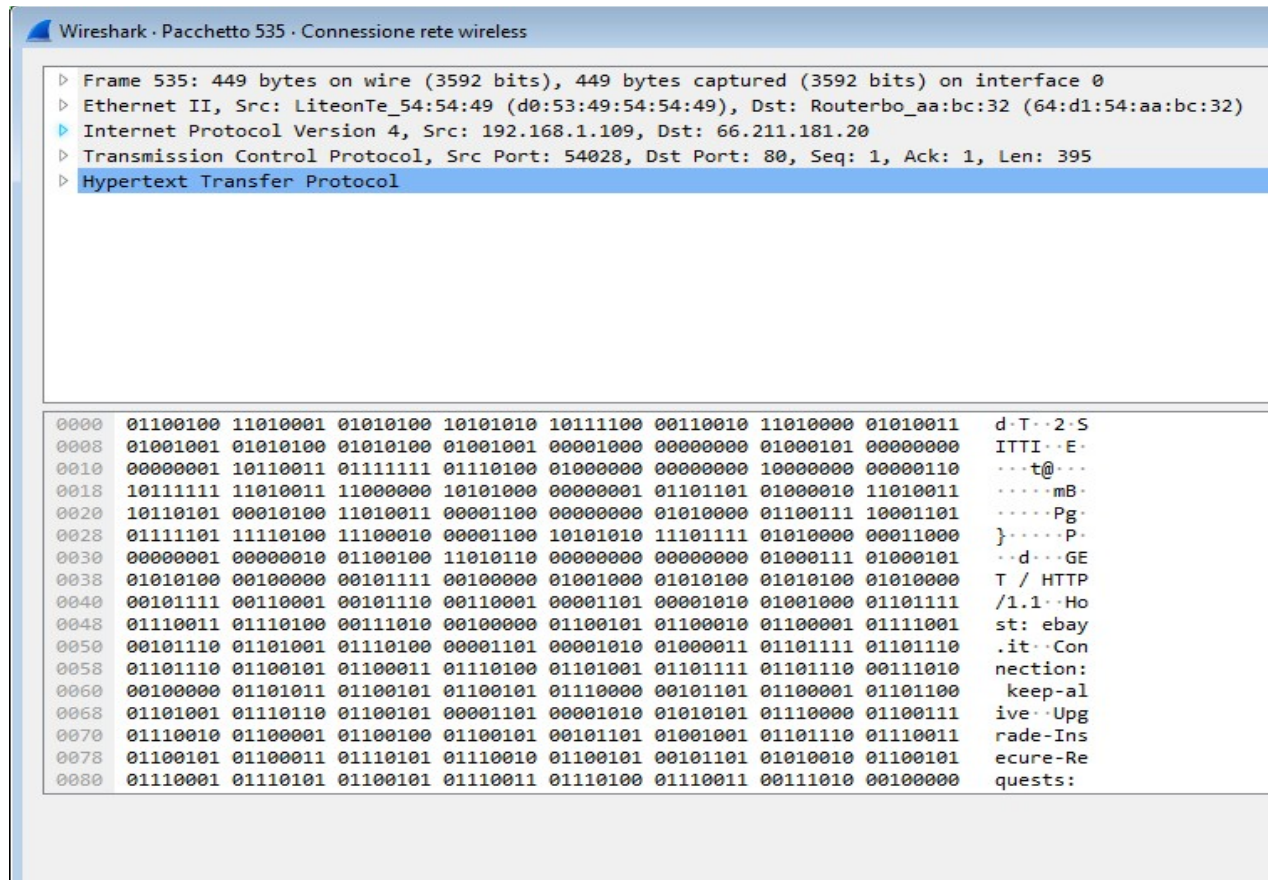
Ora per selezionare qualche pacchetto proviamo a scrivere nel campo dei filtri di visualizzazione "http", in questo modo verranno visualizzati solo i pacchetti che utilizzano questo protocollo, con un po' di fortuna potremmo aprire uno dei pacchetti indirizzati o provenienti dal sito selezionato prima.

Ma come dicevamo prima sulla finestra del dettaglio del pacchetto (cliccando su un pacchetto si apre questa finestra) abbiamo la visualizzazione dei dati secondo il modello TCP/IP della figura precedente.

Perciò oltre al primo livello rappresentato dal frame completo, avremo in ordine gli altri 4 livelli.

Nella figura seguente vediamo il dettaglio di un pacchetto http, si possono notare le seguenti righe:

- **Frame 535** (il 535 esimo pacchetto catturato) come si legge nella riga, esso è composto da 449 bytes.
- **Ethernet II**, dove troviamo l'indirizzo MAC del sorgente (Src) e del destinatario (Dst).
- **Internet Protocol IP**, dove invece troviamo gli indirizzi IP del sorgente (Src) e del destinatario (Dst).
- **Trasmission Control Protocol TCP**, dove troviamo anche le porte del PC utilizzate.
- **HyperText Transfer Protocol HTTP**, che invece contiene i dati a livello applicazione.



In pratica partendo dall'alto dopo il frame abbiamo:

- Livello Accesso alla rete** - **Ethernet II,**
- Livello Network** - **Internet Protocol IP**
- Livello Trasporto** - **Trasmission Control Protocol TCP**
- Livello Applicazione** - **HyperText Transfer Protocol HTTP**



Possiamo ovviamente scendere maggiormente nel dettaglio di ogni livello, andando ad aprire la struttura ad albero cliccando sopra alla riga scelta. In questo modo potremo vedere il contenuto ad esempio del livello Ethernet che verrà anche evidenziato nel pannello dei bytes in basso, dove possiamo vedere il contenuto in binario o in esadecimale.

```

▶ Frame 535: 449 bytes on wire (3592 bits), 449 bytes captured (3592 bits) on interface 0
▶ Ethernet II, Src: LiteonTe_54:54:49 (d0:53:49:54:54:49), Dst: Routerbo_aa:bc:32 (64:d1:54:aa:bc:32)
  ▶ Destination: Routerbo_aa:bc:32 (64:d1:54:aa:bc:32)
  ▶ Source: LiteonTe_54:54:49 (d0:53:49:54:54:49)
  Type: IPv4 (0x0800)
▶ Internet Protocol Version 4, Src: 192.168.1.109, Dst: 66.211.181.20
▶ Transmission Control Protocol, Src Port: 54028, Dst Port: 80, Seq: 1, Ack: 1, Len: 395
▶ Hypertext Transfer Protocol

```


0000	01100100 11010001 01010100 10101010 10111100 00110010 11010000 01010011	d·T·2·S
0008	01001001 01010100 01010100 01001001 00001000 00000000 01000101 00000000	ITTI·E·
0010	00000001 10110011 01111111 01110100 01000000 00000000 10000000 00000110	...t@...

