

Un po' di pratica

Gaia Maselli

Strumenti di rete

- Netstat
- host
- nslookup
- Wireshark

netstat

- Utile per conoscere lo stato corrente della rete (network status)
- Sessioni TCP attive
 - netstat -t
- Listener TCP attivi
 - netstat -ltn
- Tabella di routing
 - netstat -r
 - netstat -rn
- Interfacce
 - netstat -i

host

- DNS lookup utility
- Converte nomi in indirizzi IP
 - `host <name>`
- Converte indirizzi IP in nomi
 - `host <IP address>`
- Trova DNS server responsabili per un dominio
 - `host -t NS <dominio>`
- Trova mailserver per un dominio
 - `host -t MX <dominio>`

nslookup <host>

nslookup is a program to query Internet domain name servers

Il comando senza opzioni permette di richiedere record di tipo A al server DNS locale

```
> nslookup www.google.com
```

Risposta

```
Server:          151.100.17.15
```

```
Address: 151.100.17.15#53
```

```
Non-authoritative answer:
```

```
Name: www.google.com
```

```
Address: 173.194.35.144
```

```
Name: www.google.com
```

```
Address: 173.194.35.145
```

```
Name: www.google.com
```

```
Address: 173.194.35.146
```

```
Name: www.google.com
```

```
Address: 173.194.35.147
```

```
Name: www.google.com
```

```
Address: 173.194.35.148
```

nslookup -type=NS <dominio>

Permette di richiedere un record di tipo NS al server DNS locale

```
> nslookup -type=NS google.com
```

```
Server:          151.100.17.15
```

```
Address: 151.100.17.15#53
```

Risposta proveniente
dalla cache



Non-authoritative answer:

```
google.com      nameserver = ns4.google.com.
```

```
google.com      nameserver = ns1.google.com.
```

```
google.com      nameserver = ns2.google.com.
```

```
google.com      nameserver = ns3.google.com.
```

Authoritative answers can be found from:

```
ns1.google.com  internet address = 216.239.32.10
```

```
ns2.google.com  internet address = 216.239.34.10
```

```
ns3.google.com  internet address = 216.239.36.10
```

```
ns4.google.com  internet address = 216.239.38.10
```

