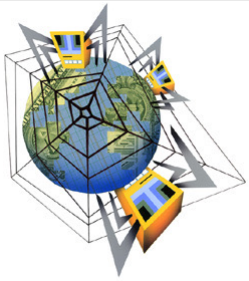
UNICAM
Università di Camerino

Laurea in INFORMATICA



Internet Reti Sicurezza A.A. 2024/2025
Capitolo 2 – APPLICATION PROTOCOLS
Fausto Marcantoni
fausto.marcantoni@unicam.it

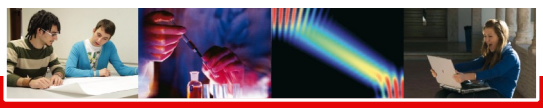
1

Dichiarazione di copyright

L'utilizzo dei contenuti della lezione sono riservati alla fruizione personale degli studenti iscritti ai corsi dell'Università di Camerino. ***Sono vietate*** *la diffusione intera o parziale di video o immagini della lezione, nonché la modifica dei contenuti senza il consenso, espresso per iscritto, del titolare o dei titolari dei diritti d'autore e di immagine.*

Copyright notice

The contents of this lesson are subject to copyright and intended only for personal use by students enrolled in courses offered by the University of Camerino. For this reason, any partial or total reproduction, adaptation, modification and/or transformation of the contents of this lesson, by any means, without the prior written authorization of the copyright owner, is strictly prohibited.



2

Reti di Elaboratori

Livello applicativo

Obiettivi generali:

- Concetti dei protocolli applicativi
- Implementazioni dei protocolli applicativi
- Paradigma client server
- Modelli dei servizi

Obiettivi specifici:

- Protocolli specifici:
  - http
  - ftp
  - smtp
  - pop
  - dns
  - snmp

3

Reti di Elaboratori

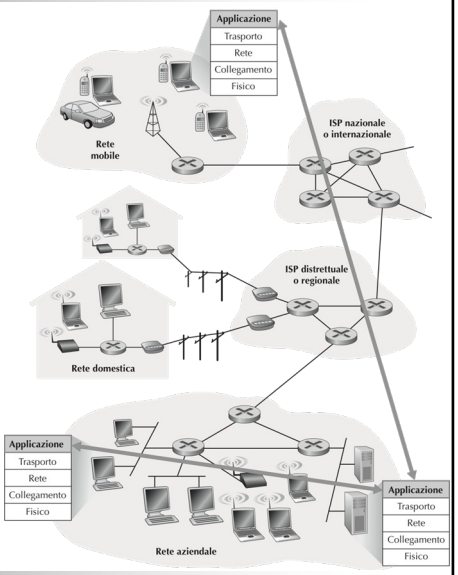
Applicazioni e protocolli applicativi

Applicazione:
processi distribuiti in comunicazione

- In esecuzione su host remoti
- Si scambiano messaggi per eseguire l'applicazione
- Es., mail, FTP, WWW

Protocolli applicativi

- Costituiscono una parte di ogni applicazione
- Definiscono il formato dei messaggi scambiati e il loro significato (azioni)
- Usano i servizi degli strati inferiori



4

Reti di Elaboratori

Applicazioni: terminologia essenziale

- Un processo è un programma in esecuzione su un host
- Sullo stesso host i processi comunicano mediante meccanismi definiti dal SO.
- Processi in esecuzione su host diversi comunicano mediante meccanismi definiti dal protocollo dello strato di applicazione (application layer protocol)
- Processi comunicanti: programmi in esecuzione su due sistemi
- Messaggi: vengono scambiati tra processi (processo mittente e processo destinatario) su due sistemi terminali attraverso la rete
- Architettura client-server: per ciascuna coppia di processi comunicanti se ne etichetta uno come **client** e l'altro come **server**.
- Interfaccia tra il processo e la rete: un processo invia messaggi nella rete e riceve messaggi dalla rete attraverso un'interfaccia software detta socket.

5

Reti di Elaboratori

Processi attivi - Windows

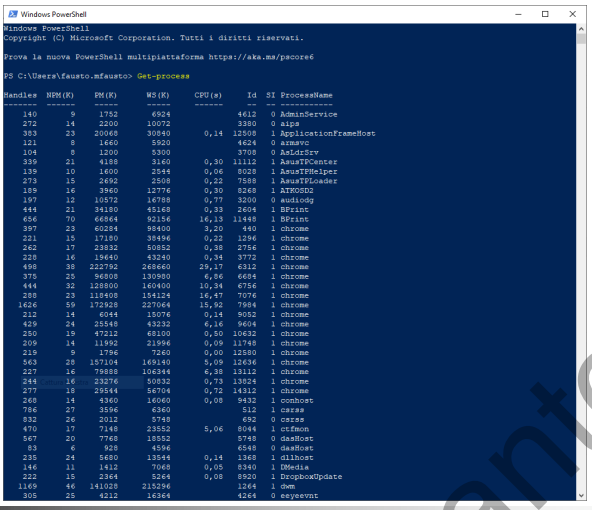
tasklist

```
Prompt dei comandi
Microsoft Windows [Versione 10.0.15063]
(c) 2017 Microsoft Corporation. Tutti i diritti sono riservati.

C:\Users\fausto.mfausto>tasklist

Nome immagine                  PID Nome sessione     Sessione n. Utilizzo mem
========================= ======== ================ =========== ============
System Idle Process              0 Services                   0          8 K
System                           4 Services                   0        960 K
smss.exe                       444 Services                   0      1.176 K
csrss.exe                      628 Services                   0      5.716 K
wininit.exe                    756 Services                   0      6.816 K
csrss.exe                      764 Console                    1      5.944 K
services.exe                   832 Services                   0     10.608 K
lsass.exe                      840 Services                   0     15.784 K
svchost.exe                    952 Services                   0      3.944 K
svchost.exe                    976 Services                   0     26.016 K
fontdrvhost.exe               1000 Services                   0      4.768 K
svchost.exe                    528 Services                   0     14.384 K
svchost.exe                    664 Services                   0      6.916 K
winlogon.exe                  1032 Console                    1      9.844 K
fontdrvhost.exe               1092 Console                    1     13.204 K
dwm.exe                       1168 Console                    1     52.368 K
svchost.exe                   1260 Services                   0      9.828 K
svchost.exe                   1284 Services                   0     15.580 K
svchost.exe                   1300 Services                   0     10.756 K
svchost.exe                   1376 Services                   0      6.164 K
svchost.exe                   1476 Services                   0     10.844 K
```


6

Processi attivi - Windows - PowerShell

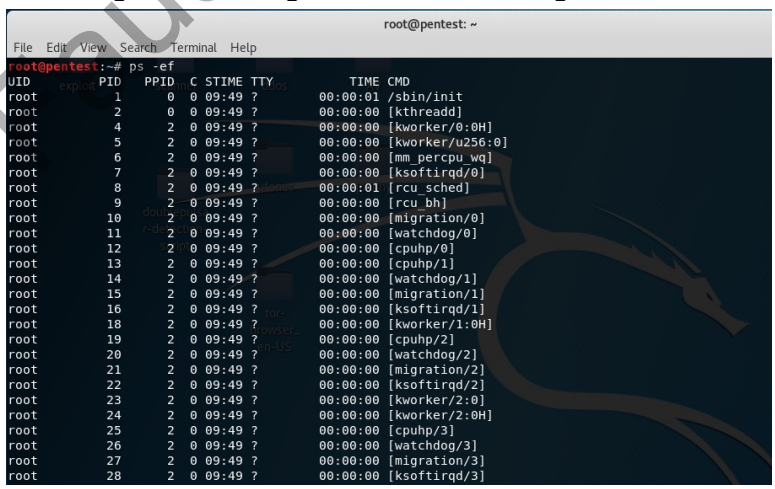
Get-Process



7

Processi attivi - Linux - BSD

ps -ef ps -aux - man ps



8

Processi attivi - MacOS

`ps -ef`

```
Last login: Fri Oct  4 16:37:52 on console
faustomarcantoni@MacBook-Pro-di-Fausto ~ % ps -aux
ps: No user named 'x'
faustomarcantoni@MacBook-Pro-di-Fausto ~ % ps -ef
  UID   PID  PPID   C STIME   TTY           TIME CMD
    0     1     0   0 Ven04pm ??        26:52.46 /sbin/launchd
    0   573     1   0 Ven04pm ??        24:00.54 /usr/libexec/logd
    0   574     1   0 Ven04pm ??         0:00.12 /usr/libexec/smd
    0   575     1   0 Ven04pm ??         5:34.13 /usr/libexec/UserEventAgent (System)
    0   577     1   0 Ven04pm ??         4:15.07 /System/Library/Frameworks/CoreServices.framewor
    0   578     1   0 Ven04pm ??         1:12.51 /System/Library/PrivateFrameworks/MediaRemote.fr
    0   581     1   0 Ven04pm ??         0:49.44 /usr/sbin/systemstats --daemon
  278   583     1   0 Ven04pm ??         0:11.71 /System/Library/PrivateFrameworks/MobileAccessor
    0   584     1   0 Ven04pm ??        61:37.15 /usr/libexec/configd
    0   585     1   0 Ven04pm ??         0:00.02 endpointsecurityd
    0   586     1   0 Ven04pm ??        29:12.05 /System/Library/CoreServices/powerd.bundle/power
    0   587     1   0 Ven04pm ??         0:28.45 /usr/libexec/IOMFB_bics_daemon
    0   589     1   0 Ven04pm ??         0:01.58 /usr/libexec/amfid
    0   591     1   0 Ven04pm ??         0:03.25 /usr/libexec/remoted
    0   593     1   0 Ven04pm ??         0:00.07 /usr/libexec/keybagd -t 15
  200   594     1   0 Ven04pm ??         0:02.81 /System/Library/PrivateFrameworks/MobileSoftware
    0   596     1   0 Ven04pm ??         0:00.68 /usr/libexec/watchdogd
    0   600     1   0 Ven04pm ??        10:37.66 /System/Library/Frameworks/CoreServices.framewor
  240   601     1   0 Ven04pm ??         0:00.91 /System/Library/CoreServices/iconservicesd
    0   602     1   0 Ven04pm ??         0:00.78 /usr/libexec/kernelmanagerd
    0   603     1   0 Ven04pm ??         0:12.02 /usr/libexec/diskarbitrationd
    0   606     1   0 Ven04pm ??         2:13.86 /usr/libexec/coreduetd
    0   607     1   0 Ven04pm ??         9:27.97 /usr/sbin/syslogd
    0   610     1   0 Ven04pm ??         0:49.56 /usr/libexec/thermalmonitord
    0   611     1   0 Ven04pm ??        41:57.32 /usr/libexec/opendirectoryd
    0   613     1   0 Ven04pm ??        12:04.64 /System/Library/PrivateFrameworks/ApplePushServi
```


9

Processi attivi - Linux - BSD

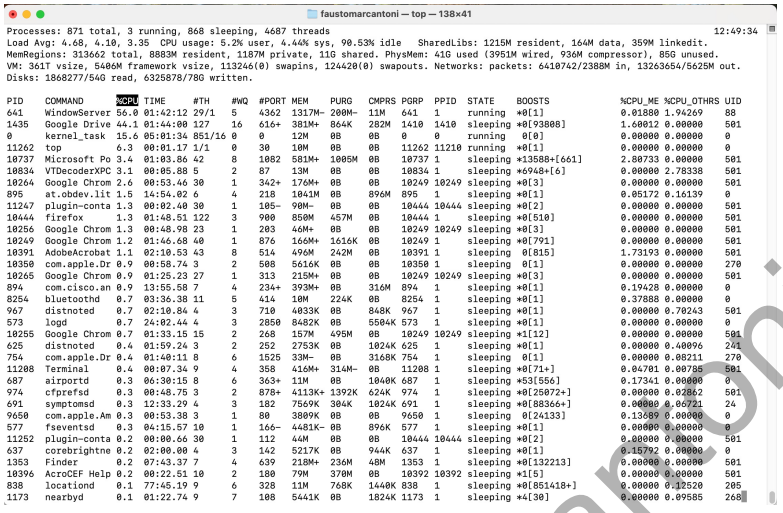
`top`

```
top - 15:46:17 up 2 min,  1 user,  load average: 0.00, 0.00, 0.00
Tasks: 262 total,   1 running, 261 sleeping,   0 stopped,   0 zombie
%Cpu(s):  0.2 us,  0.0 sy,  0.0 ni, 99.8 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :   1978.0 total,   1120.0 free,    579.1 used,    365.9 buff/cache
MiB Swap:    976.0 total,    976.0 free,      0.0 used.   1398.9 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
   1608 studente  20   0   11680   5080   2924 R   0.3   0.3   0:00.22 top
      1 root      20   0  185088  11660   8464 S   0.0   0.6   0:00.21 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:00.00 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
      5 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 slub_flushwq
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 netns
      7 root      20   0       0      0      0 I   0.0   0.0   0:00.01 kworker/0:0-cgwb_release
      8 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
      9 root      20   0       0      0      0 I   0.0   0.0   0:00.16 kworker/u4:0-events_unbound
     10 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
     11 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_rude_kthread
     12 root      20   0       0      0      0 I   0.0   0.0   0:00.00 rcu_tasks_trace_kthread
     13 root      20   0       0      0      0 S   0.0   0.0   0:00.01 ksoftirqd/0
     14 root      20   0       0      0      0 I   0.0   0.0   0:00.04 rcu_sched
     15 root      rt   0       0      0      0 S   0.0   0.0   0:00.00 migration/0
     16 root      20   0       0      0      0 I   0.0   0.0   0:00.00 kworker/0:1-events
     17 root      20   0       0      0      0 S   0.0   0.0   0:00.00 cpuhp/0
```


10

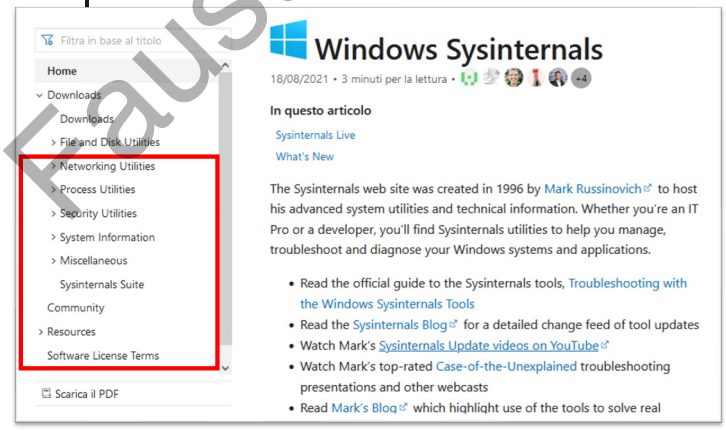
Processi Attivi - MacOs

top



11

Windows Sysinternals



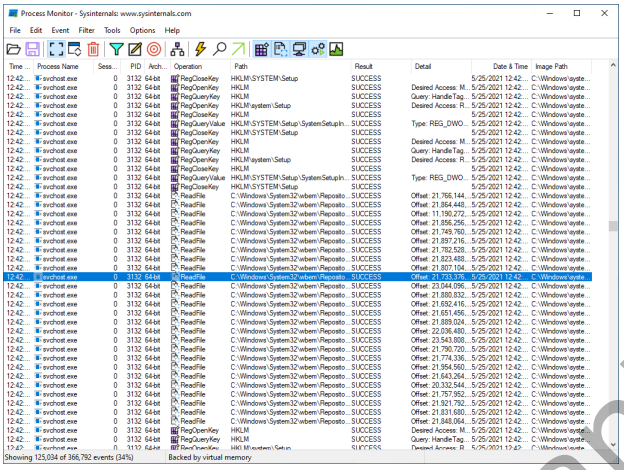
https://learn.microsoft.com/it-it/sysinternals/downloads/sysinternals-suite

12

Reti di Elaboratori

Process Monitor

Process Monitor

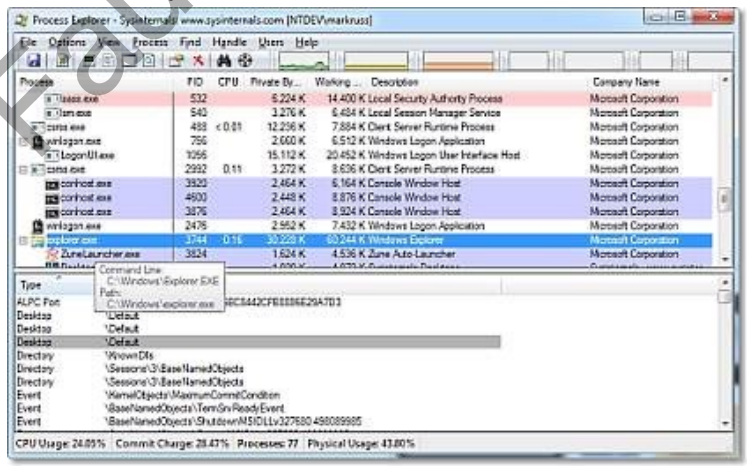


13

Reti di Elaboratori

Process Explorer

Process Explorer



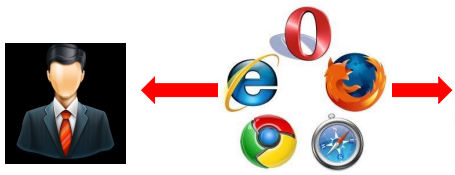
14

Applicazioni: terminologia essenziale

- Un agente utente (user agent) è:

  un'interfaccia tra l'utente e l'applicazione di rete

  - Browser Web
  - E-mail: lettore di posta
  - streaming audio/video: lettore di file audio/video



15

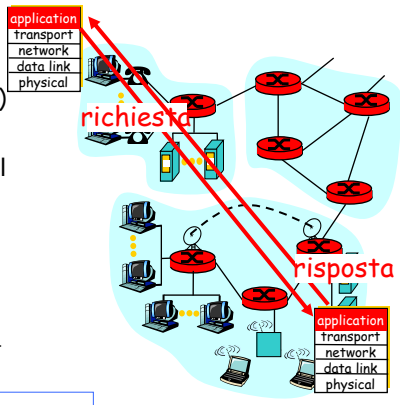
Paradigma client-server

Applicazione di rete tipica consiste di due parti: *client* e *server*

Client:
- Inizia il dialogo col server ("speaks first")
- Di solito richiede un servizio
- Nel caso del Web, il client è integrato nel browser

Server:
- Fornisce il servizio al client, su richiesta
- E' sempre pronto per le richieste

Es., un Web server invia una pagina Web richiesta, un mail server accede alla casella di posta elettronica

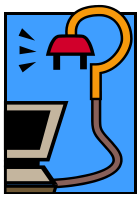


Esempi di sistemi client/server:
Database server: per la gestione di grandi moli di dati
File server: per la condivisione dei file
FTP server: per la gestione dell'upload/download dei file
Groupware: per la gestione d'informazioni riguardanti gruppi di lavoro
Print server: per la condivisione delle stampanti
Web server: per la gestione dell'interazione via web tra server e client

16

Protocolli di livello applicativo: servizi dagli strati inferiori e identificazione

API: Application Programming Interface
- Definisce l'interfaccia tra applicazione e strato di trasporto
- Socket: API Internet
  - Due processi (applicazione nel modello client server) comunicano inviando/leggendo dati nel/dal socket

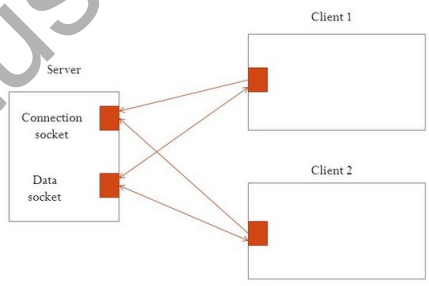


come può un processo "identificare" quello con cui intende comunicare?