Cliccando ad esempio sulla destinazione, o sulla sorgente, nel pannello dei bytes si evidenzieranno i dati, che corrispondono all'indirizzo in eadecimale scritto sopra.

```

▶ Frame 535: 449 bytes on wire (3592 bits), 449 bytes captured (3592 bits) on interface 0
▶ Ethernet II, Src: LiteonTe_54:54:49 (d0:53:49:54:54:49), Dst: Routerbo_aa:bc:32 (64:d1:54:aa:bc:32)
  ▶ Destination: Routerbo_aa:bc:32 (64:d1:54:aa:bc:32)
  ▶ Source: LiteonTe_54:54:49 (d0:53:49:54:54:49)
  Type: IPv4 (0x0800)
▶ Internet Protocol Version 4, Src: 192.168.1.109, Dst: 66.211.181.20
▶ Transmission Control Protocol, Src Port: 54028, Dst Port: 80, Seq: 1, Ack: 1, Len: 395
▶ Hypertext Transfer Protocol

```


0000	01100100 11010001 01010100 10101010 10111100 00110010 11010000 01010011	d·T·2·S
0008	01001001 01010100 01010100 01001001 00001000 00000000 01000101 00000000	ITTI·E·
0010	00000001 10110011 01111111 01110100 01000000 00000000 10000000 00000110	...t@...

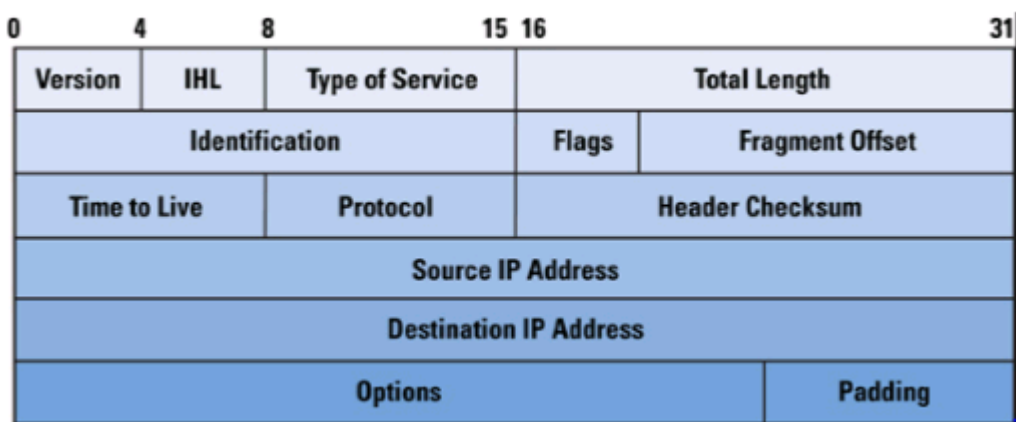
```

▶ Frame 535: 449 bytes on wire (3592 bits), 449 bytes captured (3592 bits) on interface 0
▶ Ethernet II, Src: LiteonTe_54:54:49 (d0:53:49:54:54:49), Dst: Routerbo_aa:bc:32 (64:d1:54:aa:bc:32)
  ▶ Destination: Routerbo_aa:bc:32 (64:d1:54:aa:bc:32)
  ▶ Source: LiteonTe_54:54:49 (d0:53:49:54:54:49)
  Type: IPv4 (0x0800)
▶ Internet Protocol Version 4, Src: 192.168.1.109, Dst: 66.211.181.20
▶ Transmission Control Protocol, Src Port: 54028, Dst Port: 80, Seq: 1, Ack: 1, Len: 395
▶ Hypertext Transfer Protocol

```


0000	01100100 11010001 01010100 10101010 10111100 00110010 11010000 01010011	d·T·2·S
0008	01001001 01010100 01010100 01001001 00001000 00000000 01000101 00000000	ITTI·E·
0010	00000001 10110011 01111111 01110100 01000000 00000000 10000000 00000110	...t@...

Allo stesso modo possiamo analizzare il datagramma IP, dove troviamo la struttura nota di questo pacchetto dati, composta dai seguenti campi:



Wireshark · Pacchetto 535 · Connessione rete wireless

▶ Frame 535: 449 bytes on wire (3592 bits), 449 bytes captured (3592 bits) on interface 0

▶ Ethernet II, Src: LiteonTe_54:54:49 (d0:53:49:54:54:49), Dst: Routerbo_aa:bc:32 (64:d1:54:aa:bc:32)

▶ Internet Protocol Version 4, Src: 192.168.1.109, Dst: 66.211.181.20

0100 = Version: 4

.... 0101 = Header Length: 20 bytes (5)

Differenziated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)

0000 00.. = Differentiated Services Codepoint: Default (0)

.... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)

Total Length: 435

Identification: 0x7f74 (32628)

Flags: 0x4000, Don't fragment

0... .. = Reserved bit: Not set

.1.. .. = Don't fragment: Set

..0. = More fragments: Not set

...0 0000 0000 0000 = Fragment offset: 0

Time to live: 128

Protocol: TCP (6)

Header checksum: 0xbfd3 [validation disabled]

[Header checksum status: Unverified]

Source: 192.168.1.109

Destination: 66.211.181.20

TOTAL LENGTH

IDENTIFICATION

FLAGS

FRAGMENT OFFSET

TIME TO LIVE

PROTOCOL

HEADER CHECKSUM

SOURCE IP ADDRESS

DESTINATION IP ADDRESS

▶ Transmission Control Protocol, Src Port: 54028, Dst Port: 80, Seq: 1, Ack: 1, Len: 395

▶ Hypertext Transfer Protocol

0008 01001001 01010100 01010100 01001001 00001000 00000000 01000101 00000000 ITTI...E-

0010 00000001 10110011 01111111 01110100 01000000 00000000 10000000 00000110 ...t@...

0018 10111111 11010011 11000000 10101000 00000001 01101101 01000010 11010011mB-

0020 10110101 00010100 11010011 00001100 00000000 01010000 01100111 10001101 ...Pg-

0028 01111101 11110100 11100010 00001100 10101010 11101111 01010000 00011000 }.....P-

0030 00000001 00000010 01100100 11010110 00000000 00000000 01000111 01000101 ..d...GE

Allo stesso modo possiamo aprire la parte relativa al TCP ed all’HTTP per analizzare tutti i singoli segmenti di dati che costituiscono il frame complessivo.

Nel Pannello dei bytes in basso, troviamo sempre evidenziati i dati corripodenti a ciò che evidenziamo in alto. Possiamo notare, cliccando su ogni strato, il modo di “imbustare” i dati classico del modello ISO/OSI e del TCP/IP.

Un altro pacchetto dati semplice da analizzare, è quello del protocollo ARP, digitiamo sul filtro di visualizzazione “arp”, e clicchiamo su uno dei pacchetti.

In questo caso abbiamo solo 2 livelli, Ethernet (Livello di accesso alla rete) e ARP (Livello Network)

ARP si occupa di associare ad ogni indirizzo MAC dei dispositivi, il corrispondente indirizzo IP.