Packet sniffing

Wireshark

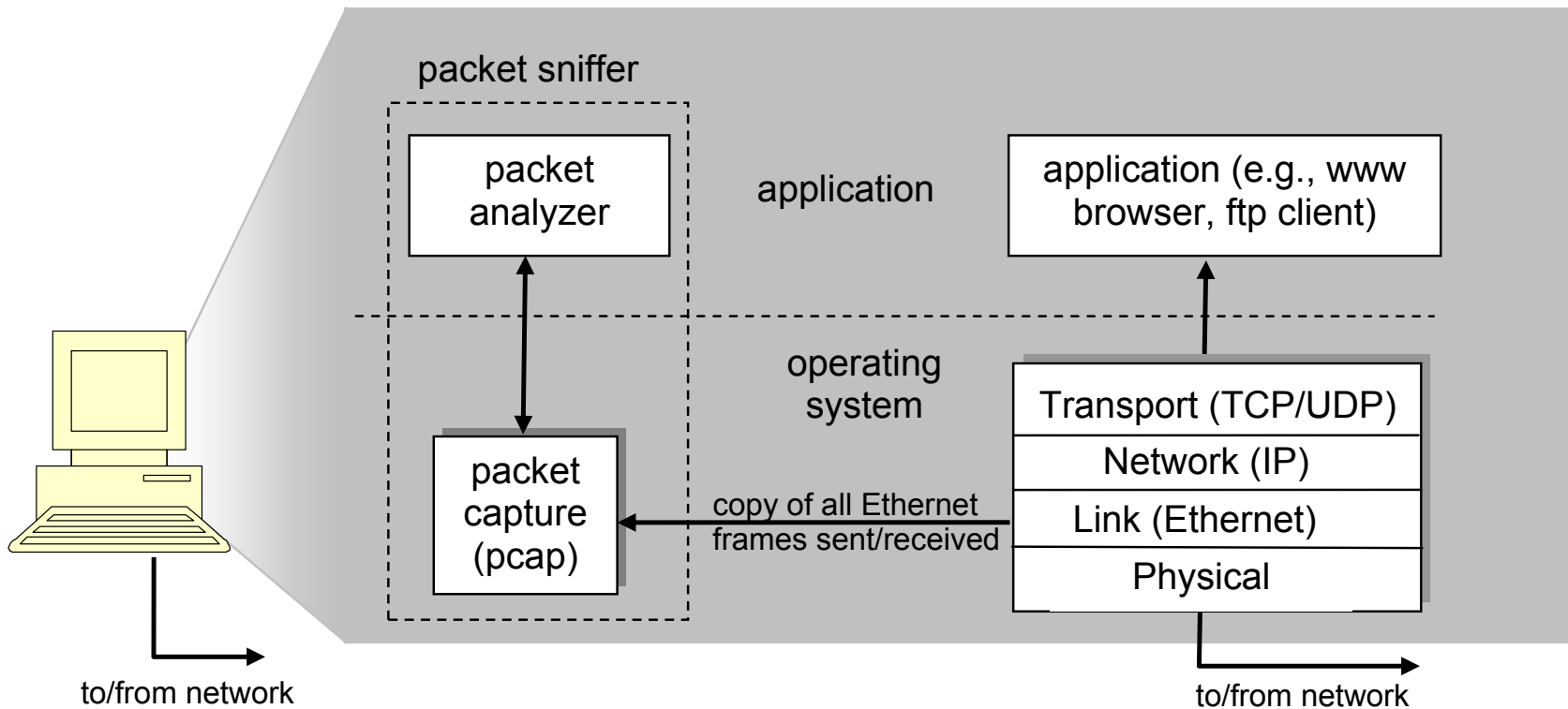
Packet sniffer: Wireshark

- Programma che cattura e analizza tutti i frame entranti e uscenti da una macchina
- Analizzatore passivo di pacchetti
 - Cattura una copia dei **frame** senza modificarli
 - Non immette traffico in rete
- Permette di analizzare i pacchetti a qualsiasi livello dello stack protocollare
 - Estraendo gli header dei vari livelli
- Permette di capire cosa succede all'interno di una rete
 - Utile per didattica
 - Utile agli amministratori di rete
 - Non è un sistema di sicurezza (ma alcuni sistemi di sicurezza si basano su sniffer di pacchetti)
- Deve essere attivo contemporaneamente all'applicazione che vogliamo monitorare
- Comuni packet sniffer: ethereal, tcpdump

Wireshark

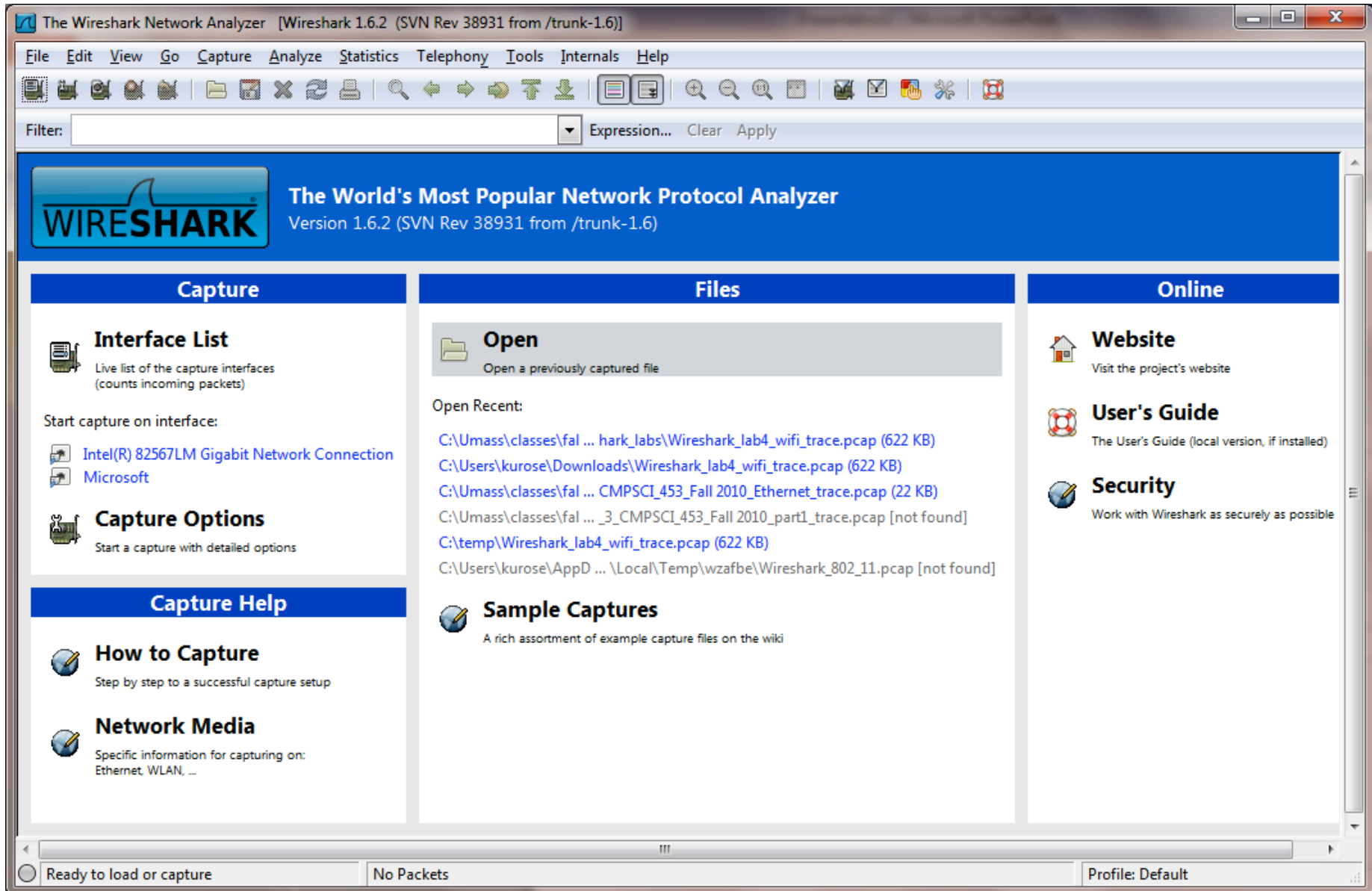
- Tutta la documentazione è disponibile sul sito www.wireshark.org
- Download
 - <http://www.wireshark.org/download.html>
- Per problemi si può consultare
 - <http://wiki.wireshark.org/CaptureSetup>
- Funziona su varie tecnologie di trasmissione: Ethernet, seriale (PPP e SLIP), 802.11 wireless LANs, e altre (se il sistema operativo supporta Wireshark).