- indirizzo IP — dell'host su cui l'altro processo è in esecuzione
- numero di porta (port number) — permette all'host ricevente di identificare il processo locale destinatario del messaggio

17

Socket




```
Socket socket = getSocket(type = "TCP")
connect(socket, address = "1.2.3.4", port = "80")
send(socket, "Hello, world!")
close(socket)
```

https://tools.ietf.org/html/rfc147

18

Reti di Elaboratori

LA CLASSE Socket

Costruire una Socket significa aprire la comunicazione verso l'altra parte

```
public Socket(String remoteHost, int remotePort)
```

Crea una socket stream e la collega alla porta specificata della macchina remota corrispondente al nome dato

Esempio

```
Socket s = new Socket("www.nasa.gov",8080);
```

```
public InputStream getInputStream()
```
– restituisce lo stream di input da cui leggere i dati (byte) che giungono dall'altra parte

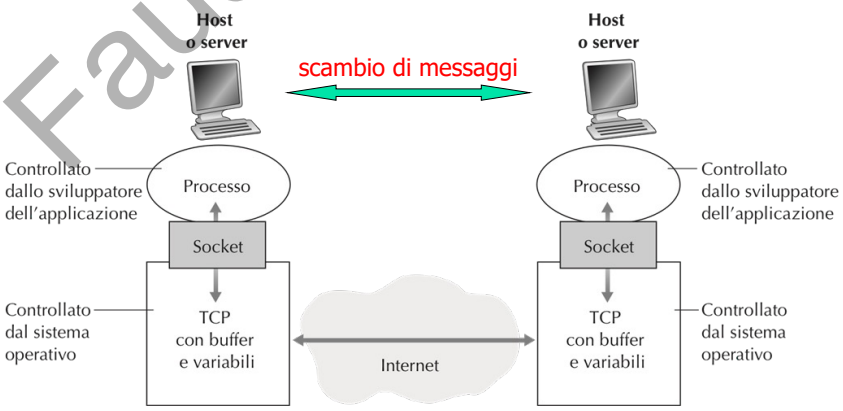
```
public OutputStream getOutputStream()
```
– restituisce lo stream di output su cui scrivere i dati (byte) da inviare all'altra parte

```
public synchronized void close()
```
– chiude la connessione e libera la risorsa

http://www0.mi.infn.it/~cmp/CorsoReti/slides03/Lab3-sockets-java.pdf

Reti di Elaboratori

Socket: funzionamento di base



Reti di Elaboratori

Socket: funzionamento di base

Un protocollo a livello di applicazione definisce come i processi di un'applicazione, in esecuzione su sistemi terminali diversi, si scambiano i messaggi.

- tipo di messaggio (di richiesta o di risposta)
- sintassi del messaggio (campi del messaggio)
- semantica del messaggio (significato delle informazioni dei campi)
- regole (come e quando avviene lo scambio dei messaggi)

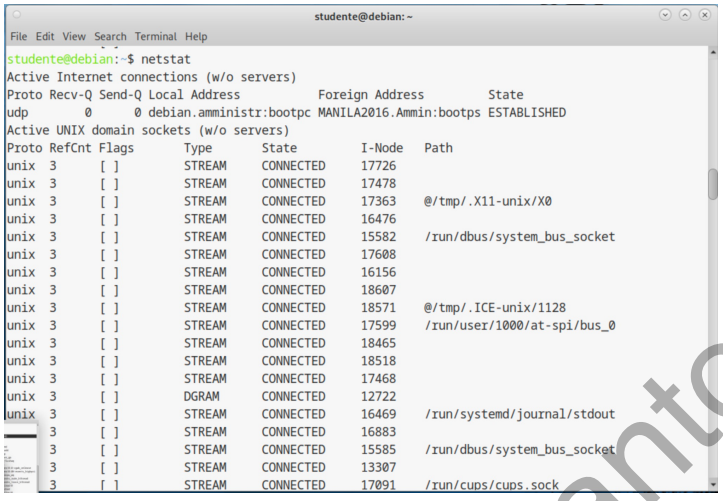
Reti di Elaboratori

netstat

Reti di Elaboratori

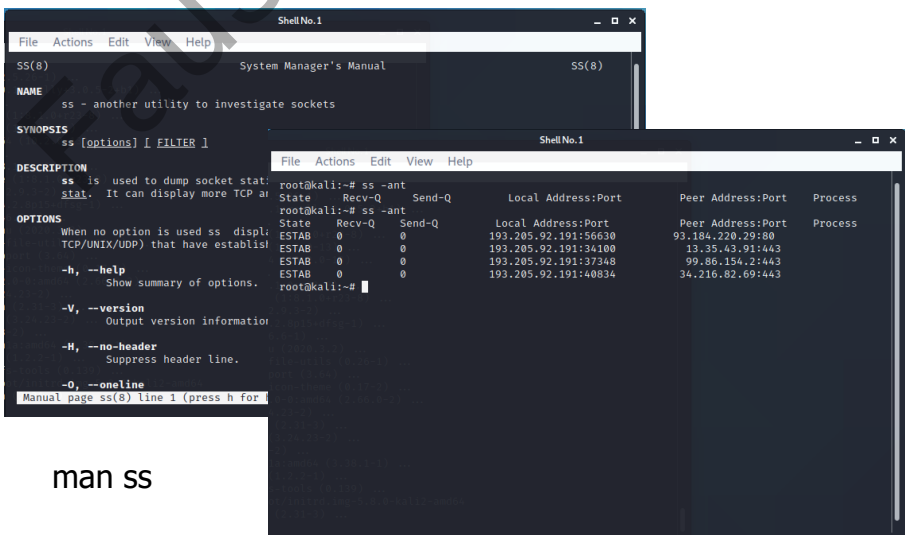
Netstat in linux

sudo apt install net-tools



Reti di Elaboratori

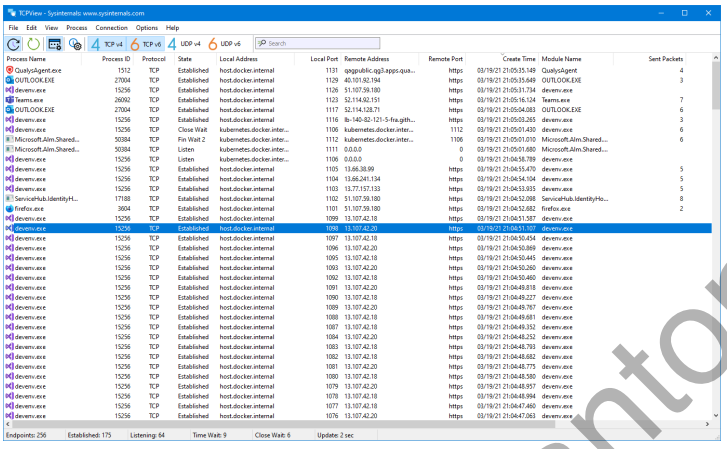
ss - another utility to investigate sockets



man ss

TCPView

TCPView



25

Requisiti delle applicazioni

Perdita dati (Data loss)

- Alcune applicazioni (es., audio) sono tolleranti (fino a un certo punto)
- Altre (es.,Web, FTP, telnet) richiedono affidabilità totale

Ritardo

- Alcune applicazioni (es., telefonia Internet, giochi interattivi in rete) richiedono una ritardo minimo per funzionare con qualità sufficiente
- Altre non ammettono ritardi (es., telefonia su internet, IPTV)

Banda

- Alcune applicazioni (soprattutto multimediali) richiedono una banda minima per poter *"funzionare"*
- Altre (dette "elastiche") usano la banda a disposizione

Nota: alcuni requisiti sono determinati da esigenze percettive umane
(es. ritardo nella telefonia Internet)

26

Reti di Elaboratori

Requisiti di alcune applicazioni di rete

Applicazione	Tolleranza alla perdita di dati	Ampiezza di banda	Sensibilità al tempo
Trasferimento file	No	Variabile	No
E-mail	No	Variabile	No
Documento Web	No	Variabile (pochi Kbps)	No
Audio/Video in tempo reale	Si	Audio: dal pochi Kbps a 1 Mbps Video: da 10 Kbps a 5 Mbps	Si: centinaia di ms.
Audio/Video memorizzati	Si	Audio: dal pochi Kbps a 1 Mbps Video: da 10 Kbps a 5 Mbps	Si: pochi secondi
Giochi interattivi	Si	Fino a 10 Kbps **(se basta???)**	Si: centinaia di ms.
Messaggistica istantanea	No	Variabile	Si e No

27

Reti di Elaboratori

Servizi offerti dai protocolli di trasporto Internet

Servizio TCP :

- ***Orientato alla connessione:***
  - richiesto "setup" tra client e server
- ***Trasporto affidabile (reliable transfer):***
  - tra processi mittente e ricevente
- ***Controllo di flusso (flow control):***
  - il mittente non sommerge il ricevente
- ***Controllo della congestione (congestion control):***
  - si limita il mittente quando la rete è sovraccarica
- ***Non offre:***
  - garanzie di banda e ritardo minimo

28

Servizi offerti dai protocolli di trasporto Internet

Servizio UDP :

- ***Trasporto non affidabile tra processi mittente e ricevente***
- ***Non offre:***
  - connessione, affidabilità, controllo di flusso, controllo di congestione, garanzie di ritardo e banda

Perché esiste/si usa UDP?

- ☑ Nessun ritardo dovuto al setup di connessione
- ☑ Semplicità dovuta all'assenza di stato di connessione
- ☑ Pochi dati di intestazione del segmento
- ☑ Nessun controllo di congestione
- ☑ Informazioni spedite a raffica

Può essere conveniente per le applicazioni multimediali quali audio, video, videoconferenza, telefonia internet, tutte applicazioni in cui è possibile avere una tolleranza nella perdita di dati.

Applicazioni Internet: loro protocolli e protocollo di transporto usato

Applicazione	Protocollo Applicativo	Protocollo di Trasporto
e-mail	smtp [RFC 821]	TCP
remote terminal	telnet [RFC 854]	TCP
www	http [RFC 2068]	TCP
name server	dns [RFC 1034 e RFC 1035]	TCP - UDP
file transfer	ftp [RFC 959]	TCP
management protocol	snmp [RFC 1157 e RFC 3416]	UDP
streaming multimedia	proprietario (es. RealNetworks)	TCP - UDP
remote file server	nfs [RFC 1094, 1813, 3530]	TCP - UDP
internet telephony	VoIP(RTP-Real-time Transport Protocol)	UDP

Protocolli a livello di applicazione

Un **protocollo a livello di applicazione** definisce come i processi di un'applicazione, in esecuzione su sistemi periferici diversi, si scambiano i messaggi.

È importante distinguere tra:

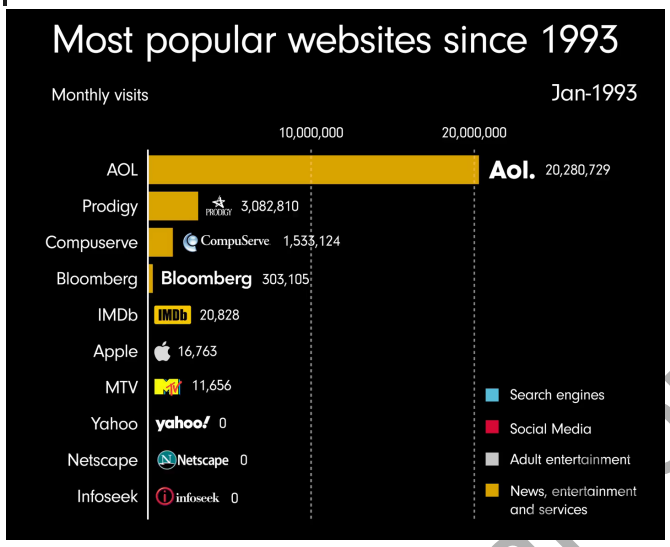
- **applicazioni di rete**
- **protocolli a livello di applicazione**

applicazioni di rete	protocolli	
il Web	http	Hypertext Transfer Protocol
la posta elettronica	smtp	Simple Mail Transfer Protocol
il servizio di directory	ldap	Lightweight Directory Access Protocol
il trasferimento di file	ftp	File Transport Protocol
le applicazioni P2P	torrent	proprietario

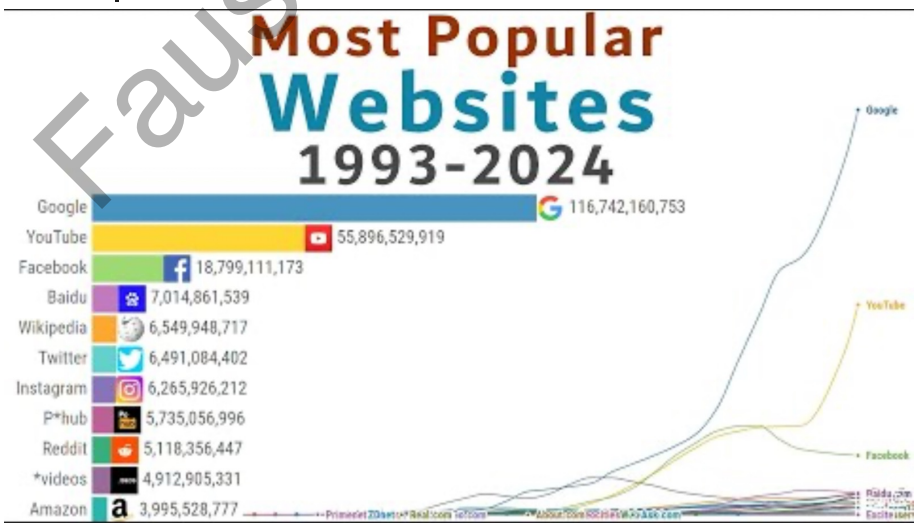
WEB - WWW - World Wide Web

WEB - WWW - World Wide Web

Most popular websites since 1993

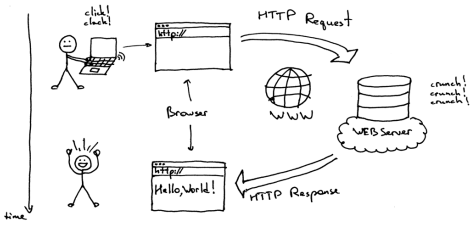


33



34

Web e HTTP



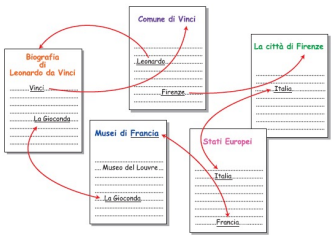
World Wide Web

Il Web è implementato attraverso un insieme di standard, i principali dei quali sono i seguenti:

- HTML: il linguaggio di markup con cui sono scritte e descritte le pagine web;
- HTTP: il protocollo di rete appartenente al livello di applicazione del modello ISO/OSI su cui è basato il Web;
- URL: lo schema di identificazione, e quindi di rintracciabilità, dei contenuti e dei servizi del Web.

Ipertesto

Un **ipertesto** è un insieme di documenti messi in relazione tra loro per mezzo di parole chiave.