Gli indirizzi sono evidenziati nella seguente immagine.

```

> Frame 223: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
- Ethernet II, Src: Routerbo_aa:bc:32 (64:d1:54:aa:bc:32), Dst: LiteonTe_54:54:49 (d0:53:49:54:54:49)
  > Destination: LiteonTe_54:54:49 (d0:53:49:54:54:49)
  > Source: Routerbo_aa:bc:32 (64:d1:54:aa:bc:32)
    Type: ARP (0x0806)
    Padding: 00000000000000000000000000000000
  - Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: Routerbo_aa:bc:32 (64:d1:54:aa:bc:32)
    Sender IP address: 192.168.1.251
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.1.109

0000  11010000 01010011 01001001 01010100 01010100 01001001 01100100 11010001  ·SITTI·
0008  01010100 10101010 10111100 00110010 00001000 00000110 00000000 00000001  T··2···
0010  00001000 00000000 00000110 00000100 00000000 00000001 01100100 11010001  ·····d·
0018  01010100 10101010 10111100 00110010 11000000 10101000 00000001 11111011  T··2···
0020  00000000 00000000 00000000 00000000 00000000 00000000 11000000 10101000  ······
0028  00000001 01101101 00000000 00000000 00000000 00000000 00000000 00000000  ·m·····
0030  00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000  ······
0038  00000000 00000000 00000000 00000000  ····

```

Un'altro pacchetto facilmente analizzabile è quello relativo al protocollo DNS (Domain Name System).
In questo caso possiamo notare nel contenuto del pacchetto DNS a livello applicazione, il nome del sito di wikipedia e il relativo indirizzo 91.168.174.192

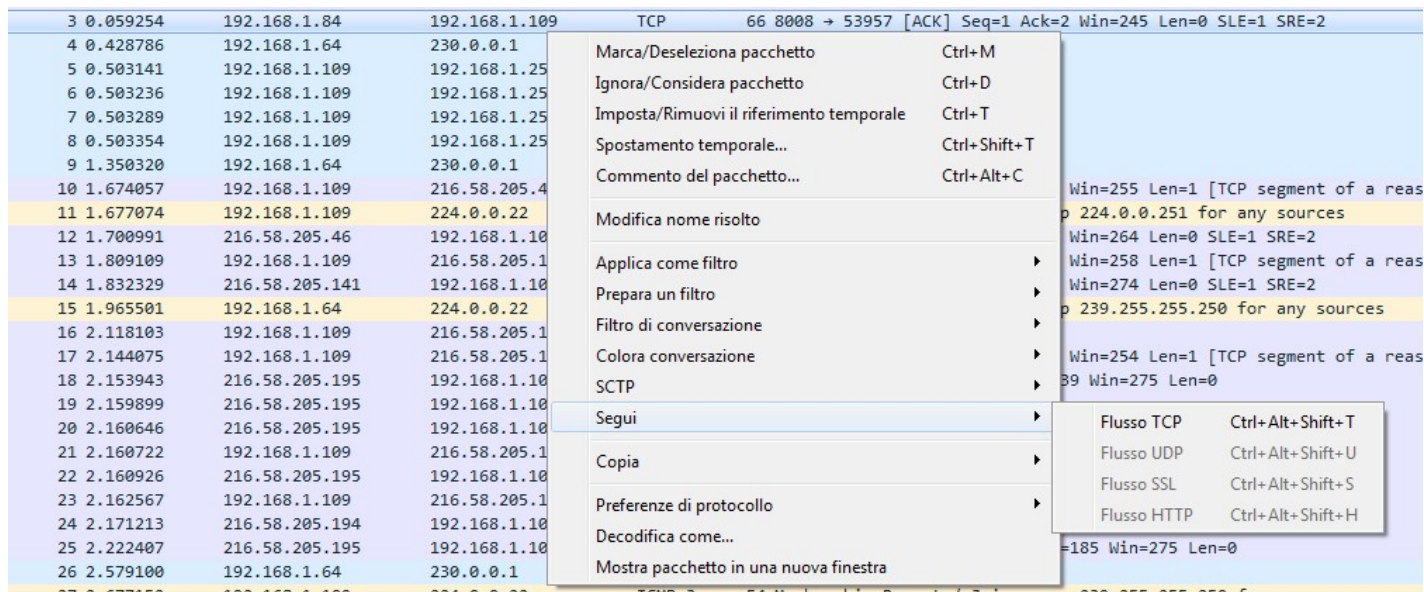
```

> Frame 2291: 94 bytes on wire (752 bits), 94 bytes captured (752 bits) on interface 0
- Ethernet II, Src: Motorola_41:8f:e7 (d0:f8:8c:41:8f:e7), Dst: LiteonTe_54:54:49 (d0:53:49:54:54:49)
- Internet Protocol Version 4, Src: 8.8.8.8, Dst: 192.168.43.24
- User Datagram Protocol, Src Port: 53, Dst Port: 50034
- Domain Name System (response)
  Transaction ID: 0x11e2
  > Flags: 0x8180 Standard query response, No error
  Questions: 1
  Answer RRs: 1
  Authority RRs: 0
  Additional RRs: 0
  - Queries
    - meta.wikimedia.org: type A, class IN
      Name: meta.wikimedia.org
      [Name Length: 18]
      [Label Count: 3]
      Type: A (Host Address) (1)
      Class: IN (0x0001)
  - Answers
    - meta.wikimedia.org: type A, class IN, addr 91.198.174.192
      [Request ID: 2256]
      [Time: 0.051110000 seconds]

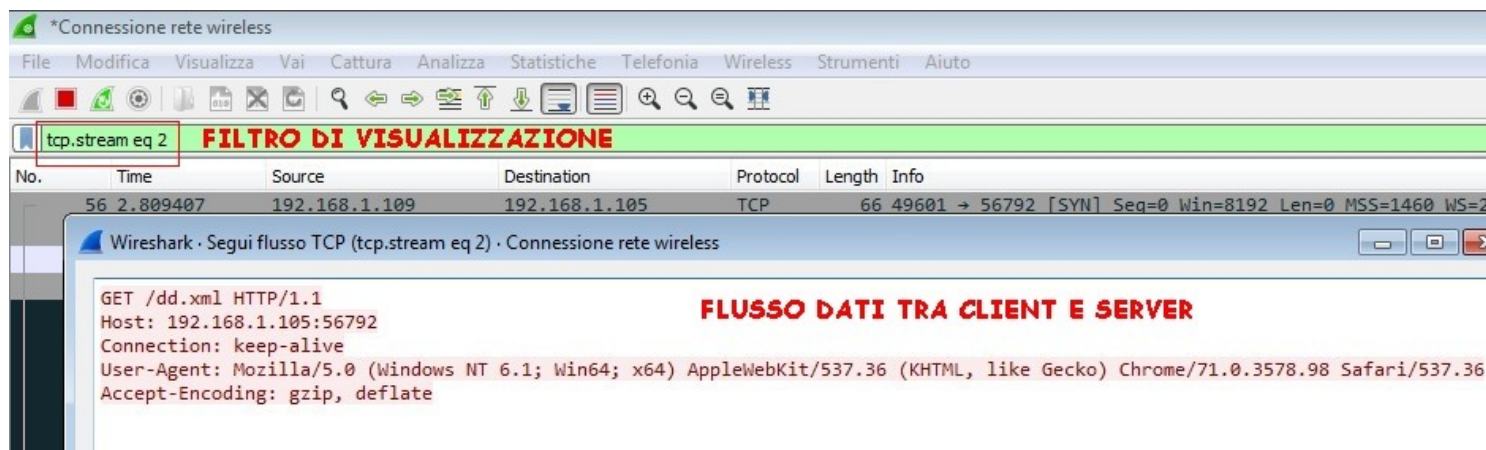
0000  11010000 01010011 01001001 01010100 01010100 01001001 11010000 11111000  ·SITTI·
0008  10001100 01000001 10001111 11100111 00001000 00000000 01000101 00000000  ·A····E·
0010  00000000 01010000 11101111 01101110 00000000 00000000 01110011 00010001  ·P····s·
0018  01011100 01011110 00001000 00001000 00001000 00001000 11000000 10101000  \^·····
0020  00101011 00011000 00000000 00110101 11000011 01110010 00000000 00111100  +··5·r<
0028  00101000 10100100 00010001 11100010 10000001 10000000 00000000 00000001  (·····
0030  00000000 00000001 00000000 00000000 00000000 00000000 00000100 01101101  ······m
0038  01100101 01110100 01100001 00001001 01110111 01101001 01101011 01101001  eta·wiki
0040  01101101 01100101 01100100 01101001 01100001 00000011 01101111 01110010  media·or
0048  01100111 00000000 00000000 00000001 00000000 00000001 11000000 00001100  g·····
0050  00000000 00000001 00000000 00000001 00000000 00000000 00000010 01010111  ······k
0058  00000000 00000100 01011011 11000110 10101110 11000000  ······