Wireshark: architecture



pcap: Packet capture library

Finestra iniziale di Wireshark



Interfaccia grafica

command
menus

display filter
specification

listing of
captured
packets

details of
selected
packet
header

packet content
in hexadecimal
and ASCII

Filter: Expression... Clear Apply

No. -	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.1.46	128.121.50.122	TCP	1163 > http [SYN] Seq=0 Len=0 MSS=1460
2	0.127987	128.121.50.122	192.168.1.46	TCP	http > 1163 [SYN, ACK] Seq=0 Ack=1 win=57
3	0.128232	192.168.1.46	128.121.50.122	TCP	1163 > http [ACK] Seq=1 Ack=1 win=65535
4	0.153700	192.168.1.46	128.121.50.122	HTTP	GET /news/ HTTP/1.1
5	0.329641	128.121.50.122	192.168.1.46	TCP	[TCP segment of a reassembled PDU]
6	0.330326	128.121.50.122	192.168.1.46	HTTP	[TCP Previous segment lost] Continuation
7	0.330467	192.168.1.46	128.121.50.122	TCP	1163 > http [ACK] Seq=657 Ack=1082 win=64
8	0.342042	128.121.50.122	192.168.1.46	TCP	[TCP Retransmission] [TCP segment of a re

Frame 4 (710 bytes on wire, 710 bytes captured)

- Ethernet II, Src: Netgear_61:8e:6d (00:09:5b:61:8e:6d), Dst: WestellT_9f:92:b9 (00:0f:db:9f:92:b9)
- Internet Protocol, Src: 192.168.1.46 (192.168.1.46), Dst: 128.121.50.122 (128.121.50.122)
- Transmission Control Protocol, Src Port: 1163 (1163), Dst Port: http (80), Seq: 1, Ack: 1, Len: 656
- Hypertext Transfer Protocol
 - GET /news/ HTTP/1.1\r\n
 - Host: www.wireshark.org\r\n
 - User-Agent: Mozilla/5.0 (Windows; U; windows NT 5.1; en-US; rv:1.8.1.4) Gecko/20070515 Firefox/2.0.0.4\r\n
 - Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5\r\n
 - Accept-Language: en-us,en;q=0.5\r\n
 - Accept-Encoding: gzip,deflate\r\n
 - Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
 - Keep-Alive: 300\r\n
 - Connection: keep-alive\r\n
 - Referer: http://www.wireshark.org/faq.html\r\n
 - Cookie: __utma=87653150.62471437.1181007382.1181007382.1181169142.2; __utmz=87653150.1181007382.1.1.utm