WWW: terminologia essenziale

- Pagina Web:
  - È costituita da "oggetti" (di solito: pagina HTML iniziale+oggetti indirizzati)
  - È indirizzata da una URL
- URL (Uniform Resource Locator)
  - Identifica un oggetto nella rete e specifica il modo per accedere ad esso
  - Ha tre componenti:
    - protocollo
    - nome dell'host
    - percorso nell'host

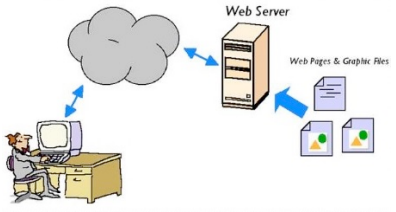
`http://www.someSchool.edu/someDept/pic.gif`

http://tools.ietf.org/html/rfc3986

37

WWW: terminologia essenziale

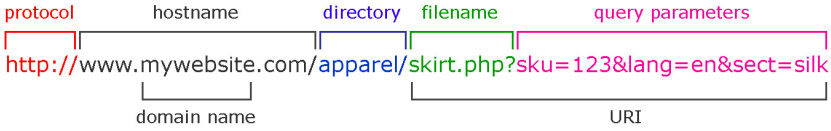
- Uno user agent per il Web è detto browser:
  - MS Internet Explorer/Edge/Cromium
  - Mozilla Firefox
  - Opera
  - Chrome
  - Safari
  - Brave
- Un server per il Web è detto Web server:
  - Apache (pubblico dominio)
  - MS Internet Information Server
  - lighttpd
  - nginx



38

Reti di Elaboratori

WWW: terminologia essenziale



Protocollo: Visualizza il protocollo utilizzato per l'accesso al server. I protocolli più comuni sono l'HTTP, HTTPS, FTP, MMS ecc. Se il protocollo non viene specificato generalmente il browser utilizza "HTTP://" come predefinito.

Hostname: visualizza l'indirizzo fisico del server su cui risiede la risorsa. Come detto in precedenza può essere costituito da un nome di dominio o da un Indirizzo IP.

Directory: è opzionale e, se presente, indica il pathname nel file system del server che identifica la risorsa, che generalmente è una pagina web, un'immagine o un file multimediale.

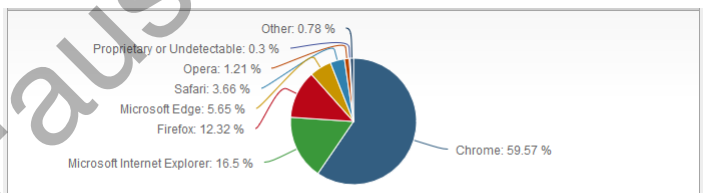
Filename: anche questo opzionale, visualizza il nome e l'estensione della risorsa che stiamo visualizzando.

Querystring: opzionale, se richiesto al termine dell'url è possibile aggiungere una query string separandola utilizzando il simbolo "?". La querstring è una stringa di caratteri che consente di passare al server uno o più parametri.

39

Reti di Elaboratori

WWW: quali sono I browser più utilizzati



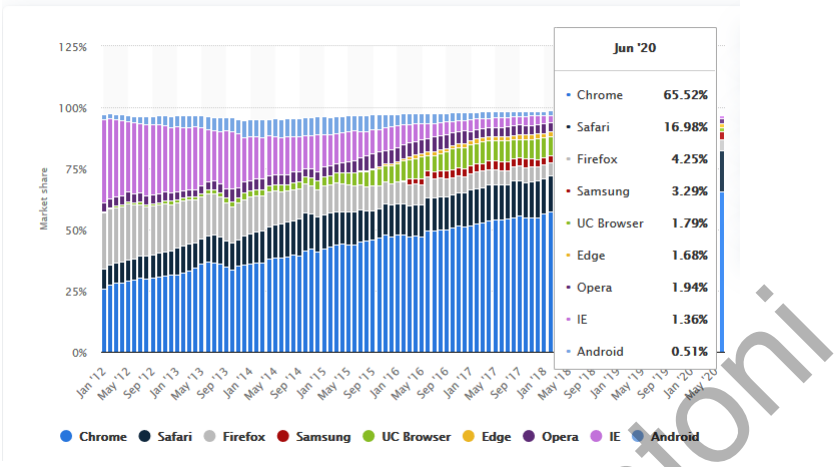
BROWSER	TOTAL MARKET SHARE
Chrome	59.57%
Microsoft Internet Explorer	16.50%
Firefox	12.32%
Microsoft Edge	5.65%
Safari	3.66%
Opera	1.21%
Proprietary or Undetectable	0.30%
Konqueror	0.01%
Android Browser	0.00%
Obigo	0.00%

http://marketshare.hitslink.com/report.aspx?qprid=0

40

Reti di Elaboratori

WWW: quali sono I browser più utilizzati

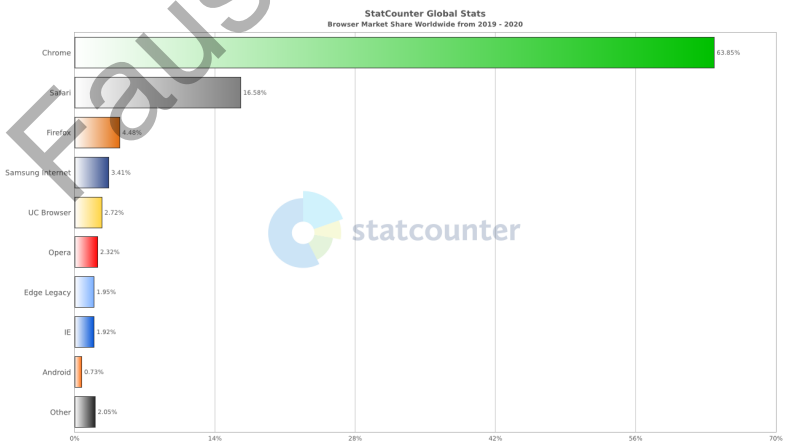


https://www.statista.com/statistics/268254/market-share-of-internet-browsers-worldwide-since-2009/

41

Reti di Elaboratori

WWW: quali sono I browser più utilizzati



https://gs.statcounter.com/browser-market-share#yearly-2019-2020-bar

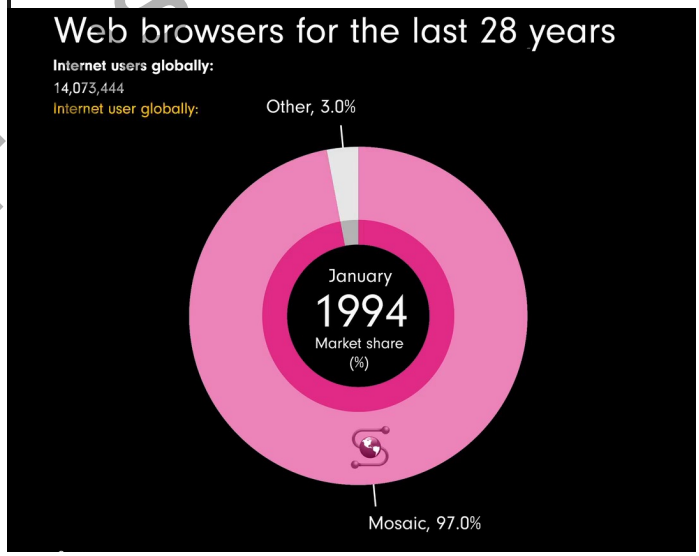
42

Mosaic

https://www.my-internet-explorer.com/mosaic/

43

The Rise and Fall of Popular Web Browsers Since 1994

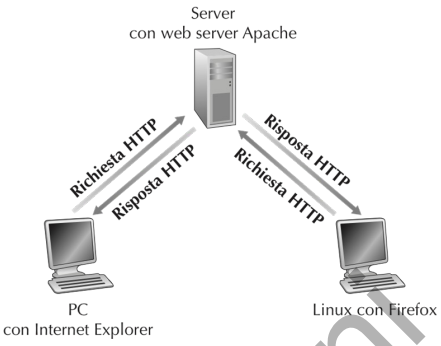


44

Il Web: protocollo http

http: hypertext transfer protocol

- Protocollo di livello applicativo per il Web
- Usa il modello client/server
  - ***client:*** browser che richiede, riceve e "mostra" oggetti Web
  - ***server:*** Web server che invia oggetti in risposta alle richieste
- http1.0: RFC 1945
- http1.1: RFC 2068



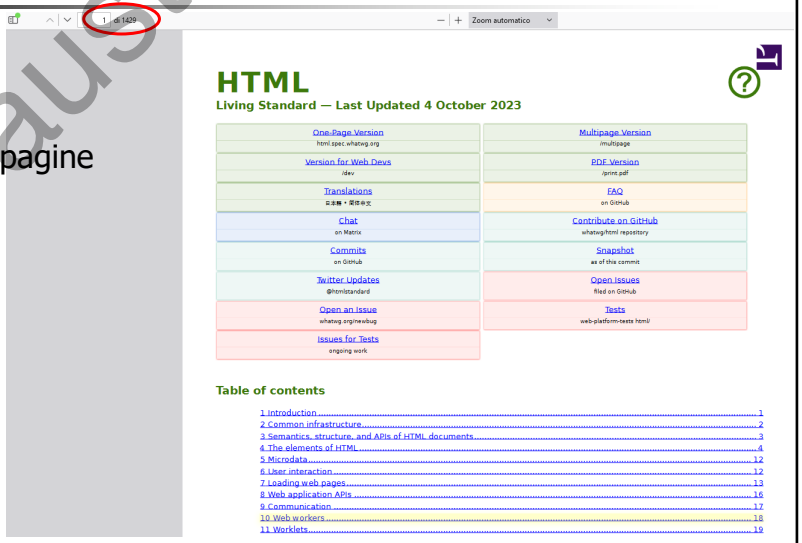
HTML ≠ HTTP

- http://www.w3.org/
- http://www.w3.org/MarkUp/

45

HTML 5

1476 pagine



https://html.spec.whatwg.org/multipage/

46

Storia di HTTP

Versioni HTTP:

0.9: un semplicissimo protocollo client-server di sola richiesta di risorse HTML, senza flessibilità né nella direzione, né nel formato delle risorse. Utilizzata nel primo prototipo WWW e nei primi server NCSA.

1.0 (RFC 1945): il protocollo diventa generico e definisce la statelessness (apolidia), e definisce alcuni metodi anche per l'upload di dati. Utilizzato fino al 1998-99

1.1 (RFC 2068, 2069 e poi 2616, 2617): la versione attuale di HTTP, specifica meglio i meccanismi di caching, permette multi-homing e connessioni persistenti.

HTTP-NG doveva essere la naturale evoluzione di HTTP, ma il WG IETF fallì miseramente nel raggiungere l'obiettivo, e l'evoluzione del protocollo si fermò.

2.0 (RFC 7540): ottimizzare il trasferimento delle informazioni tra server e browser, minore latenza, compressione degli header HTTP, multiplexing di richieste e risposte per ricevere più oggetti in un'unica sessione.

47

Il protocollo http

Client-server

In HTTP esistono due ruoli specifici: il *client* attiva la connessione e richiede dei servizi. Il *server* accetta la connessione, nel caso identifica il richiedente, e risponde alla richiesta. Alla fine chiude la connessione.

Protocollo generico

HTTP è indipendente dal formato dati con cui vengono trasmesse le risorse. Può funzionare per documenti HTML come per binari, eseguibili, oggetti distribuiti o altre strutture dati più o meno complicate.

Statelessness (apolidia)

Il server non è tenuto a mantenere informazioni che persistano tra una connessione e la successiva sulla natura, identità e precedenti richieste di un client. Il client è tenuto a ricreare da zero il contesto necessario al server per rispondere.

48

Reti di Elaboratori

Il protocollo http (cont.)

http: usa TCP:

- Il client inizia una connessione TCP (socket) verso il server sulla porta 80
- Il server accetta la connesione TCP dal client (sempre in ascolto sulla porta 80)
- Vengono scambiati messaggi http (messaggi del protocollo di livello applicativo) tra il browser (client http) e il Web server (server http)
- La connessione TCP è chiusa

http è "stateless"

- Il server non mantiene informazione sulle richieste precedenti del client

Non confondere i file di log dell'applicativo

I protocolli che mantengono informazione di stato sono complessi (es. TCP) !

49

Reti di Elaboratori

HTTP connections

Nonpersistent HTTP

- Al massimo un oggetto è trasmesso per ogni connessione TCP.
- HTTP/1.0 usa nonpersistent HTTP

Persistent HTTP

- Oggetti multipli possono essere trasmessi in una singola connessione TCP
- HTTP/1.1 usa persistent connections in default mode

50

Reti di Elaboratori

Esempio http non persistente

L' utente accede alla URL:
http://www.someSchool.edu/someDepartment/home.index
(contiene testo e riferimenti a 10 immagini jpeg)

1a. Il client http inizia una connessione TCP verso il server (processo) http sull' host www.someSchool.edu. La porta 80 è quella standard (default) per i server http.

1b. Il server http presso l' host www.someSchool.edu è "in ascolto" sulla porta 80. "Accetta" la richiesta di connessione e ne dà conferma al client

2. Il client http invia un messaggio di richiesta http (*request message*) contenente la URL

3. Il server http riceve il messaggio di richiesta, costruisce un messaggio di risposta (*response message*) contenente l'oggetto richiesto (someDepartment/home.index), inoltra il messaggio nel socket

Tempo

51

Reti di Elaboratori

Esempio http (cont.)

4. Il server http chiude la connessione TCP.

5. Il client http riceve il messaggio di risposta contenente il file html, visualizza la pagina html. Analizzando **(parser)** il file html, il browser trova i riferimenti a 10 oggetti jpeg

parser è un software in grado di effettuare l'analisi sintattica di un testo rispetto ad un determinato linguaggio

6. I passi 1-4 sono ripetuti per ciascuno dei 10 oggetti jpeg

Tempo

52

Response time modeling

Definizione di RTT (round trip time):
tempo occorrente ad un piccolo pacchetto per viaggiare dal client al server e ritornare al client

Response time:

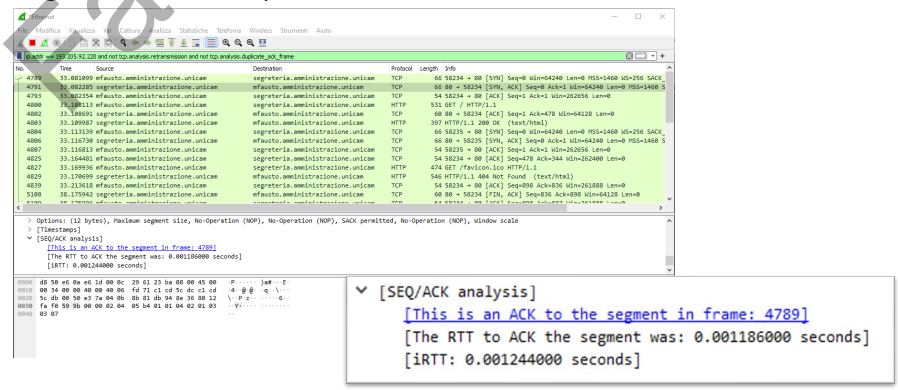
- 1 RTT per iniziare la connessione TCP
- 1 RTT per HTTP request e pochi bytes per HTTP response di ritorno
- file transmission time

total = 2RTT+transmit time

```
[SEQ/ACK analysis]
    [This is an ACK to the segment in frame: 181]
    [The RTT to ACK the segment was: 0.020975000 seconds]
    [iRTT: 0.021034000 seconds]
```


Collegarsi al sito web→ analizzare RTT

Aprire wireshrk
Selezionare l'interfaccia usata
Collegarsi al sito web: http://90.147.42.137

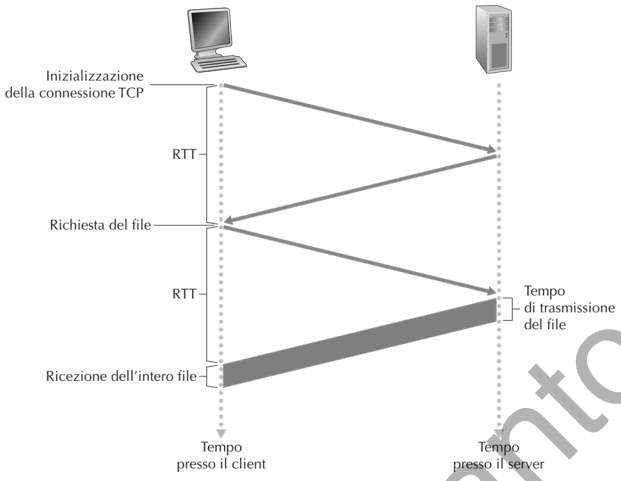


ip.addr == 90.147.42.137 and not tcp.analysis.retransmission and not tcp.analysis.duplicate_ack_frame

Reti di Elaboratori

Response time modeling

Calcolo approssimato del tempo necessario per richiedere e ricevere un file HTML.