```

Cliccando inoltre su un qualsiasi pacchetto dati con il tasto destro del mouse si può seguire il flusso dati tra client e server



Seguendo un flusso dati automaticamente viene creato il relativo filtro di visualizzazione.



Anche senza seguire il flusso dati con il pulsante destro si può cliccare sopra un pacchetto dati ed impostare un filtro di visualizzazione.



Quanto detto finora è solo una breve introduzione all'utilizzo di un software molto potente per l'analisi del flusso dati di una rete.

Ovviamente l'utilizzo di Wireshark non è utile solo per affrontare l'analisi dei pacchetti di rete da un punto di vista didattico, ma anche e soprattutto per studiare le tecniche di attacco e di difesa di una rete al fine di migliorarne la sicurezza.

Per quest'importante utilizzo dobbiamo però mettere in campo altre conoscenze ed altri strumenti che ci consentiranno di simulare ogni situazione e verificare tutte le criticità che si possono avere.

Per questo possiamo anticipare senza entrare nei dettagli, la necessità di installare un potente sistema operativo indispensabile per testare e mettere a dura prova la sicurezza di una rete e cioè Kali Linux.

Questo sistema operativo può essere installato su una macchina ma come primo approccio potrebbe essere utile installarlo in una macchina virtuale (VM) con Virtual Box.

In questo modo avremo il sistema operativo Kali Linux dentro al nostro PC, che potrà essere avviato insieme al sistema operativo già avviato, senza alcun rischio.

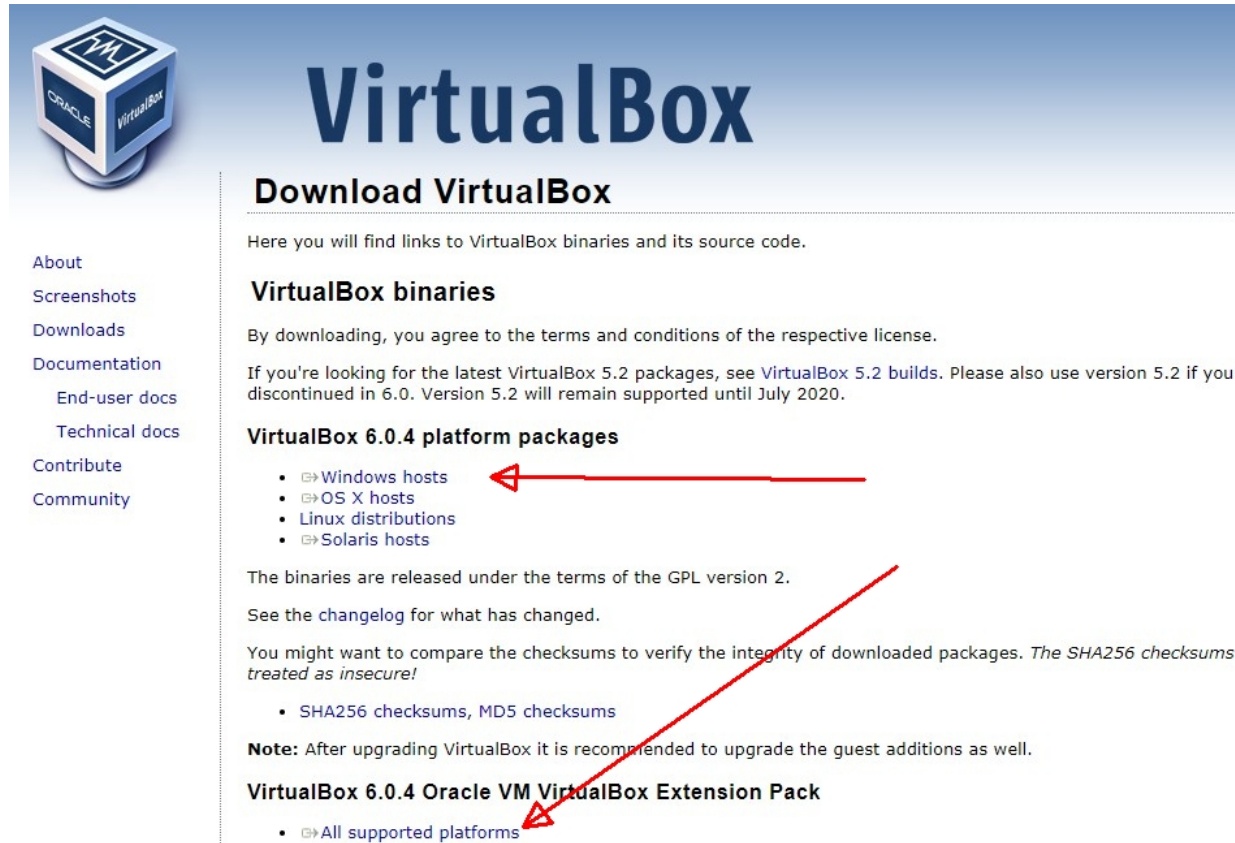
Di seguito le istruzioni per l'installazione di Virtual Box e di Kali Linux.

VIRTUAL BOX

La prima operazione da fare è quella di installare l'ultima versione di Virtual Box, scaricandola per il proprio sistema operativo dal link <https://www.virtualbox.org/wiki/Downloads>

L'installazione è praticamente automatica cliccando sul file scaricato, basta confermare tutte le opzioni richieste.

Dopo aver installato Virtual Box occorre scaricare ed installare (sempre cliccando sul file scaricato) le **Virtual Box Extension Pack**, sempre allo stesso link.