0000 00 0f db 9f 92 b9 00 09 5b 61 8e 6d 08 00 45 00 [a.m..E.
0010 02 b8 0f 25 40 00 80 06 74 51 c0 a8 01 2e 80 79 ...%0... tQ.....y
0020 32 7a 04 8b 00 50 ed bc 8e 1b 4e c6 f1 18 50 18 2z...P...N...P.
0030 ff ff 77 74 00 00 47 45 54 20 2f 6e 65 77 73 2f ..wt..GE T /news/
0040 20 48 54 54 50 2f 31 2e 31 0d 0a 48 6f 73 74 3a HTTP/1. 1..Host:
0050 20 77 77 77 2e 77 69 72 65 73 68 61 72 6b 2e 6f www.wir eshark.o
0060 72 67 0d 0a 55 73 65 72 2d 41 67 65 6e 74 3a 20 rg..User -Agent:
0070 4d 6f 7a 69 6c 6c 61 2f 35 2e 30 20 28 57 69 6e Mozilla/ 5.0 (win
0080 64 6f 77 73 3b 20 55 3b 20 57 69 6e 64 6f 77 73 dows; U; windows
0090 20 4e 54 20 35 2e 31 3b 20 65 6e 2d 55 53 3b 20 NT 5.1; en-US;
00a0 72 76 3a 31 2e 38 2e 31 2e 34 29 20 47 65 63 6b rv:1.8.1. 4) Geck
00b0 6f 2f 32 30 30 37 30 35 31 35 20 46 69 72 65 66 o/200705 15 Firef

File: "C:\DOCUME~1\PAULAW~1\LOCALS~1\Temp\etherXXXa00324" 453 KB 00:00:... P: 671 D: 671 M: 0 Drops: 0

Menu

- The **command menus** are standard pulldown menus located at the top of the window. Of interest to us now are the File and Capture menus. The File menu allows you to save captured packet data or open a file containing previously captured packet data, and exit the Wireshark application. The Capture menu allows you to begin packet capture.

Packet listing

The **packet-listing window** displays one-line summary for each packet captured, including the packet number (assigned by Wireshark; this is *not* a packet number contained in any protocol's header), the time at which the packet was captured, the packet's source and destination addresses, the protocol type, and protocol-specific information contained in the packet. The packet listing can be sorted according to any of these categories by clicking on a column name. The protocol type field lists the highest-level protocol that sent or received this packet, i.e., the protocol that is the source or ultimate sink for this packet.

packet-header details window

The **packet-header details window** provides details about the packet selected (highlighted) in the packet-listing window. (To select a packet in the packet-listing window, place the cursor over the packet's one-line summary in the packet-listing window and click with the left mouse button.). These details include information about the Ethernet frame (assuming the packet was sent/received over an Ethernet interface) and IP datagram that contains this packet. The amount of Ethernet and IP-layer detail displayed can be expanded or minimized by clicking on the plus minus boxes to the left of the Ethernet frame or IP datagram line in the packet details window. If the packet has been carried over TCP or UDP, TCP or UDP details will also be displayed, which can similarly be expanded or minimized. Finally, details about the highest-level protocol that sent or received this packet are also provided.

packet-contents window

The **packet-contents window** displays the entire contents of the captured frame, in both ASCII and hexadecimal format.

Esercizio 1

- Avviare il browser e pulire la cache
- Avviare Wireshark
- Avviare la cattura dei pacchetti
- Tornare sul browser e inserire la URL:

<http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html>

- Dopo aver visualizzato la pagina, interrompere la cattura dei pacchetti di Wireshark
- Osservare la lista di pacchetti scambiati e quali protocolli sono stati utilizzati

Esercizio 1: analisi del traffico

- Filtrare i pacchetti, digitando **http** nel campo *filter*, e cliccando su **Apply**
 - Trovare ed esaminare il messaggio GET
 - Trovare ed esaminare il messaggio di risposta
- Rimuovere il filtro http e filtrare i pacchetti DNS
 - Esaminare DNS query e response
 - Numero di porta sorgente e destinazione?
- Analizzare pacchetti TCP
- Confrontare le richieste e risposte HTTP e DNS con il formato dei messaggi visti a lezione

Esercizio 1 alternativo

- Eseguire esercizio 1 collegandosi al seguente link:

<http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file3.html>

- Osservare variazioni nel traffico

Esercizio 2

- Avviare il browser
- Avviare Wireshark
- Avviare la cattura dei pacchetti
- Tornare sul browser e inserire la URL:
<http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file4.html>
- Dopo aver visualizzato la pagina, interrompere la cattura dei pacchetti di Wireshark

Esercizio 2: analisi del traffico

- Osservare la lista di pacchetti scambiati e quali protocolli sono stati utilizzati
 - DNS
 - HTTP
 - TCP
- Le due immagini vengono scaricate in seriale e parallelo?
- Si usano connessioni persistenti o non persistenti?
- Visitare una qualsiasi altra pagina contenenti immagini e verificare il risultati

Esercizi

- Utilizzare Wireshark per analizzare il traffico durante:
 - il download di una pagina web contenente immagini e testo
 - Una richiesta/risposta DNS effettuata con nslookup
 - L'invio di una email