55

Reti di Elaboratori

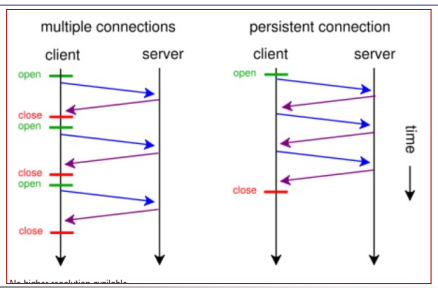
Persistent HTTP

Nonpersistent HTTP:

- richiede 2 RTTs per object
- OS deve lavorare per allocare i buffer del TCP e le variabili del TCP devono essere conservate sia nel client che nel server
- In questo caso possono essere aperte connessioni TCP in parallelo

Persistent HTTP:

- server lascia aperta la connessione TCP dopo aver spedito la risposta
- i messaggi HTTP tra lo stesso client e server usano la medesima connessione



56

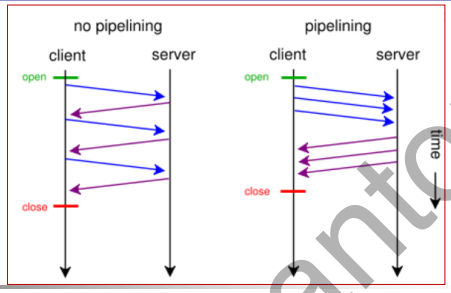
Persistent HTTP

Persistent without pipelining (senza parallelismo):

- Il client emette una nuova richiesta solo quando la risposta precedente è stata ricevuta
- richiede 1 RTT per ciascun oggetto referenziato

Persistent with pipelining (con parallelismo):

- default in HTTP/1.1
- Il client emette una richiesta non appena incontra un riferimento
- richiede 1 RTT per tutti gli oggetti referenziati



57

Nel nostro esempio : una pagina + 10 oggetti

Esercizio: ***una pagina + 10 oggetti***

Calcolare il "response time" per le diverse tipologie di connessioni

- HTTP non persistente
  - 2RTT + 20RTT
- HTTP persistente senza parallelismo
  - 2RTT + 10RTT
- HTTP persistente con parallelismo
  - 2RTT + 1RTT

58

Formato dei messagi http

- Due tipi di messaggi http: *request, response*
- Messaggio http request:
  - ASCII (formato testo leggibile)

Chiudi la connessione al termine della richiesta

Request line (GET, POST, HEAD commands)

header lines

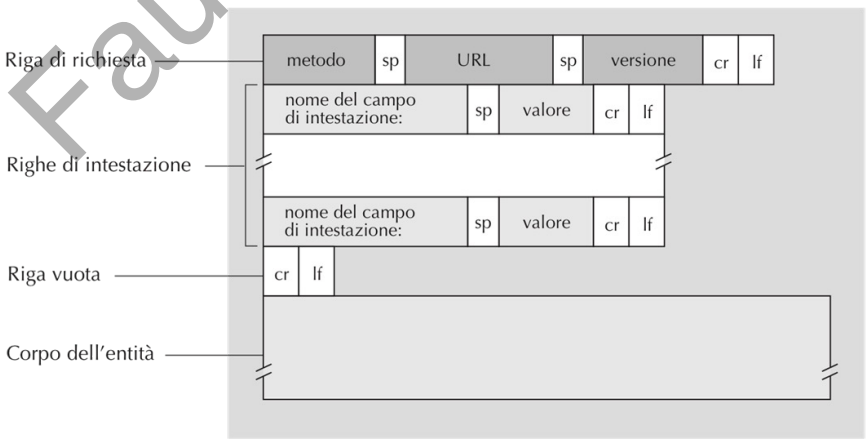
Carriage return, line feed indica fine messaggio

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
Connection: close
User-agent: Mozilla/4.0
Accept: text/html, image/gif,image/jpeg
Accept-language:fr
```

(extra carriage return, line feed)

59

Message http request : formato generale



60

Message http request : formato generale

- **HEAD**
  - Simile a GET, ma richiede solo le informazioni dell'header.
  - Utile per controllare se un documento esiste e quanto è recente.
- **POST**
  - Simile a GET, ma codifica gli input in modo diverso.
  - Utile per proporre i contenuti dei form a un programma CGI.

CGI: Common Gateway Interface

- **PUT**
  - Trasferisce un documento al server.
  - Presente a partire dalla versione HTTP/1.1.
- **DELETE**
  - Elimina un documento dal server.
  - Presente a partire dalla versione HTTP/1.1.

Message http request : formato generale

- **TRACE**
  - Attraverso questo metodo un client può vedere cosa viene ricevuto dall'altra parte della catena delle richieste in modo tale da eseguire delle diagnosi
  - La richiesta TRACE non deve includere un corpo
  - Se la richiesta è valida la risposta dovrebbe contenere una risposta 200 (OK) e l'intero messaggio di richiesta nel corpo dell'entità con Content-Type "message/http"
  - Le risposte non dovrebbero essere soggette a cache.
- **CONNECT**
  - Usata con i proxy per tunnel (es. SSL)

Reti di Elaboratori

The Method token

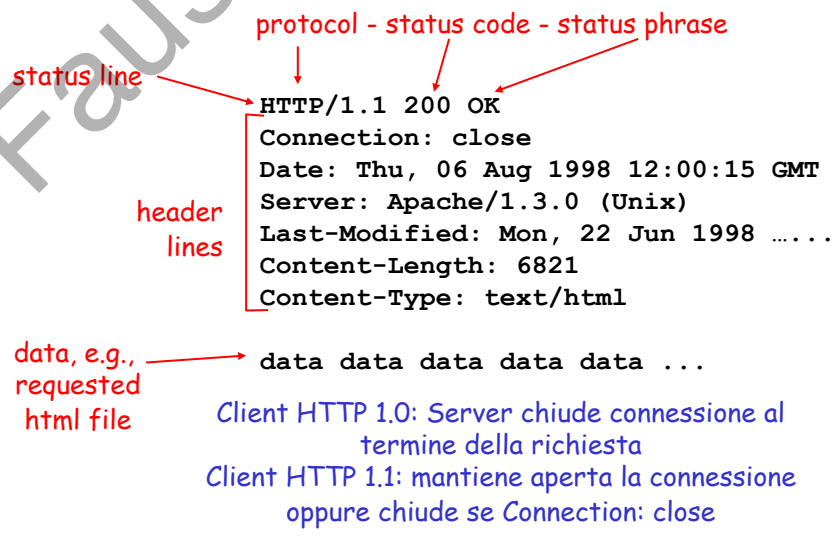
https://www.w3.org/Protocols/rfc2616/rfc2616-sec5.html
https://www.rfc-editor.org/rfc/rfc9110.html

Method Name	Description	Section
GET	Transfer a current representation of the target resource.	9.3.1
HEAD	Same as GET, but do not transfer the response content.	9.3.2
POST	Perform resource-specific processing on the request content.	9.3.3
PUT	Replace all current representations of the target resource with the request content.	9.3.4
DELETE	Remove all current representations of the target resource.	9.3.5
CONNECT	Establish a tunnel to the server identified by the target resource.	9.3.6
OPTIONS	Describe the communication options for the target resource.	9.3.7
TRACE	Perform a message loop-back test along the path to the target resource.	9.3.8

63

Reti di Elaboratori

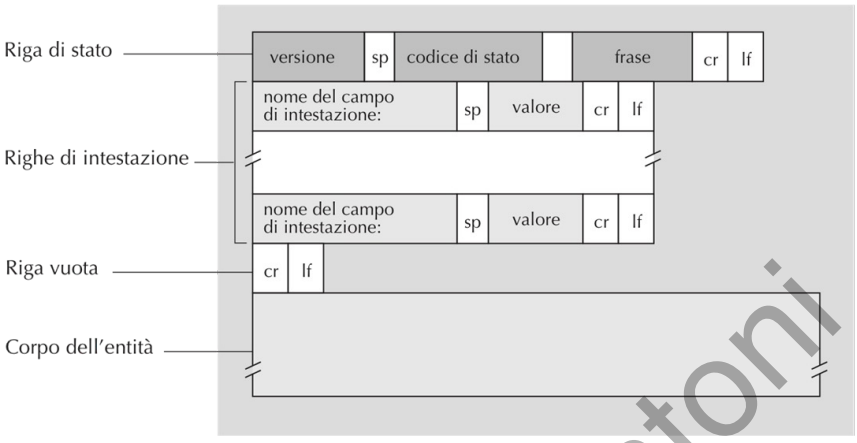
Formato del messaggio http response



Client HTTP 1.0: Server chiude connessione al termine della richiesta
Client HTTP 1.1: mantiene aperta la connessione oppure chiude se Connection: close

64

Message http response : formato generale



65

Hypertext Transfer Protocol (HTTP) Field Name Registry

iana
Internet Assigned Numbers Authority

Hypertext Transfer Protocol (HTTP) Field Name Registry

Created
2021-10-01
Last Updated
2022-08-25
Available Formats
XML HTML Plain text

Registry included below

- Hypertext Transfer Protocol (HTTP) Field Name Registry

https://www.iana.org/assignments/http-fields/http-fields.xhtml

66

Reti di Elaboratori

Risposta: codici di stato

Prima riga del messaggio di risposta server->client.
Alcuni esempi:

200 OK
- Successo, oggetto richiesto più avanti nel messaggio

301 Moved Permanently
- L'oggetto richiesto è stato spostato. Il nuovo indirizzo è specificato più avanti (Location:)

400 Bad Request
- Richiesta incomprensibile al server

404 Not Found
- Il documento non è stato trovato sul server

505 HTTP Version Not Supported
- Il server non dispone della versione del protocollo

67

Reti di Elaboratori

Risposta: codici di stato

15. Status Codes

The status code of a response is a three-digit integer code that describes the result of the request and the semantics of the response, including whether the request was successful and what content is enclosed (if any). All valid status codes are within the range of 100 to 599, inclusive.

The first digit of the status code defines the class of response. The last two digits do not have any categorization role. There are five values for the first digit:

- 1xx (Informational): The request was received, continuing process
- 2xx (Successful): The request was successfully received, understood, and accepted
- 3xx (Redirection): Further action needs to be taken in order to complete the request
- 4xx (Client Error): The request contains bad syntax or cannot be fulfilled
- 5xx (Server Error): The server failed to fulfill an apparently valid request

https://www.rfc-editor.org/rfc/rfc9110.html#name-status-codes

68

Risposta: codici di stato

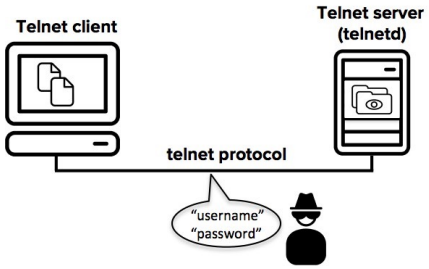
```
Status-Code    =
      "100"  ; Section 10.1.1: Continue
    | "101"  ; Section 10.1.2: Switching Protocols
    | "200"  ; Section 10.2.1: OK
    | "201"  ; Section 10.2.2: Created
    | "202"  ; Section 10.2.3: Accepted
    | "203"  ; Section 10.2.4: Non-Authoritative Information
    | "204"  ; Section 10.2.5: No Content
    | "205"  ; Section 10.2.6: Reset Content
    | "206"  ; Section 10.2.7: Partial Content
    | "300"  ; Section 10.3.1: Multiple Choices
    | "301"  ; Section 10.3.2: Moved Permanently
    | "302"  ; Section 10.3.3: Found
    | "303"  ; Section 10.3.4: See Other
    | "304"  ; Section 10.3.5: Not Modified
    | "305"  ; Section 10.3.6: Use Proxy
    | "307"  ; Section 10.3.8: Temporary Redirect
    | "400"  ; Section 10.4.1: Bad Request
    | "401"  ; Section 10.4.2: Unauthorized
    | "402"  ; Section 10.4.3: Payment Required
    | "403"  ; Section 10.4.4: Forbidden
    | "404"  ; Section 10.4.5: Not Found
    | "405"  ; Section 10.4.6: Method Not Allowed
    | "406"  ; Section 10.4.7: Not Acceptable
    | "407"  ; Section 10.4.8: Proxy Authentication Requ
    | "408"  ; Section 10.4.9: Request Time-out
    | "409"  ; Section 10.4.10: Conflict
    | "410"  ; Section 10.4.11: Gone
    | "411"  ; Section 10.4.12: Length Required
    | "412"  ; Section 10.4.13: Precondition Failed
    | "413"  ; Section 10.4.14: Request Entity Too Large
    | "414"  ; Section 10.4.15: Request-URI Too Large
    | "415"  ; Section 10.4.16: Unsupported Media Type
    | "416"  ; Section 10.4.17: Requested range not sati
    | "417"  ; Section 10.4.18: Expectation Failed
    | "500"  ; Section 10.5.1: Internal Server Error
    | "501"  ; Section 10.5.2: Not Implemented
    | "502"  ; Section 10.5.3: Bad Gateway
    | "503"  ; Section 10.5.4: Service Unavailable
    | "504"  ; Section 10.5.5: Gateway Time-out
    | "505"  ; Section 10.5.6: HTTP Version not supporte
    | extension-code
```

https://www.rfc-editor.org/rfc/rfc9110.html#name-status-codes

69

Cos'è "telnet"

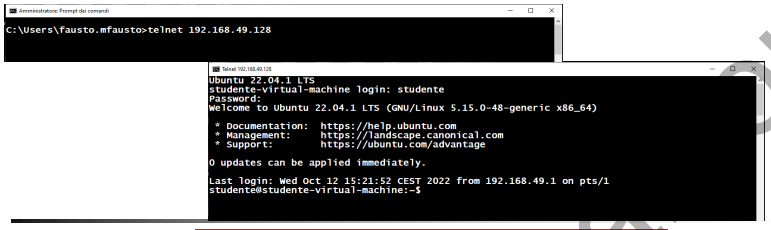
- Telnet è un'applicazione standard di Internet ed è disponibile nella maggior parte delle implementazioni del TCP/IP, indipendentemente dal sistema operativo host.
- Si tratta di un semplice protocollo di **login remoto**, implementato secondo un modello di tipo client-server, *che permette ad un utente attestato ad una certa macchina di stabilire una connessione TCP con un server di login che si trova su un'altra macchina*.
- La porta che utilizza è la numero 23



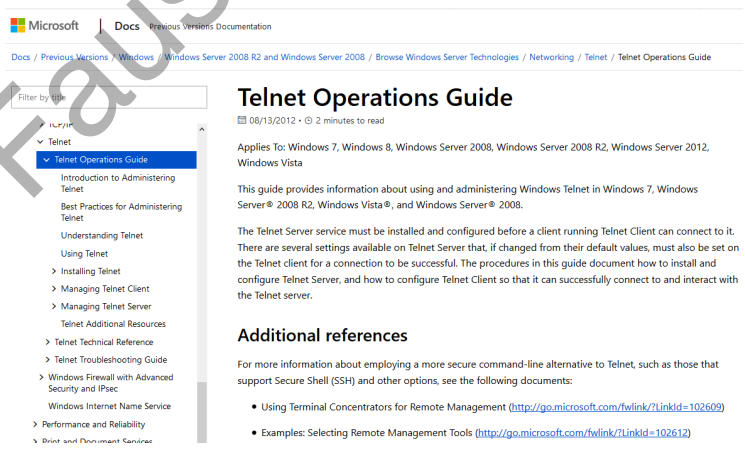
70

Cos'è "telnet"

- Telnet (il client) rilancia i caratteri battuti sulla tastiera dell'utente direttamente al calcolatore remoto come se essi fossero battuti su una tastiera direttamente connessa ad esso. Telnet (il server: telnetd) rimanda l'output della macchina remota indietro fino allo schermo dell'utente.
- Il servizio è definito trasparente perché dà l'apparenza che la tastiera e lo schermo dell'utente siano attaccati direttamente alla macchina remota.
- Telnet non molto sofisticato, ma **era** largamente diffuso.
- Il codice client di Telnet permette all'utente di specificare la macchina remota a cui ci si vuole connettere dando il suo nome di dominio oppure il suo indirizzo IP e specificando la porta



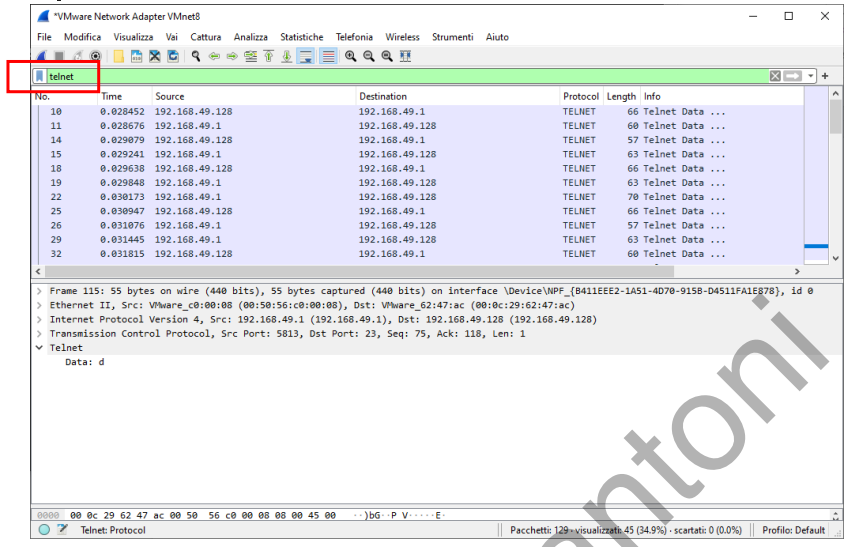
Telnet per Windows



https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2008-R2-and-2008/cc753164(v%3dws.10)

Reti di Elaboratori

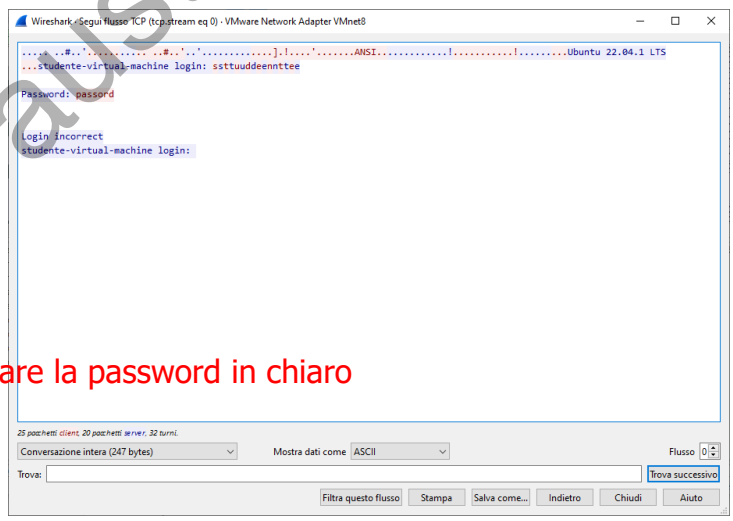
telnet - wireshark



73

Reti di Elaboratori

telnet - wireshark



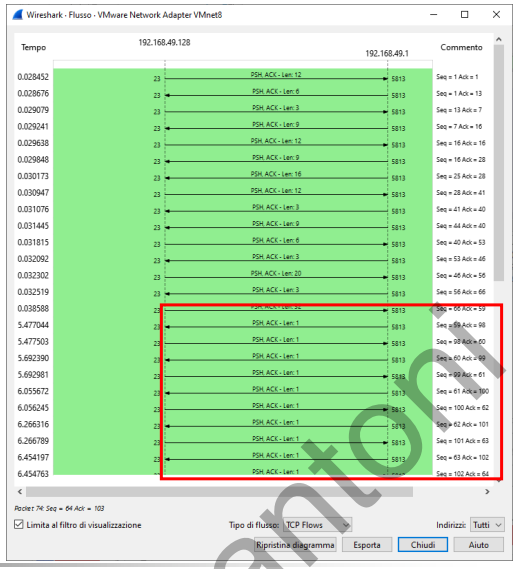
Verificare la password in chiaro

74

Reti di Elaboratori

telnet - wireshark

Notare la lunghezza di 1 byte nel pacchetto della digitazione di username e password



Reti di Elaboratori

Prova (client)

1. Telnet verso un Web server:

```
telnet <<mio indirizzo IP>> 80
```

Apre connessione TCP verso la porta 80 (default)

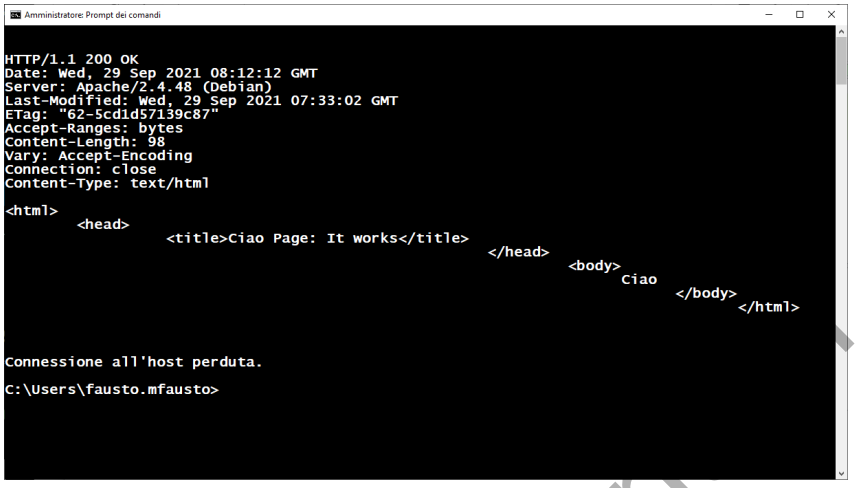
2. Si digita una richiesta http GET:

GET — Digitando ciò (carriage return due volte), si invia una richiesta GET al server http

3. Si osservi la risposta!

```
GET / HTTP/1.1
HOST: www.unicam.it
USER-AGENT: Fausto
```

Prova (client) - risposta



Commentare la risposta ottenuta

77

User Agent HTTP

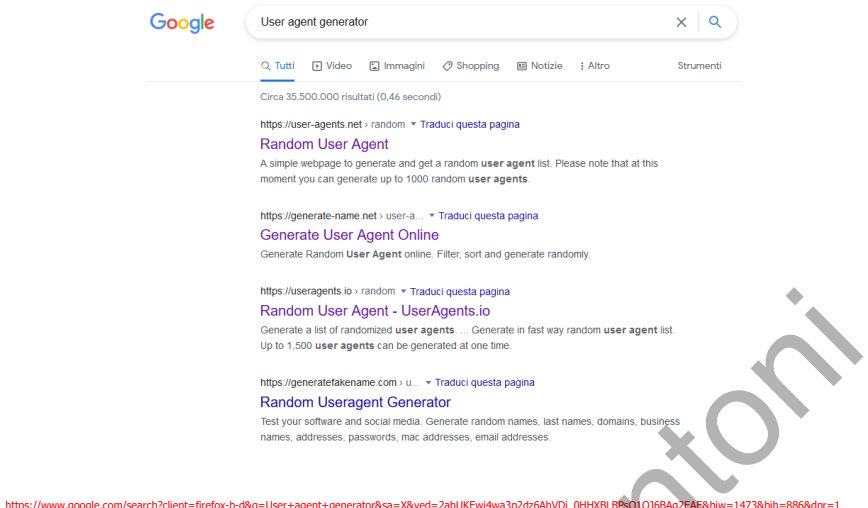
User-Agent: <product> / <product-version> <comment>

https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/User-Agent

- ✓ Android User Agents
- ✓ iPhone User Agents
- ✓ MS Windows User Agents
- ✓ Tablet User Agents
- ✓ Desktop User Agents
- ✓ Set Top Box User Agents
- ✓ Games Console User Agents
- ✓ Bots and Crawlers User Agents
- ✓ E-Readers User Agents

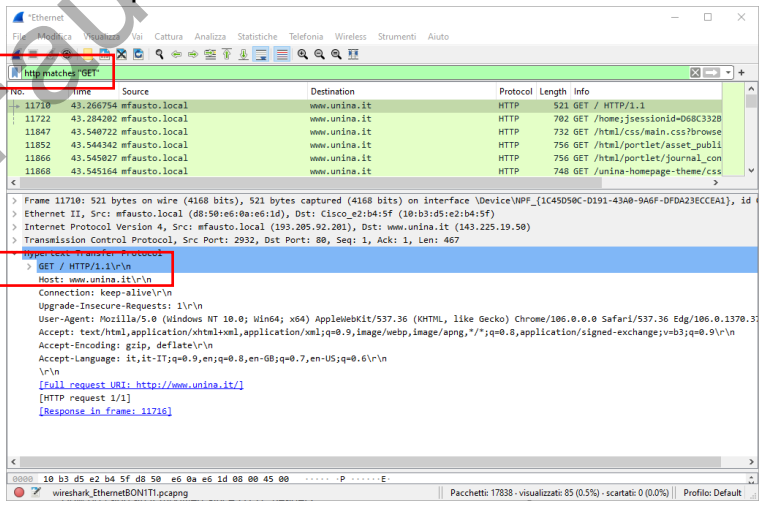
78

User agent generator



https://www.google.com/search?client=firefox-b-d&q=User+agent+generator&sa=X&ved=2ahUKEwid4a3p2dz6AhVDi_0HHXBLBPsQ1QJ6BAg2EAE&biw=1473&bih=886&dpr=1

http matches "GET"



Reti di Elaboratori

Prova (client) - risposta

```
telnet <<mio indirizzo IP>> 80
```

- digitare e commentare:
  - abcdef
  - GET /index.html HTTP/1.0 (?)
  - HEAD
  - HEAD /index.html HTTP/1.0 (?)
  - POST
  - GET /index.html HTTP/1.1 (?)

Esercitazione con

internet reti sicurezza

Esercitazioni

81

Reti di Elaboratori

I cookies

- **HTTP è stateless**: non esiste nessuna struttura ulteriore alla connessione, e il server non è tenuto a mantenere informazioni su connessioni precedenti.
- Un cookie (non è del protocollo HTTP, è un'estensione di Netscape, proposta nell'RFC 2109 e poi ancora RFC 2965 e infine in RFC 6265) è **una breve informazione scambiata tra il server ed il client**.
- Il **client mantiene lo stato di precedenti connessioni**, e lo manda al server di pertinenza ogni volta che richiede un documento.
- Il termine *cookie* (anche *magic cookie*) in informatica indica un **blocco di dati opaco** (i.e.: non interpretabile) lasciato in consegna ad un richiedente per poter ristabilire in seguito il suo diritto alla risorsa richiesta (come il tagliando di una lavanderia)

82

HTTP State Management Mechanism

HTTP State Management Mechanism

Abstract

This document defines the HTTP Cookie and Set-Cookie header fields. These header fields can be used by HTTP servers to store state (called cookies) at HTTP user agents, letting the servers maintain a stateful session over the mostly stateless HTTP protocol. Although cookies have many historical infelicities that degrade their security and privacy, the Cookie and Set-Cookie header fields are widely used on the Internet. This document obsoletes RFC 2965.

https://www.rfc-editor.org/rfc/rfc6265

83

Cookie [RFC2109] obsoleto

4 componenti

1. Una riga di intestazione nel messaggio di risposta HTTP
2. Una riga di intestazione nel messaggio di richiesta HTTP
3. Un file cookie memorizzato sul sistema client dell'utenet e gestito dal browser
4. Un database sul sito (server)

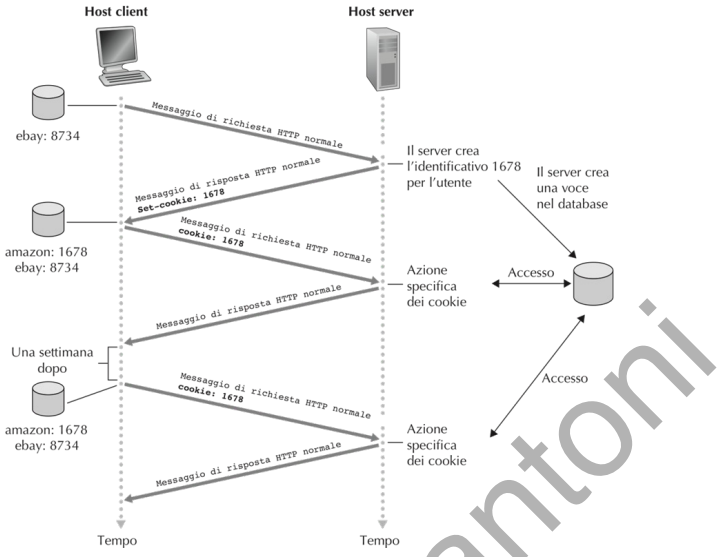
esempio

```
PI
P=0403710006620D1105077461006865 6A161D647E721804797E7A027808557F7F7C6A0205
punto-informatico.it/
1536
3164909568
29887167
1043606064
29813841
*
```


84

Reti di Elaboratori

Interazione client-server: i cookie



85

Reti di Elaboratori

Using the Set-Cookie header


```
   == Server -> User Agent ==

   Set-Cookie: SID=31d4d96e407aad42

   == User Agent -> Server ==

   Cookie: SID=31d4d96e407aad42
```

The server can alter the default scope of the cookie using the Path and Domain attributes. For example, the server can instruct the user agent to return the cookie to every path and every subdomain of example.com.

```
   == Server -> User Agent ==

   Set-Cookie: SID=31d4d96e407aad42; Path=/; Domain=example.com

   == User Agent -> Server ==

   Cookie: SID=31d4d96e407aad42
```

As shown in the next example, the server can store multiple cookies at the user agent. For example, the server can store a session identifier as well as the user's preferred language by returning two Set-Cookie header fields. Notice that the server uses the Secure and HttpOnly attributes to provide additional security protections for the more sensitive session identifier (see Section 4.1.2.)

```
   == Server -> User Agent ==

   Set-Cookie: SID=31d4d96e407aad42; Path=/; Secure; HttpOnly
   Set-Cookie: lang=en-US; Path=/; Domain=example.com

   == User Agent -> Server ==

   Cookie: SID=31d4d96e407aad42; lang=en-US
```

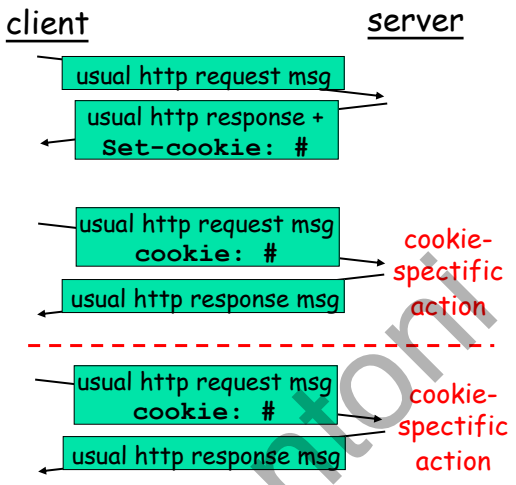
https://datatracker.ietf.org/doc/html/rfc6265#section-3.1

86

Reti di Elaboratori

Cookie [RFC2109]

- Il server invia un "cookie" al client con la richiesta
  `Set-cookie: 1678`
- Il client presenta il cookie in accessi successivi
  `cookie: 1678`
- Il server controlla il cookie presentato
  - Autenticazione
  - Traccia delle preferenze dell'utente
  - Profilzione utente



87

Reti di Elaboratori

Come funzionano i Cookie

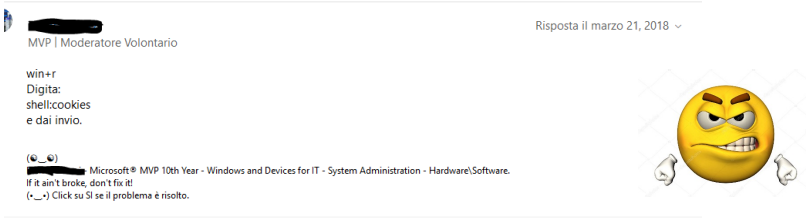
- Un cookie, dall'inglese "biscotto", è un piccolo file di testo che viene ricevuto dal browser durante la navigazione e viene memorizzato in un opportuno
- **repository** →

`C:\Users\fausto.mfausto\AppData\Local\Microsoft\Windows\INetCookies`

Windows key and R together. Type ***shell:cookies*** and click OK.

http://windows.microsoft.com/it-it/windows/view-temporary-internet-files#1TC=windows-7

https://www.thewindowsclub.com/cookies-folder-location-windows



88

Reti di Elaboratori

Visualizzare i cookies

Google Chrome stores all cookies in a single file called *Cookies*.
The file is located at the following path:
C:\Users\Your User Name\AppData\Local\Google\Chrome\User Data\Default
C:\Users\fausto.mfausto\AppData\Local\Google\Chrome\User Data\Default
Mozilla Firefox stores all the cookies, from all the websites that you visit, in a single file called *cookies.sqlite*.
You can find it in your Firefox profile folder, at the following path:
C:\Users\Your User Name\AppData\Roaming\Mozilla\Firefox\Profiles
C:\Users\fausto.mfausto\AppData\Roaming\Mozilla\Firefox\Profiles
Opera keeps all the cookies in a single file called *Cookies*, just like *Google Chrome*.
The *Cookies* file is located at the following path:
C:\Users\Your User Name\AppData\Roaming\Opera Software\Opera Stable
If you are running a Windows 10 version before 1709 Fall Creators Update,
you can find the cookies files created by **Internet Explorer** at this path:
C:\Users\Your User Name\AppData\Local\Microsoft\Windows\INetCookies
In the latest versions of Windows 10, **Microsoft Edge does not have a specific cookies file.**
If you have an older version of Windows 10, before 1709 Fall Creators Update, you might be able to find one or more cookies files in these locations:
"C:\Users\Your User Name\AppData\Local\Packages\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\AC\MicrosoftEdge\Cookies"
"C:\Users\Your User Name\AppData\Local\Packages\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\AC\#!001\MicrosoftEdge\Cookies"
"C:\Users\Your User Name\AppData\Local\Packages\Microsoft.MicrosoftEdge_8wekyb3d8bbwe\AC\#!002\MicrosoftEdge\Cookies"

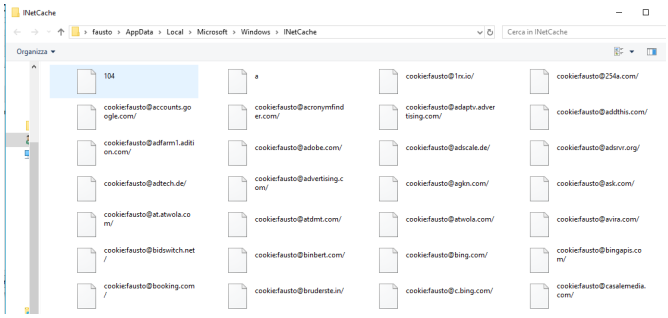
https://www.digitalcitizen.life/cookies-location-windows-10

89

Reti di Elaboratori

Come funzionano i Cookie

- Non tutti i siti rilasciano cookies.
- Facoltativo e prettamente applicativo.
- Poterlo rileggere in un secondo tempo (comportamento applicativo di una certa complessità).
- Reale pericolo per la privacy



90

Come funzionano i Cookie

- E' possibile rendere facoltativa l'accettazione dei cookie da parte dei browser web.
- All'interno di un cookie sono memorizzate informazioni ad uso e consumo del server web che ha chiesto al browser la memorizzazione del cookie stesso.
- La struttura di un cookie è assimilabile ad una struttura XML