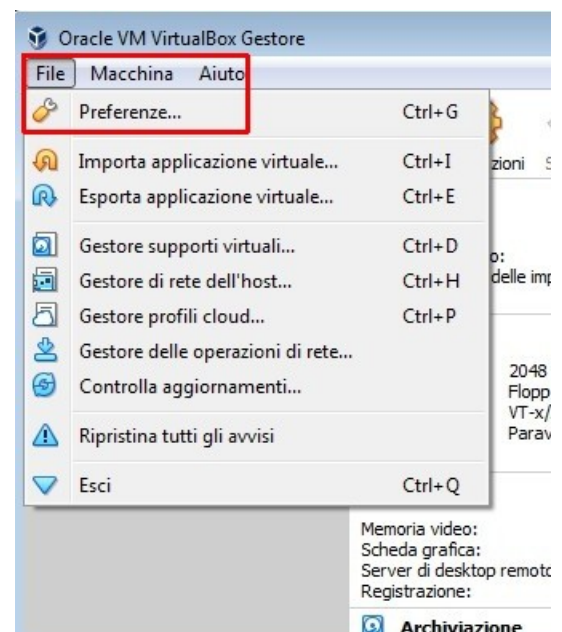
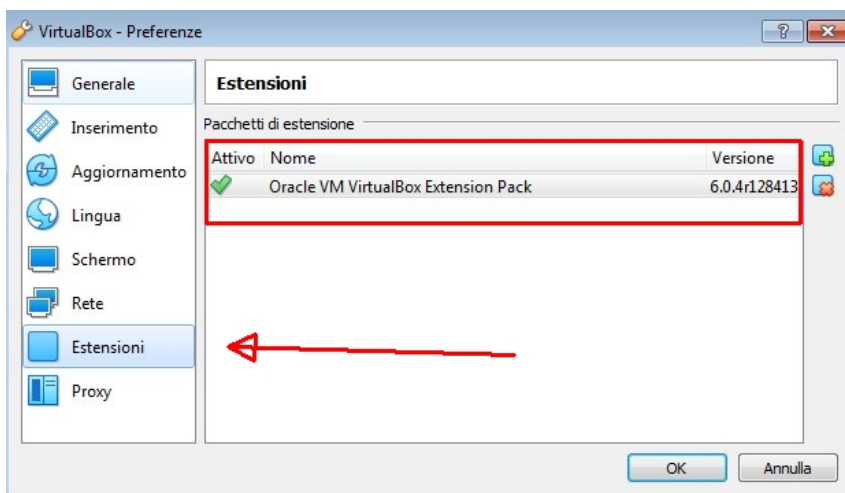


The screenshot shows the VirtualBox website's download page. On the left is a sidebar with links: About, Screenshots, Downloads, Documentation (with sub-links for End-user docs and Technical docs), Contribute, and Community. The main content area is titled 'Download VirtualBox' and contains the following sections:

- VirtualBox binaries**: A section stating that downloading implies agreement to the license terms. It provides a link to 'VirtualBox 5.2 builds' and notes that version 5.2 is supported until July 2020.
- VirtualBox 6.0.4 platform packages**: A list of links for different operating systems: Windows hosts, OS X hosts, Linux distributions, and Solaris hosts. A red arrow points to the 'Windows hosts' link.
- VirtualBox 6.0.4 Oracle VM VirtualBox Extension Pack**: A list of links for different platforms, with a red arrow pointing to the 'All supported platforms' link.

Additional text on the page includes a note about GPL version 2 terms, a changelog link, and a warning about SHA256 checksums being insecure.

Dopo l'installazione lanciando Virtual Box e cliccando su **File-Preferenze**, dovremmo trovare le Etension Pack installate.



KALI LINUX

Seconda cosa da fare è scaricare la ISO del sistema operativo KALI, reperibile al link:

<https://www.kali.org/downloads/>

.Nella sezione downloads, troviamo le varie versioni di KALI Linux, consiglio di installare l'ultima versione completa (non light) a 64 bit.

Il file ISO (Immagine del Sistema Operativo) si può scaricare direttamente cliccando su HTTP, o indirettamente cliccando su Torrent e scaricando così il file torrent da dare in pasto al programma bitTorrent o ad altri programmi del genere che gestiscono questo tipo di condivisione di file.



links to **download Kali Linux** in its latest official release. For a release history, check our Kali Linux Releases page. Please note: You can find unofficial, untested weekly releases at <http://cdimage.kali.org/kali-weekly/>.

Blog Downloads Training Documentati

Image Name	Download	Size	Version	sha256sum
Kali Linux Light 64 Bit	HTTP Torrent	867M	2018.4	ad63589f761a4344e930486e05e9d3652b8c8badb2e0f808951861ed489db1f6
Kali Linux Light Armhf	HTTP Torrent	630M	2018.4	4b409b7f0650741400b2c3c9076333f6c52211205c4a2828d677f1099d3e5d64
Kali Linux Light 32 Bit	HTTP Torrent	863M	2018.4	0659674f841d91b71bd2503e352ded588ec17d0e976c9fee4345dad35ace83b1
Kali Linux 64 Bit	HTTP Torrent	3.0G	2018.4	7c65d6a319448efe4ee1be5b5a93d48ef30687d4e3f507896b46b9c2226a0ed0
Kali Linux 32 Bit	HTTP Torrent	3.1G	2018.4	14e53cd797d673db31437c36d51bab0f0a0b6ef9ab277c6c90b9f1fc9d96c291

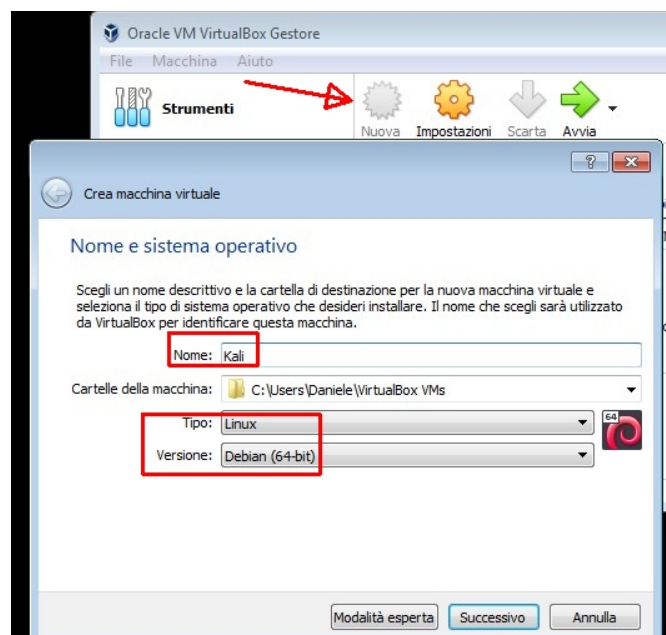
MACCHINA VIRTUALE KALI LINUX

Ora occorre creare la macchina virtual di KALI Linux nell'ambiente di Virtual Box.

Lanciando Virtual Box clicchiamo su nuova ed alla finestra successiva impostiamo il tipo di sistema operativo "Linux" e la versione "Debian 64bit".

Possiamo dare un nome qualsiasi alla macchina virtuale scegliamo magari un nome che ci indica il S.O. e cioè Kali.

Poi proseguiamo con il tasto "successivo".



Successivamente verrà richiesta la RAM da utilizzare, si consigliano almeno 2GB.

Dimensione della memoria

Seleziona la quantità di memoria (RAM) in megabyte che sarà allocata per la macchina virtuale.

La quantità di memoria consigliata è **1024 MB**.



Successivo

Annulla

Verrà chiesto di creare un disco fisso.

Disco fisso

Se lo desideri, puoi aggiungere un disco fisso virtuale alla nuova macchina. Puoi creare un nuovo file di disco fisso, selezionarne uno dall'elenco o da un'altra posizione utilizzando l'icona della cartella.

Se hai bisogno di una configurazione di archiviazione più complessa, puoi saltare questo passaggio e modificare le impostazioni della macchina dopo averla creata.

La dimensione consigliata del disco fisso è **8,00 GB**.