```
<COOKIE
        NOME="name"
        VALORE="text"
        SCADENZA="period"
        SICURO="Yes/No"
        PERCORSO="urls"
        DOMINIO=".domain.com" >
```

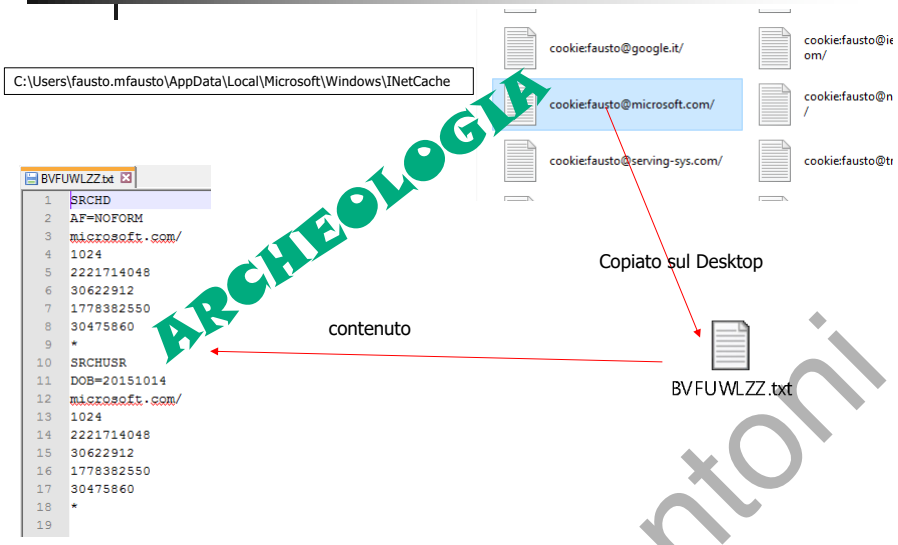
- Notate che quasi tutte le informazioni siano state memorizzate "criptate", a maggior tutela della privacy dell'utente che ha memorizzato il cookie.
- Spesso il formato SQLite

Set-Cookie Syntax

```
RFC 6265                HTTP State Management Mechanism          April 2011

set-cookie-header = "Set-Cookie:" SP set-cookie-string
set-cookie-string = cookie-pair *( ";" SP cookie-av )
cookie-pair       = cookie-name "=" cookie-value
cookie-name       = token
cookie-value      = *cookie-octet / ( DQUOTE *cookie-octet DQUOTE )
cookie-octet      = %x21 / %x23-2B / %x2D-3A / %x3C-5B / %x5D-7E
                      ; US-ASCII characters excluding CTLs,
                      ; whitespace DQUOTE, comma, semicolon,
                      ; and backslash
token             = <token, defined in [RFC2616], Section 2.2>

cookie-av         = expires-av / max-age-av / domain-av /
                    path-av / secure-av / httponly-av /
                    extension-av
expires-av        = "Expires=" sane-cookie-date
sane-cookie-date  = <rfc1123-date, defined in [RFC2616], Section 3.3.1>
max-age-av        = "Max-Age=" non-zero-digit *DIGIT
                      ; In practice, both expires-av and max-age-av
                      ; are limited to dates representable by the
                      ; user agent.
non-zero-digit    = %x31-39
                      ; digits 1 through 9
domain-av         = "Domain=" domain-value
domain-value      = <subdomain>
                      ; defined in [RFC1034], Section 3.5, as
                      ; enhanced by [RFC1123], Section 2.1
path-av           = "Path=" path-value
path-value        = <any CHAR except CTLs or ";">
secure-av         = "Secure"
httponly-av       = "HttpOnly"
extension-av      = <any CHAR except CTLs or ";">
```

https://www.rfc-editor.org/rfc/rfc6265.html#page-13

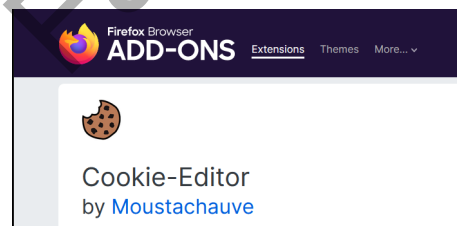
Esempio – non funziona

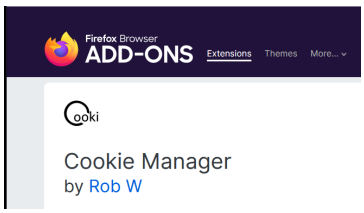


93

Firefox extension

Per chi fa le pagine web





https://addons.mozilla.org/en-US/firefox/search/?q=cookie

94

Reti di Elaboratori

Come funzionano i Cookie

- Un cookie è un file di testo. Non può fare nulla se non trasportare informazioni.
- Il server, tendenzialmente, utilizzerà il cookie come collegamento tra il client (e quindi l'utilizzatore a tutti gli effetti) ed un profilo personalizzato che si sarà creato.
- Ad ogni altro accesso il server sarà in grado di "riconoscerci" utilizzando il cookie come *chiave* per individuare il *profilo memorizzato* sul server stesso e sarà in grado, di arricchire il profilo *raccogliendo informazioni sulla nostra navigazione* all'interno del sito stesso.
- Perché si utilizzano i cookies?
  - profilo personalizzato all'interno dell'applicazione, ad esempio la lingua utilizzata e le impostazioni di visualizzazione.
  - autenticazione, ad esempio in un'applicazione di webmail, non sarà necessario inserire username e password ogni volta che si cerca di accedere al servizio. (sicurezza=0)

95

Reti di Elaboratori

I third party cookie

- Un uso subdolo (ma alcuni lo giustificano) dei cookie è l'inserimento di cookie nei banner e nelle pubblicità.
- Questo permette ad un'agenzia di banner di seguire la navigazione di un utente attraverso tutti i siti a cui fornisce banner, e quindi fornire una profilazione più precisa del navigatore, con effetti discutibili sulla sua privacy.
- RFC 2965 esplicitamente proibisce questo tipo di comportamento, che è però largamente ignorato dai produttori di browser e dai fornitori di banner.
- Si aggiunga che molte versioni di browser hanno bug che permettono a codice Javascript malevolo, nascosto dentro alle pagine dei browser, di "*sniffare*" i contenuti dei cookie destinati ad altri domini.

96

Reti di Elaboratori

pubblicità comportamentale

la pubblicità comportamentale è una pratica che si basa sull'attività di navigazione in rete e permette ai brand di inviare messaggi pubblicitari agli utenti della rete in funzione dei loro interessi.

Your Online Choices
a guide to online behavioural advertising

https://www.youronlinechoices.com/it/

Benvenuti nella guida sulla pubblicità comportamentale e la privacy online. In questo sito web troverete informazioni su come funziona la pubblicità comportamentale e molte informazioni sui cookie oltre alle tappe da seguire per proteggere la privacy su internet.

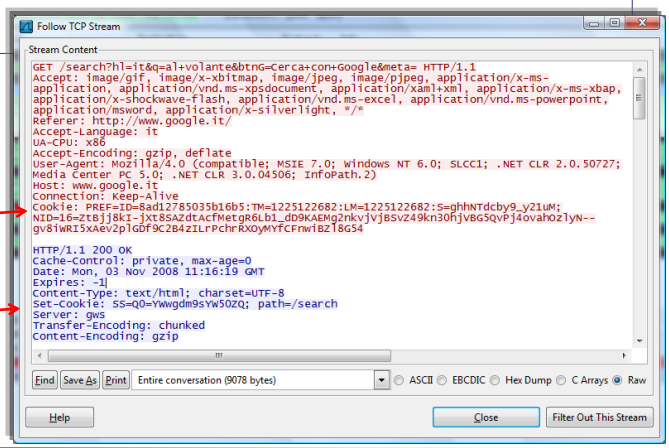
https://www.youronlinechoices.com/it/le-tue-scelte

97

Reti di Elaboratori

Esercitazione: cookie

- Utilizzando lo sniffer
  - Aprire una sessione http
  - Verificare che la sessione "contenga" un cookie
  - Verificare lo scambio di messaggi tra client e server
- Trovare il cookie
- Visualizzare il cookie



98

Reti di Elaboratori

Filtro http

First some theoretical backgraound:

- **Cookies:** Cookies are a way for a HTTP server to store information on the client, which will be presented back to the server in following requests. The purpose is to have a way of maintaining information between the client and the server to simulate a session (http by itself has no notion of sessions, it's just a way to exchange objects).
- **Query Strings:** A query string is a way for the client to submit (dynamic) data to the server. Mostly this is done by having a FORM on a webpage that can be filled in and when it's submitted the filled in values are transferred to the server in the "Query String". This can be done with the GET method, in which the query string will be added to the requested URL. Or it can be done with the POST method in which the query string will be sent as HTTP data, after the HTTP headers.

Both Cookies and Query Strings are completely independent of each other, but are widely used together. The way they are used depends on the way the web application has been written.

To filter all requests that contain a cookie, use:

```
http.cookie
http.cookie contains <cookiename>
```

To filter for query strings:

```
http.request.uri contains "?" or http.request.method=="POST"
```

This of course only works with HTTP as HTTPS traffic is encrypted. However, if you do have access to the private key used on the HTTPS server, you are able to decrypt the HTTPS traffic which makes the HTTP traffic inside the HTTPS traffic visible.

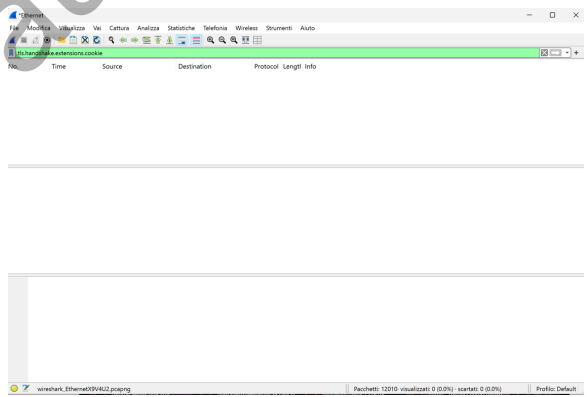
https://osqa-ask.wireshark.org/questions/751/cookie-and-query-strings/

99

Reti di Elaboratori

Filtro per cookie in TLS

tls.handshake.extensions.cookie	Cookie	Byte sequence
tls.handshake.extensions.cookie_len	Cookie length	Unsigned integer (16 bits)



100

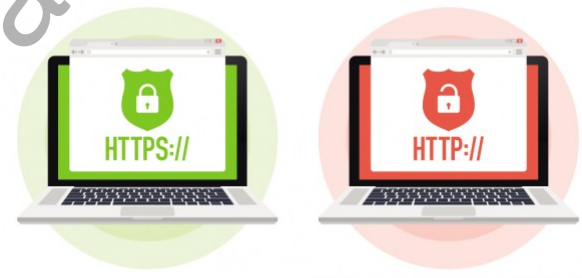
https://sqlitebrowser.org

DB Browser for SQLite

The Official home of the DB Browser for SQLite

https://sqlitebrowser.org/dl/

101



102

Modelli di sicurezza

Ci sono due modi per fornire un trasporto sicuro (non intercettabile):

- **Usare un'infrastruttura di trasporto sicura**
  - Il protocollo non cambia, ma ogni pacchetto trasmesso nello scambio di informazioni viene gestito in maniera sicura dal protocollo di trasporto
- **Usare un protocollo sicuro a livello applicazione**
  - Si usa un protocollo anche diverso, che si occupa di gestire la trasmissione delle informazioni.

HTTPS (RFC 2818)
Introdotto da Netscape, trasmette i dati in HTTP semplice su un protocollo di trasporto (SSL) che crittografa tutti i pacchetti.
Il server ascolta su una porta diversa (per default la porta 443), e si usa uno schema di URI diverso (introdotto da **https://**)

S-HTTP (RFC 2660)
Poco diffuso, incapsula richieste e risposte HTTP in un messaggio crittografato, o secondo un formato MIME apposito (MIME Object Security Services, MOSS), o un formato terzo (Cryptographic Message Syntax, CMS).
E' più efficiente ma più complesso.

Transport Layer Security TLS

Transport Layer Security TLS

Transport Layer Security, o TLS, è un protocollo di sicurezza ampiamente diffuso progettato per semplificare **la privacy e la sicurezza dei dati per le comunicazioni su Internet**.

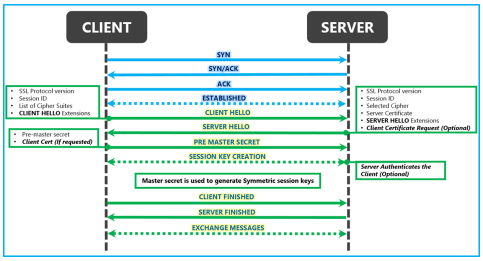
L'uso principale di TLS è la crittografia della comunicazione tra applicazioni Web e server

TLS può essere utilizzato anche per crittografare altre comunicazioni come e-mail, messaggistica e Voice over IP (VoIP)

TLS è stato proposto dalla Internet Engineering Task Force (IETF)

La versione più recente è TLS 1.3, pubblicata nel 20

Cos'è un handshake TLS?



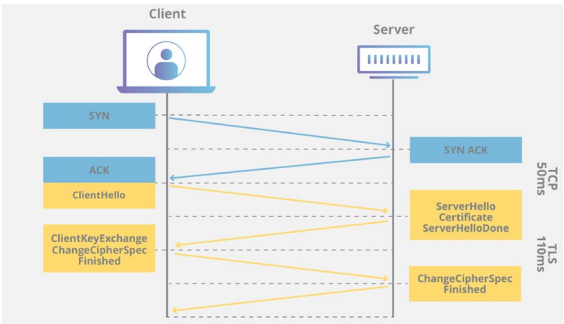
TLS è un **protocollo di crittografia** progettato per proteggere le comunicazioni Internet.
Un **handshake TLS** è il processo che avvia una sessione di comunicazione che utilizza la crittografia TLS.
Durante un handshake TLS, le due parti comunicanti si **scambiano messaggi per riconoscersi, verificarsi a vicenda, stabilire gli algoritmi di crittografia** che utilizzeranno e concordare le chiavi di sessione.

https://www.cloudflare.com/it-it/learning/ssl/what-happens-in-a-tls-handshake/
https://www.evemilano.com/tls-session-resumption/

handshake TLS

Un handshake TLS si verifica ogni volta che un utente accede a un sito Web tramite HTTPS e il browser inizia prima a interrogare il server di origine del sito Web.

Gli handshake TLS si verificano dopo che una connessione TCP è stata aperta tramite un handshake TCP.



Reti di Elaboratori

Quali sono le fasi di un handshake TLS

1. **Il messaggio 'client hello':** il client inizia l'handshake inviando un messaggio "hello" al server. Il messaggio includerà la versione TLS supportata dal client, le suite di crittografia supportate e una stringa di byte casuali nota come "casuale client".
2. **Il messaggio 'server hello':** in risposta al messaggio di saluto del client, il server invia un messaggio contenente il certificato SSL del server, la suite di cifratura scelta dal server e la stringa "casuale server", un'altra stringa casuale di byte generata dal server.
3. **Autenticazione:** il client verifica il certificato SSL del server con l'autorità di certificazione che lo ha emesso. Ciò conferma che il server è chi dice di essere e che il client sta interagendo con l'effettivo proprietario del dominio.
4. **Il segreto premaster:** il client invia un'altra stringa casuale di byte, il "segreto premaster". Il segreto premaster è crittografato con la chiave pubblica e può essere decrittografato solo con la chiave privata dal server. (Il client ottiene la chiave pubblica dal certificato SSL del server.)
5. **Chiave privata utilizzata:** il server decritta il segreto premaster.
6. **Chiavi di sessione create:** sia il client che il server generano chiavi di sessione dalla stringa casuale client, dalla stringa casuale server e dal segreto premaster. Dovrebbero arrivare agli stessi risultati.
7. **Il client è pronto:** il client invia un messaggio "terminato" che viene crittografato con una chiave di sessione.
8. **Il server è pronto:** il server invia un messaggio "terminato" crittografato con una chiave di sessione.
9. **Crittografia simmetrica sicura ottenuta**: l'handshake è stato completato e la comunicazione continua utilizzando le chiavi di sessione.

Reti di Elaboratori

Definizione di una sessione sicura tramite TLS

Il protocollo TLS Handshake prevede la procedura seguente:

1. Il client invia un messaggio "Client hello" al server, insieme al valore casuale del client e ai pacchetti di crittografia supportati.
2. Il server risponde inviando un messaggio "Server hello" al client, insieme al valore casuale del server.
3. Il server invia il certificato al client per l'autenticazione e può richiedere un certificato dal client. Il server invia il messaggio "Server hello done".
4. Se il server ha richiesto un certificato dal client, il client lo invia.
5. Il client crea un segreto pre-master casuale e lo crittografa con la chiave pubblica dal certificato del server, inviando il segreto pre-master crittografato al server.
6. Il server riceve il segreto pre-master. Il server e il client generano le chiavi master secret e sessione in base al segreto pre-master.
7. Il client invia una notifica "Modifica spec di crittografia" al server per indicare che il client inizierà a usare le nuove chiavi di sessione per l'hashing e la crittografia dei messaggi. Il client invia anche un messaggio "Client completato".
8. Il server riceve "Cambia spec di crittografia" e commuta lo stato di sicurezza del livello record alla crittografia simmetrica usando le chiavi di sessione. Il server invia un messaggio "Server completato" al client.
9. Il client e il server possono ora scambiare i dati dell'applicazione tramite il canale protetto stabilito. Tutti i messaggi inviati dal client al server e dal server al client vengono crittografati usando la chiave di sessione.

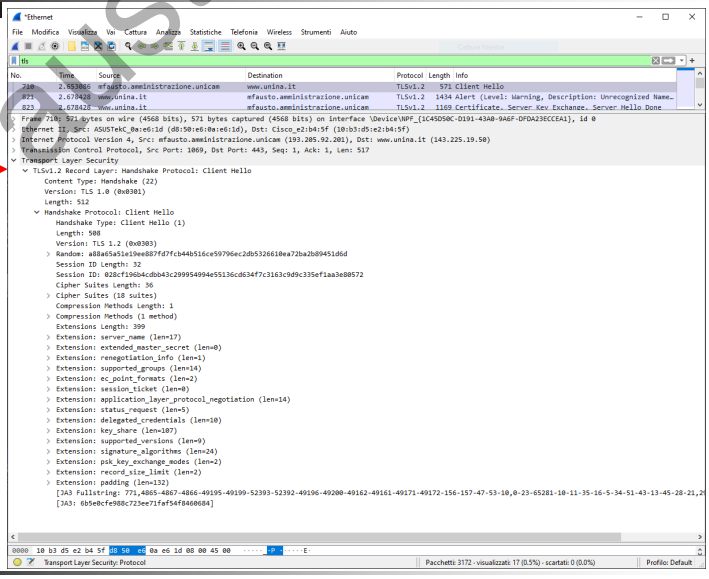
Sessione https

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	mfausto.amministrazione.unicam	antoine.unicam.it	TCP	66	2448 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PE
2	0.000300	antoine.unicam.it	mfausto.amministrazione.unicam	TCP	66	443 → 2448 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SAC
3	0.000362	mfausto.amministrazione.unicam	antoine.unicam.it	TCP	54	2448 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
4	0.000593	mfausto.amministrazione.unicam	antoine.unicam.it	TLSv1.3	571	Client Hello
5	0.000815	antoine.unicam.it	mfausto.amministrazione.unicam	TCP	60	443 → 2448 [ACK] Seq=1 Ack=518 Win=64128 Len=0
6	0.004151	antoine.unicam.it	mfausto.amministrazione.unicam	TLSv1.3	1514	Server Hello, Change Cipher Spec, Application Data
7	0.004151	antoine.unicam.it	mfausto.amministrazione.unicam	TLSv1.3	1044	Application Data, Application Data, Application Data
8	0.004216	mfausto.amministrazione.unicam	antoine.unicam.it	TCP	54	2448 → 443 [ACK] Seq=518 Ack=2451 Win=2102272 Len=0
9	0.024443	mfausto.amministrazione.unicam	antoine.unicam.it	TLSv1.3	118	Change Cipher Spec, Application Data
10	0.024994	antoine.unicam.it	mfausto.amministrazione.unicam	TLSv1.3	357	Application Data
11	0.025140	antoine.unicam.it	mfausto.amministrazione.unicam	TLSv1.3	357	Application Data
12	0.025168	mfausto.amministrazione.unicam	antoine.unicam.it	TCP	54	2448 → 443 [ACK] Seq=582 Ack=3057 Win=2101760 Len=0
13	0.029904	mfausto.amministrazione.unicam	antoine.unicam.it	TCP	66	2449 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PE
14	0.030182	antoine.unicam.it	mfausto.amministrazione.unicam	TCP	66	443 → 2449 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460 SAC
15	0.030231	mfausto.amministrazione.unicam	antoine.unicam.it	TCP	54	2449 → 443 [ACK] Seq=1 Ack=1 Win=2102272 Len=0
16	0.030447	mfausto.amministrazione.unicam	antoine.unicam.it	TLSv1.3	672	Client Hello
17	0.030713	antoine.unicam.it	mfausto.amministrazione.unicam	TCP	60	443 → 2449 [ACK] Seq=1 Ack=619 Win=64128 Len=0
18	0.031828	antoine.unicam.it	mfausto.amministrazione.unicam	TLSv1.3	298	Server Hello, Change Cipher Spec, Application Data, Applicatio
19	0.032074	mfausto.amministrazione.unicam	antoine.unicam.it	TLSv1.3	118	Change Cipher Spec, Application Data
20	0.032500	antoine.unicam.it	mfausto.amministrazione.unicam	TLSv1.3	357	Application Data
21	0.084823	mfausto.amministrazione.unicam	antoine.unicam.it	TCP	54	2449 → 443 [ACK] Seq=683 Ack=548 Win=2101760 Len=0

Provare con il file

antoine_https.pcapng

Filtro: TLS



"tls.handshake.type == 1" for Client Hello
"tls.handshake.type == 2" for Server Hello

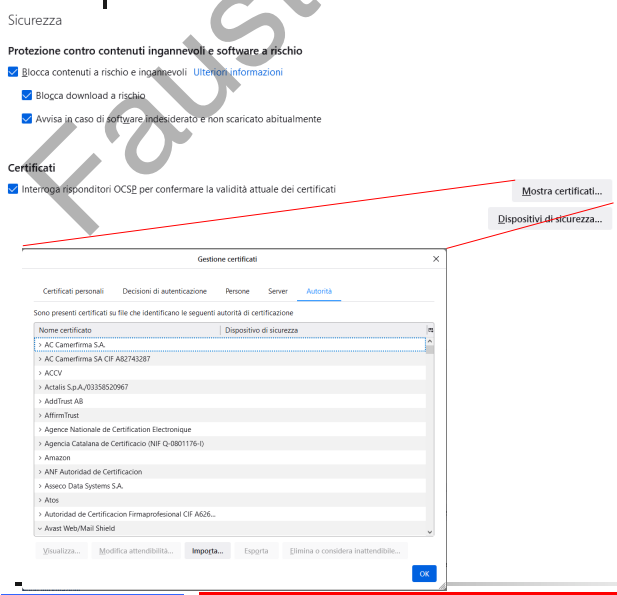
tls.handshake.type==1

No.	Time	Source	Destination	Protocol	Length	Info
138	2.398339	mfausto.amministrazione.unicam	wd-prod-ss-eu-west-2-fe.westeurope…	TLSv1.2	571	Client Hello
151	2.422994	mfausto.amministrazione.unicam	172.16.0.15	TLSv1.3	571	Client Hello
191	2.477768	mfausto.amministrazione.unicam	wd-prod-ss-eu-west-2-fe.westeurope…	TLSv1.2	571	Client Hello
249	2.547114	mfausto.amministrazione.unicam	mil04s44-in-f8.1e100.net	QUIC	1292	Initial, DCID=
254	2.547854	mfausto.amministrazione.unicam	172.16.0.15	TLSv1.3	654	Client Hello

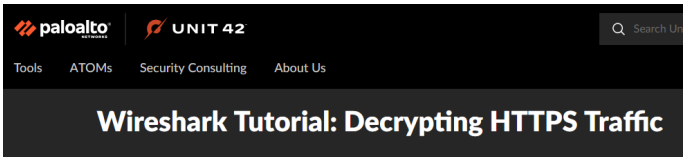
tls.handshake.type==2

No.	Time	Source	Destination	Protocol	Length	Info
159	2.433705	wd-prod-ss-eu-west-2-fe.westeurope…	mfausto.amministrazione.unicam	TLSv1.2	1266	Server Hello, Ce
165	2.435359	172.16.0.15	mfausto.amministrazione.unicam	TLSv1.3	1434	Server Hello, Ch
237	2.513741	wd-prod-ss-eu-west-2-fe.westeurope…	mfausto.amministrazione.unicam	TLSv1.2	1266	Server Hello, Ce
264	2.561090	172.16.0.15	mfausto.amministrazione.unicam	TLSv1.3	295	Server Hello, Ch
269	2.563766	mil04s44-in-f8.1e100.net	mfausto.amministrazione.unicam	QUIC	1292	Initial, SCID=b8

Certificati



Wireshark Tutorial: Decrypting HTTPS Traffic

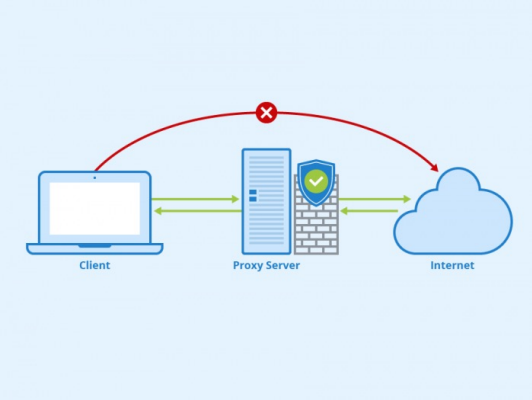


https://unit42.paloaltonetworks.com/wireshark-tutorial-decrypting-https-traffic/

113

Web Cache (proxy server)

Web Cache - Proxy server

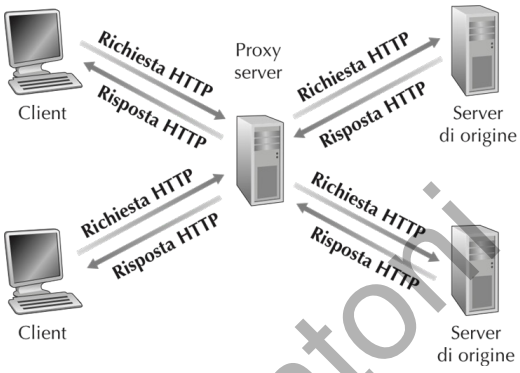


114

Web Cache (proxy server)

Obiettivo: rispondere alle richieste evitando di accedere al server remoto

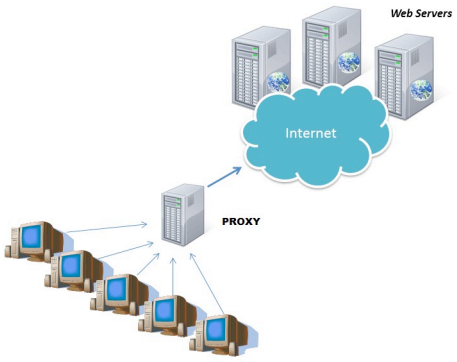
- L'utente configura il browser: accesso attraverso web cache
- Il client invia tutte le richieste al proxy
- La cache restituisce l'oggetto se presente
  - Altrimenti l'oggetto è richiesto prima al server e poi è restituito al client



Perché il Web Caching?

Assunzione: la cache è "vicina" al client (es., stessa rete locale)

- Tempo di risposta minore: la cache è "più vicina" al client
- Diminuisce il traffico verso server lontani
  - Il link di uscita della rete di un ISP istituzionale/locale è spesso un collo di bottiglia
- Filtraggio dei pacchetti/siti
  - sicurezza
- Accounting degli utenti
  - autenticazione (captive portal)
  - reportistica
  - monitor/controllo utenti (es. scuola)



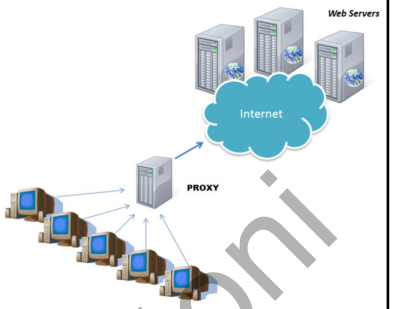
Reti di Elaboratori

Perché il Web Caching?

- Ritardo TOTALE = Ritardo sulla LAN + Ritardo sul LINK di accesso + Ritardo di Internet
- Esempio
  - dimensione media oggetti = 100 Kbit
  - richieste al secondo sulla rete Aziendale =15 (in media)
  - ritardo medio di Internet = 2 sec
- Intensità del traffico sulla LAN =

  (15 richieste/sec.) x (100 Kbit/richiesta) / (10 Mbit/sec.) = 0,15
- Intensità del traffico sul LINK =

  (15 richieste/sec.) x (100 Kbit/richiesta)/ (1,5 Mbit/sec.) = 1
- COSA FARE ?
  - Aumentare la velocità del LINK sino a 100 Mbit/sec (COSTOSA)
  - WEB CACHE →
  - 40% di richieste in media soddisfatte dalla cache
  - Con il 60% delle richieste l'intensità del traffico sul link passa da 1 a 0,6



117

Reti di Elaboratori

GET condizionale (conditional GET)

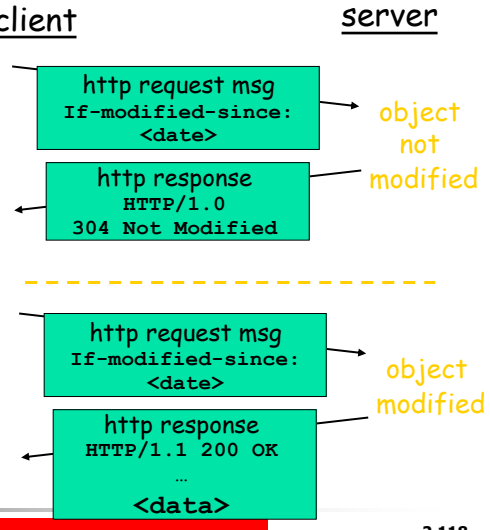
HTTP presenta un meccanismo che permette alla cache di verificare se i dati sono aggiornati

- Obiettivo: non inviare oggetti che il client ha già in cache
- client: data dell'oggetto memorizzato in cache

  `If-modified-since: <date>`

  (questa è la data della precedente GET)
- server: la risposta è vuota se l'oggetto in cache è aggiornato:

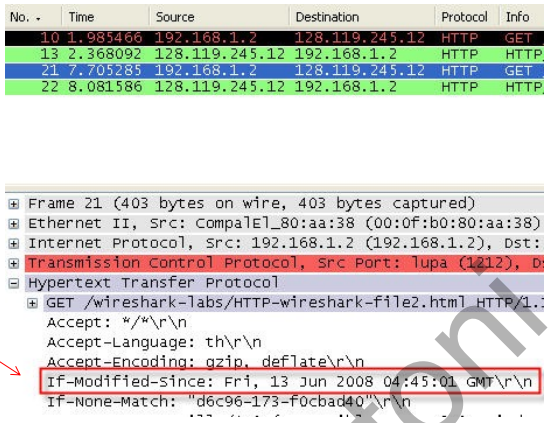
  `HTTP/1.0 304 Not Modified`



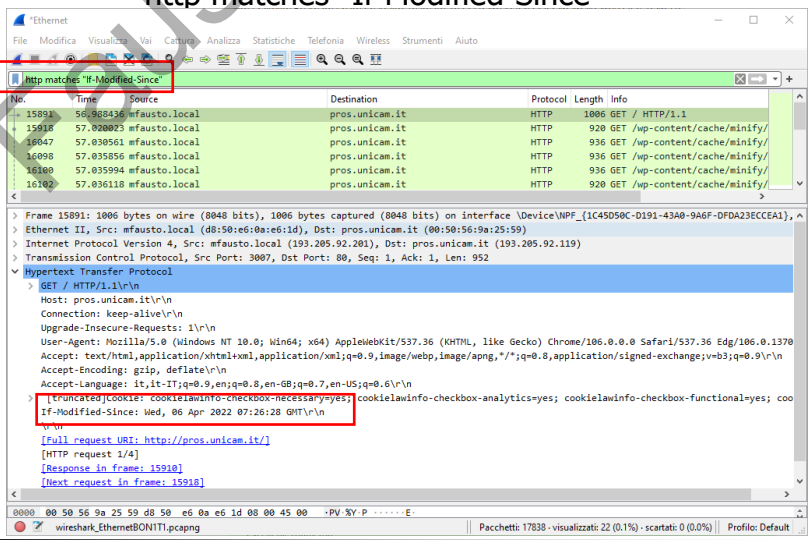
118

Esercitazione

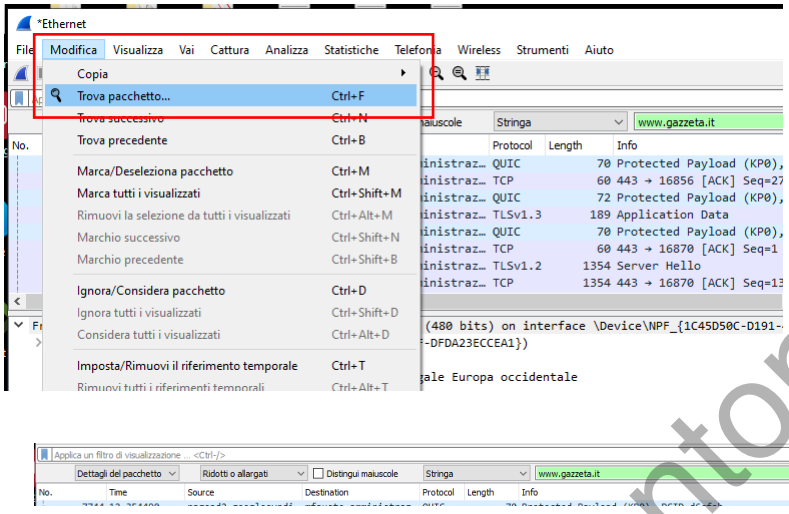
Esercitazione con



http matches "If-Modified-Since"



Cercare una stringa con wireshark



121

Esercitazioni

internet reti sicurezza

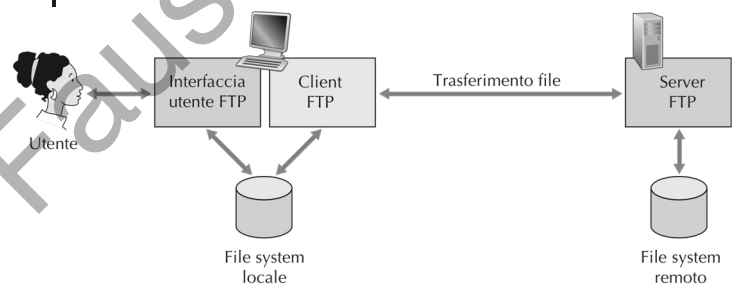
Esercitazioni

122

Protocolli di livello applicativo

- ✓ **FTP: file transfer protocol**
- ✓ **TFTP: Trivial File Transfer Protocol**
- ✓ **Posta elettronica**
  - ➢ **smtp**
  - ➢ **pop3 - imap**
- ✓ **DNS: Domain Name System**
- ✓ **SNMP: Simple Network Management Protocol**
- ✓ **Remote login: telnet - ssh**

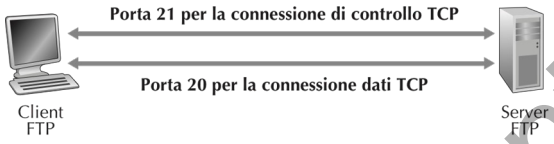
ftp: File transfer protocol



- Trasferimento file da/verso un host remoto
- Usa il modello client/server
  - *client:* parte che richiede il trasferimento (da/verso l' host remoto)
  - *server:* host remoto
- ftp: RFC 959
- ftp server: porta 21/20

ftp: connessioni controllo e dati separate

- Il client contatta il server sulla porta 21, specificando TCP come protocollo di trasporto
- due connessioni TCP parallele:
  - **controllo:** scambio di messaggi di controllo tra client e server.
    "controllo fuori banda"
  - **dati:** trasferimento dati da/verso il server
- Entrambi le connessioni aperte dal client
- Il server ftp mantiene info di "stato": directory corrente, autenticazione
- Una nuova connessione per ogni file trasferito



125

ftp comandi, risposte

Esempi di comandi:

- Inviati come testo ASCII mediante il canale di controllo
- `USER username`
- `PASS password`
- `LIST` richiede la lista dei file nella directory corrente (ls)
- `RETR <file>` richiede (get) un file
- `STOR <file>` scarica (put) un file sull' host remoto

Esempi di codici

- Codice di stato e frase (come in http)
- `331 Username OK, password required`
- `125 data connection already open; transfer starting`
- `425 Can't open data connection`
- `452 Error writing file`

126

Esempio ftp 1/2