- ☐ Non aggiungere un disco fisso virtuale
- ☒ Crea subito un nuovo disco fisso virtuale
- ☐ Usa un file di disco fisso virtuale esistente

Kali.vdi (Normale, 20,71 GB)

Crea

Annulla

Verrà chiesta la tipologia del disco fisso.

Lasciare l'opzione di default.

Tipo di file del disco fisso

Scegli il tipo del file che desideri utilizzare per il nuovo disco fisso virtuale. Se non hai bisogno di utilizzarlo con altri programmi di virtualizzazione, puoi lasciare inalterata questa opzione.

- ☒ VDI (VirtualBox Disk Image)
- ☐ VHD (Virtual Hard Disk)
- ☐ VMDK (Virtual Machine Disk)

Modalità esperta

Successivo

Annulla

Verrà richiesto se utilizzare l'allocazione dinamica del disco (scelta consigliata). Con l'allocazione dinamica le dimensioni sul disco fisico del PC della macchina virtuale, crescono fino alla dimensione massima che sceglieremo poi.

In caso contrario verrebbe utilizzato uno spazio fisso sul disco rigido delle dimensioni indicate.

Infine scegliamo la dimensione del disco della macchina virtuale, si consigliano almeno 20GB.

Archiviazione su disco fisso fisico

Scegli se il nuovo disco fisso virtuale deve crescere in base all'utilizzo (allocato dinamicamente) o se deve essere creato alla dimensione massima (dimensione specificata).

Un file di disco fisso **allocato dinamicamente** utilizzerà solo lo spazio del disco fisico che si riempie (fino alla massima **dimensione specificata**), ma non si ridurrà automaticamente se lo spazio viene liberato.

Un file di disco fisso a **dimensione specificata** richiede normalmente più tempo per la creazione su alcuni sistemi, ma è spesso più veloce nell'utilizzo.

- ☒ Allocato dinamicamente
- ☐ Dimensione specificata

Successivo

Annulla

Posizione file e dimensione

Digita il nome del nuovo disco fisso virtuale nella casella seguente o fai clic sull'icona della cartella per selezionare una cartella di destinazione diversa.

Kali

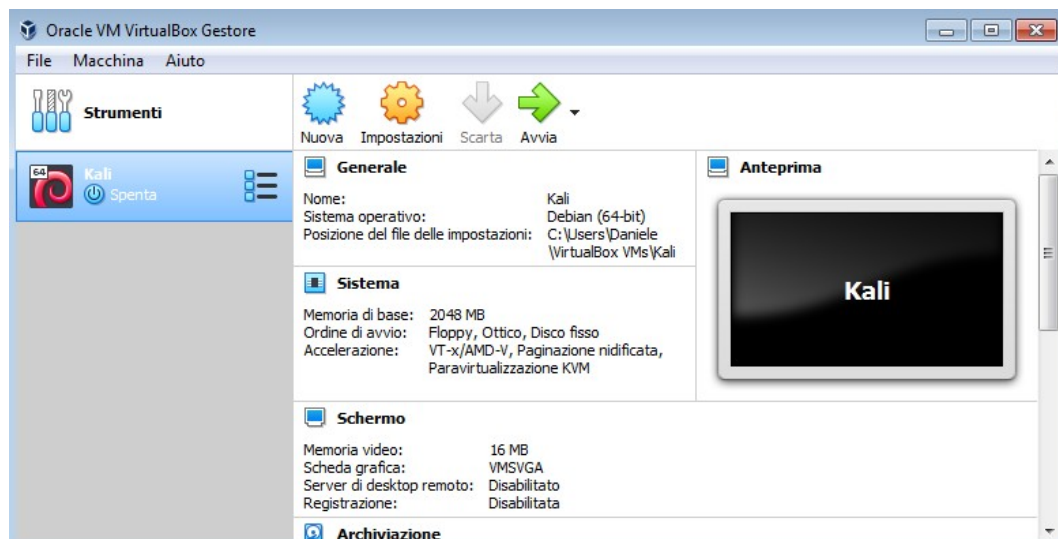
Seleziona la dimensione del disco fisso virtuale in megabyte. Questa dimensione è il limite sul totale dei dati che una macchina virtuale sarà in grado di archiviare su disco.

4,00 MB 20,26 GB 2,00 TB

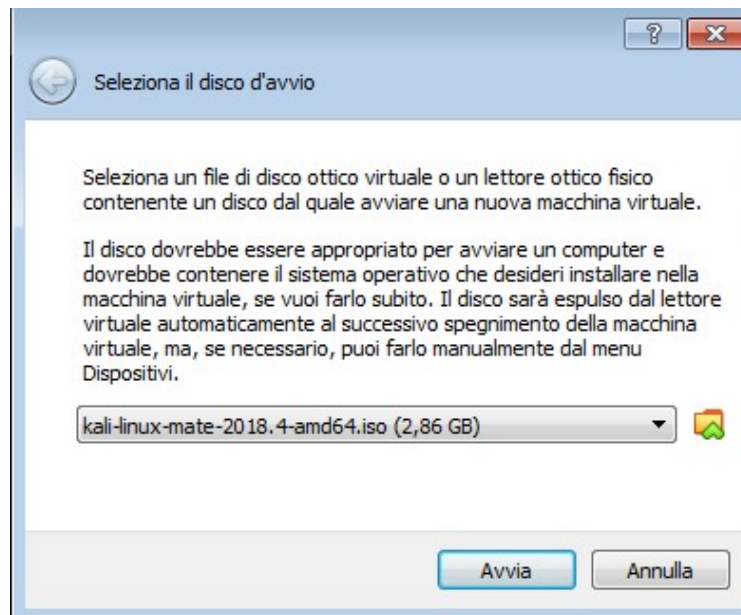
Crea

Annulla

Al termine cliccando sul tasto "Crea" la nostra macchina virtuale è pronta per l'installazione. E' come se avessimo assemblato un PC con tutte le caratteristiche richieste, RAM, disco fisso. La parte successiva sarà l'installazione del sistema operativo dall'immagine precedentemente scaricata.



Cliccando sul tasto AVVIA di virtual box, ci verrò chiesto inizialmente dove trovare l'immagine ISO precedentemente scaricata. Cliccando sul pulsante con la cartella a destra andare a selezionare il file ISO scaricato.

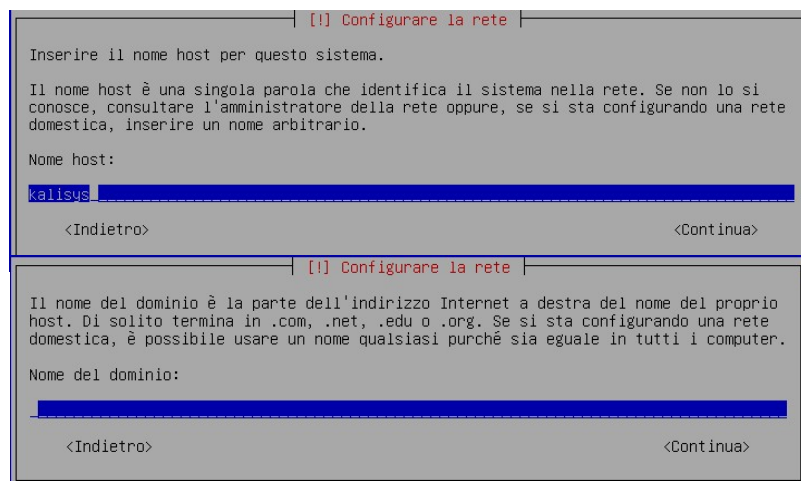


Successivamente si avvierà il classico menù di ogni sistema Linux che prevede la possibilità di una Live, cioè l'avvio del sistema operativo senza installazione. Noi invece selezioneremo subito l'installazione.



Le prime richieste fatte durante l'installazione riguardano la lingua del sistema e della tastiera, impostare ovviamente **l'Italiano**.

Verrà poi chiesto il nome del sistema, possiamo scegliere il nome che vogliamo, in questo caso **kalisys**.



Possiamo lasciar vuoto il nome del dominio.

Verrà poi chiesta la **password** di root con relativa conferma, si consiglia di **salvare la password di root**, in quanto indispensabile in ogni operazione.

Successivamente dovremo partizionare il disco fisso, si consiglia di lasciare le seguenti impostazioni di default.

```
| [!!] Partizionamento dei dischi |  
  
Il programma d'installazione può guidare nel partizionare un disco o, se si preferisce, è  
possibile procedere manualmente. Anche usando la procedura guidata si potranno  
successivamente vedere i risultati e adattarli alle proprie esigenze.  
  
Scegliendo il partizionamento guidato per l'intero disco, sarà chiesto il disco da usare.  
  
Metodo di partizionamento:  
  
Guidato - usa l'intero disco  
Guidato - usa l'intero disco e imposta LVM  
Guidato - usa l'intero disco e imposta LVM cifrato  
Manuale  
  
<Indietro>
```

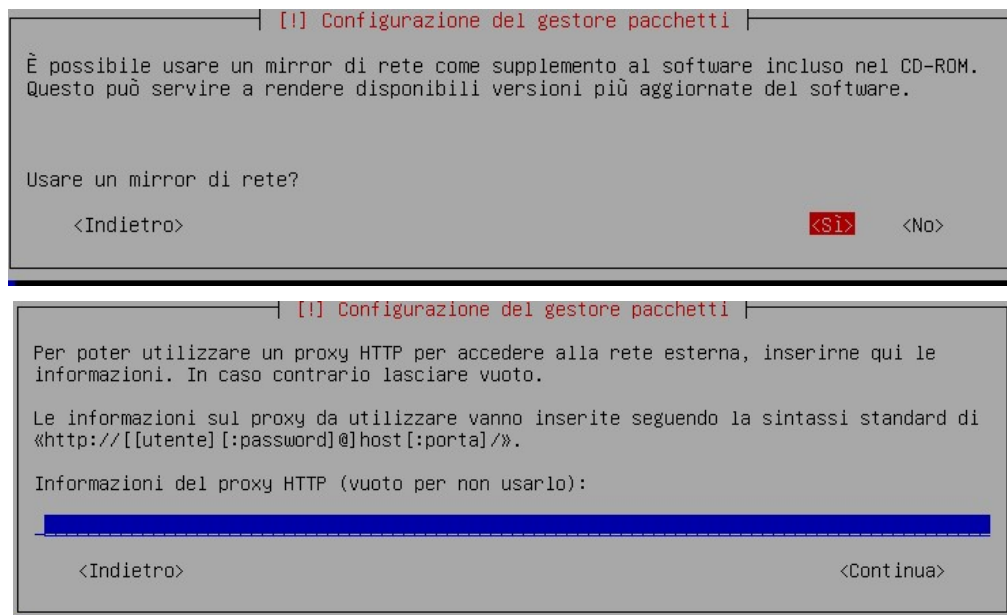
```
| [!!] Partizionamento dei dischi |  
  
Tutti i dati presenti sul disco selezionato saranno eliminati dopo la conferma  
dell'applicazione delle modifiche.  
  
Selezionare il disco da partizionare:  
  
SCSI1 (0,0,0) (sda) - 21.8 GB ATA VBOX HARDDISK  
  
<Indietro>
```

```
| [!] Partizionamento dei dischi |  
  
Selezionato da partizionare:  
  
SCSI1 (0,0,0) (sda) - ATA VBOX HARDDISK: 21.8 GB  
  
Il disco può essere partizionato usando uno dei diversi schemi. Se insicuri scegliere il  
primo.  
  
Schema di partizionamento:  
  
Tutti i file in una partizione (per nuovi utenti)  
Partizione /home separata  
Partizioni /home, /var e /tmp separate  
  
<Indietro>
```

```
| [!!!] Partizionamento dei dischi |  
  
Questa è un'anteprima delle partizioni e dei punti di mount attualmente configurati.  
Selezionare una partizione per modificarne le impostazioni (file system, punto di mount,  
ecc.), uno spazio libero per creare delle partizioni o un dispositivo per inizializzarne  
la tabella delle partizioni.  
  
Partizionamento guidato  
Configurare il RAID software  
Configurare il Logical Volume Manager  
Configurare volumi cifrati  
Configurare volumi iSCSI  
  
SCSI1 (0,0,0) (sda) - 21.8 GB ATA VBOX HARDDISK  
n° 1 primaria 19.6 GB f ext4 /  
n° 5 logica 2.1 GB f swap swap  
  
Annullare le modifiche alle partizioni  
Terminare il partizionamento e scrivere le modifiche sul disco  
  
<Indietro>
```

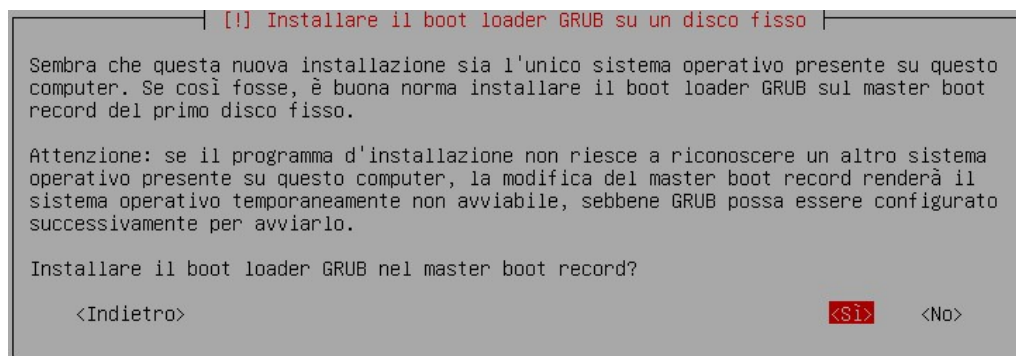
```
| [!!] Partizionamento dei dischi |  
  
Scegliendo di continuare, le modifiche elencate di seguito verranno scritte sui dischi;  
altrimenti è possibile fare ulteriori modifiche manualmente.  
  
Le tabelle delle partizioni dei seguenti dispositivi sono state modificate:  
SCSI1 (0,0,0) (sda)  
  
Le seguenti partizioni stanno per essere formattate:  
partizione n° 1 di SCSI1 (0,0,0) (sda) con ext4  
partizione n° 5 di SCSI1 (0,0,0) (sda) con swap  
  
Scrivere le modifiche sui dischi?  
  
<Si> <No>
```


Successivamente verrà chiesto se usare un mirror di rete (scelta consigliata) ed eventualmente presente, le impostazioni del Server Proxy. Nel nostro caso non essendoci il Proxy il campo va lasciato vuoto.

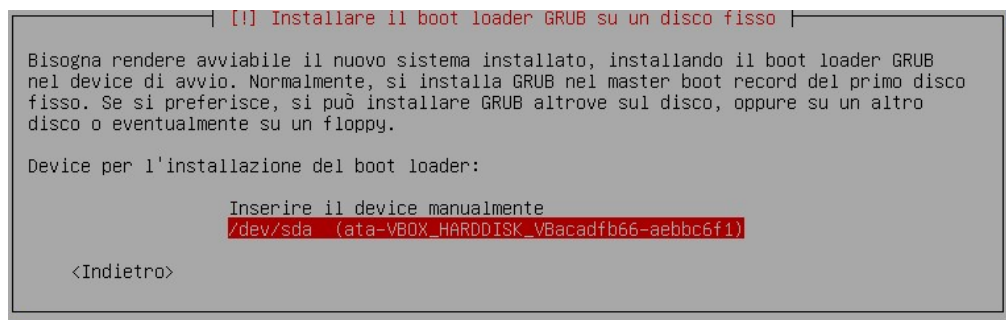


La fase successiva riguarda il GRUB (GNU GRand Unified Bootloader) e cioè quella parte del sistema operativo che si occupa della fase di BOOT (avvio) del sistema stesso. Il GRUB normalmente va a modificare l'MBR (Master Boot Record) cioè quella zona fisica dell'Hard Disk del PC che viene letta all'avvio per far partire il sistema operativo della macchina.

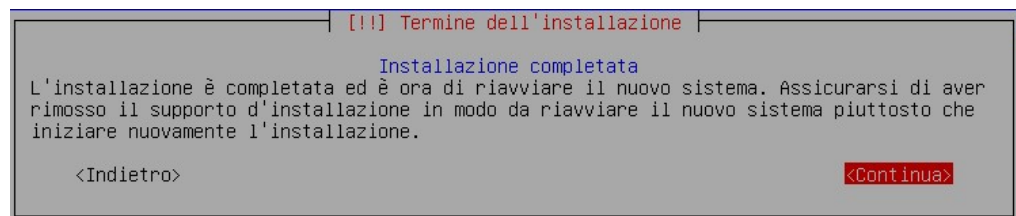
Se installassimo Kali su un PC reale dovremmo stare attenti a questa impostazione, nel caso nostro trattandosi di una macchina virtuale, possiamo eseguire questa operazione a cuor leggero.



Perciò confermiamo le opzioni secondo queste schermate.



L'installazione è ora terminata e la macchina virtuale con Kali è pronta per essere avviata.



Avviando Kali Linux da Virtual Box con il pulsante “AVVIA”, il sistema operativo partirà nella finestra di Virtual Box, ed una volta avviato chiederà nome utente e password.

Il nome utente da inserire è **root** la password è quella scelta in fase di installazione.

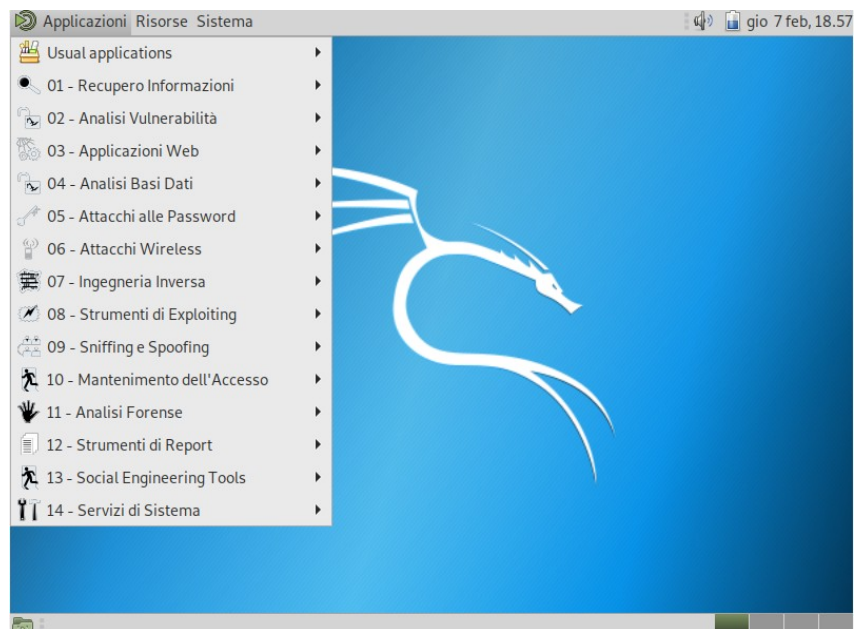
```
Kali GNU/Linux Rolling kalisys tty1
kalisys login: root
```

Successivamente ci troveremo
il terminale con il prompt
dei comandi, a questo

```
Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@kalisys:~# startx_
```

punto possiamo decidere se far partire l'interfaccia grafica con il comando “**startx**”

Consiglio di verificare
l'accesso ad internet, tramite
l'avvio del browser **firefox**
presente nelle applicazioni
“**usual application**”
e “**internet**”.



e accedendo al terminale
tramite “**usual application**”
“**strumenti di sistema**”
e “**terminale di mate**”,
dobbiamo digitare
apt-get update, che accede ad internet
per controllare la presenza delle
liste dei pacchetti presenti
nelle repository del sistema operativo,
indicate nel file **/etc/apt/sources.list**.

```
root@Kalisys:
File Modifica Visualizza Cerca Terminale Aiuto
root@Kalisys:~# apt-get update
Scaricamento di:1 http://kali.download/ka
Scaricamento di:2 http://kali.download/ka
[17,1 MB]
Scaricamento di:3 http://kali.download/ka
ges [188 kB]
Recuperati 17,3 MB in 25s (680 kB/s)
Lettura elenco dei pacchetti... Fatto
root@Kalisys:~#
```

In caso non ci fosse l'accesso ad internet, occorre
configurare correttamente la rete dell'macchina
virtuale accedendo alle impostazioni, nel nostro caso
abbiamo questa situazione.

Ovviamente ogni caso andrà valutato singolarmente
e con un po' di conoscenza delle basi ed un po' di
intuito si riuscirà a rendere funzionante la
connessione ad internet.

Rete

Scheda 1 Scheda 2 Scheda 3 Scheda 4

☒ Abilita scheda di rete

Connessa a: Scheda con bridge

Nome: Qualcomm Atheros AR956x Wireless Network Adapter

Avanzate

Tipo di scheda: Intel PRO/1000 MT Desktop (82540EM)

Modalità promiscua: Nega

Indirizzo MAC: 080027618555

☒ Cavo connesso

CONCLUSIONI

Possiamo definire quanto detto finora una premessa di quello che sarà il vero e complesso lavoro dell'analisi di una rete. Seguendo questa guida, possiamo però muovere i primi passi e conoscere anche superficialmente i complessi strumenti descritti prima.

In seguito affronteremo nel dettaglio determinate situazioni che ci consentiranno maggiormente di comprendere le varie problematiche delle reti, utilizzando proprio gli strumenti precedentemente descritti.