```
C:\Documents and Settings\mf>ftp stesi.cs.unicam.it
Connesso a stesi.cs.unicam.it.
220 FTP Server ready.
Utente (stesi.cs.unicam.it:(none)): anonymous
331 Anonymous login ok, send your complete email address as your password.
Password:
230-
                *** Welcome to this anonymous ftp server! ***

     You are user 1 out of a maximum of 10 authorized anonymous logins.
     The current time here is Wed Oct 18 16:31:41 2006.
     If you experience any problems here, contact : root@localhost


230 Anonymous login ok, restrictions apply.
ftp> dir
200 PORT command successful
150 Opening ASCII mode data connection for file list
drwxr-xr-x   4 ftp      ftp          4096 Sep 14 14:42 .
drwxr-xr-x   4 ftp      ftp          4096 Sep 14 14:42 ..
drwxr-xr-x   2 ftp      ftp          4096 Aug 23 14:52 pub
d-wx-wx--x   2 ftp      ftp          4096 Aug 23 14:52 uploads
-rw-r--r--   1 ftp      ftp           224 Nov  9  2004 welcome.msg
226 Transfer complete.
ftp: 309 byte ricevuti in 0,00secondi 309000,00Kbyte/sec)
ftp>
```


127

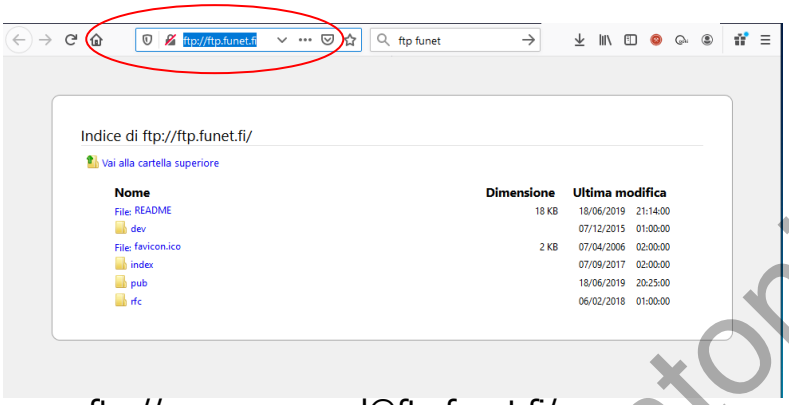
Esempio ftp 2/2

```
ftp> dir
200 PORT command successful
150 Opening ASCII mode data connection for file list
drwxr-xr-x   4 ftp      ftp          4096 Sep 14 14:42 .
drwxr-xr-x   4 ftp      ftp          4096 Sep 14 14:42 ..
drwxr-xr-x   2 ftp      ftp          4096 Aug 23 14:52 pub
d-wx-wx--x   2 ftp      ftp          4096 Aug 23 14:52 uploads
-rw-r--r--   1 ftp      ftp           224 Nov  9  2004 welcome.msg
226 Transfer complete.
ftp: 309 byte ricevuti in 0,00secondi 309000,00Kbyte/sec)
ftp> cd pub
250 CWD command successful
ftp> dir
200 PORT command successful
150 Opening ASCII mode data connection for file list
drwxr-xr-x   2 ftp      ftp          4096 Aug 23 14:52 .
drwxr-xr-x   4 ftp      ftp          4096 Sep 14 14:42 ..
226 Transfer complete.
ftp: 117 byte ricevuti in 0,00secondi 117000,00Kbyte/sec)
ftp> help
I comandi possono essere abbreviati. I comandi sono:

!               delete          literal         prompt          send
?               debug           ls              put             status
append          dir             mdelete         pwd             trace
ascii           disconnect      mdir            quit            type
bell            get             mget            quote           user
binary          glob            mkdir           recv            verbose
bye             hash            mls             remotehelp
cd              help            mput            rename
close           lcd             open            rmdir
ftp> bye

C:\Documents and Settings\mf>
```


128

Esempio ftp via browser

ftp://ftp.funet.fi/



ftp://user:password@ftp.funet.fi/

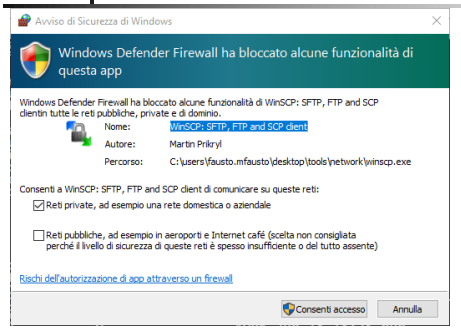
129

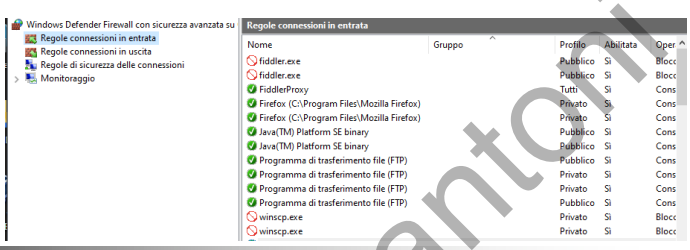
Esempio ftp via terminale

```
Prompt dei comandi - ftp ftp.funet.fi
Microsoft Windows [Versione 10.0.18362.418]
(c) 2019 Microsoft Corporation. Tutti i diritti sono riservati.

C:\Users\fausto.mfausto>ftp ftp.funet.fi
Connesso a ftp.funet.fi.
220---------- Welcome to Pure-FTPd [privsep] ----------
220-You are user number 56 of 1000 allowed.
220-Local time is now 11:17. Server port: 21.
220-Only anonymous FTP is allowed here
220-IPv6 connections are also welcome on this server.
220 You will be disconnected after 30 minutes of inactivity.
504 Unknown command
Utente (ftp.funet.fi:(none)): anonymous
331 Any password will work
Password:
230 Any password will work
ftp> dir
200 PORT command successful
150 Connecting to port 3445
-rw-rw-r--    1 108        42              18211 Jun 18 19:14 README
drwxr-xr-x    2 0          0                 125 Dec  7  2015 dev
-rwxr-xr-x    1 0          0                1078 Apr  7  2006 favicon.ico
drwxrwxr-x    3 50028      200             12288 Sep  7  2017 index
drwxr-xr-x    3 0          0                4096 Jun 18 18:25 pub
drwxrwxr-x    9 0          0             1835008 Feb  6  2018 rfc
226-Options: -l
226 6 matches total
ftp: 418 bytes received in 0.02secondi 26.13Kbyte/sec)
ftp>
```


130

Windows Firewall





131

How to configure Windows firewall from the command line

Start / Stop firewall service

```
netsh firewall set opmode DISABLE
netsh firewall set opmode ENABLE
```

Allow/Deny Ports

```
netsh firewall add portopening TCP _port_number_ _name_ DISABLE ALL
netsh firewall add portopening TCP 3264 CCMAIL DISABLE ALL

netsh firewall add portopening TCP _port_number_ _name_ ENABLE ALL
netsh firewall add portopening TCP 8443 PLESK-ADMIN ENABLE ALL
```

Programs to not allow TCP/UDP Socket Connections

```
netsh firewall add allowedprogram _path_ _name_ DISABLE ALL
netsh firewall add allowedprogram C:\Windows\System32\ftp.exe FTP DISABLE ALL
```

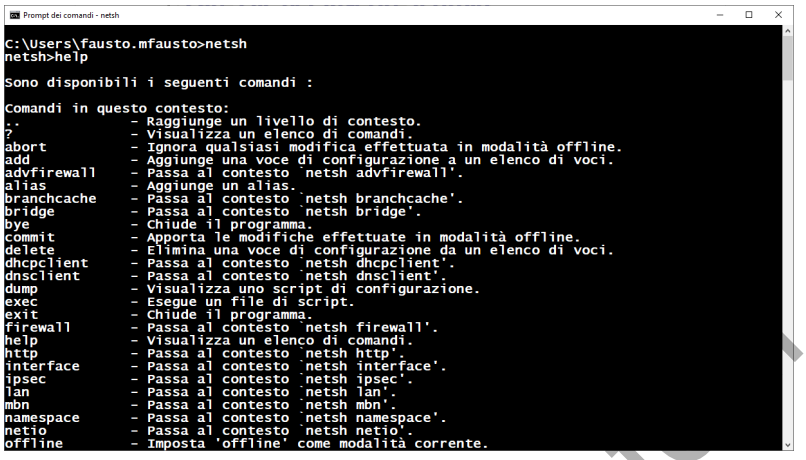
Allow/Deny Desktop Popup

```
netsh firewall set notifications DISABLE
```

https://www.simplified.guide/windows/firewall-command

132

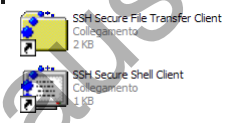
netsh



https://docs.microsoft.com/it-it/windows-server/networking/technologies/netsh/netsh-contexts

133

Altri sistemi client ftp



https://shareware.unc.edu/

PuTTY

http://en.wikipedia.org/wiki/PuTTY
http://www.chiark.greenend.org.uk/~sgtatham/putty/

WinSCP

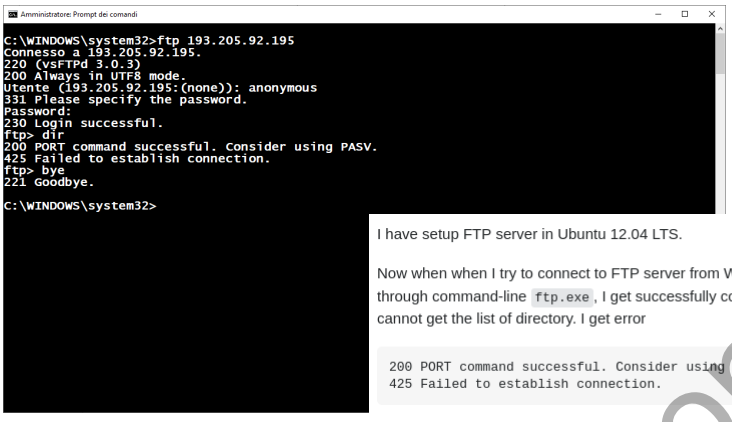
http://winscp.net/eng/docs/free_ssh_client_for_windows

https://filezilla-project.org/

https://osxdaily.com/2018/08/07/get-install-ftp-mac-os/

134

ftp su Windows 10 - problem



I have setup FTP server in Ubuntu 12.04 LTS.

Now when when I try to connect to FTP server from Windows 7 through command-line `ftp.exe`, I get successfully connected but I cannot get the list of directory. I get error

```
200 PORT command successful. Consider using PASV.
425 Failed to establish connection.
```

https://stackoverflow.com/questions/19516263/200-port-command-successful-consider-using-pasv-425-failed-to-establish-connec

PASV o PORT

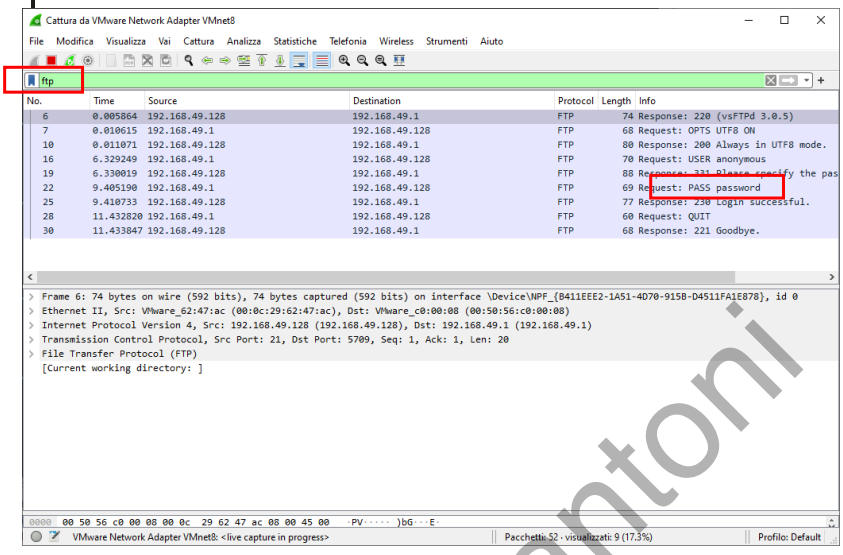
D : PASV o PORT , cosa sono e quale usare?

R : Molti degli utenti ftp non sanno cosa siano i comandi PASV (Passive) e PORT , ma naturalmente questo non è un difetto . Innanzi tutto occorre sapere **che ogni sessione FTP utilizza 2 connessioni TCP** , e quando ci si connette ad un server FTP quest'ultimo utilizza la prima , e quando si richiede una LISTA DIRECTORY o un TRASFERIMENTO FILE è richiesta la seconda connessione TCP , e questi 2 comandi sono utilizzati per COSTRUIRE (negoziare) la seconda connessione TCP .
Potrà sembrare strano il fatto che esistano 2 comandi per la connessione DATI , e non uno solo , e questa potrebbe essere una domanda interessante . Qualche tempo fa , il comando PORT era probabilmente sufficiente , tuttavia con l'introduzione in quantità di software FIREWALL e dispositivi NAT , il comando PORT è diventato inutile , così fu introdotto il **comando PASV , anche conosciuto come trasferimento dati compatibile FIREWALL** .
Vediamo in dettaglio qual è la differenza tra questi 2 comandi :
PORT : il cliente ftp CHIEDE al server ftp di collegarsi all'indirizzo IP del cliente ftp (con un numero di porta)
PASV : il server ftp CHIEDE al cliente ftp di collegarsi all'indirizzo IP del server ftp (con un numero di porta)
Così non è difficile realizzare quanto segue :
PORT fallirà : se il cliente ftp non conosce il suo reale indirizzo (succede quando il cliente è dietro ad un NAT)
PASV fallirà : se il server ftp non conosce il suo reale indirizzo (succede quando il server è dietro ad un NAT)
Dove è possibile vedere i comandi PORT/PASV e relative risposte ?! Bene , procuratevi un buon cliente FTP con una finestra che mostri i comandi :) , per esempio : FlashFXP , CuteFTP , Windows Commander e molti altri che non elenchiamo .. ma naturalmente , esistono alcuni clienti FTP che nascondono questi comandi/risposte, in modo che non si vedano affatto .
Adesso dovrebbe essere chiaro cosa configurare nel caso si incontrino problemi con PORT and PASV ..

http://www.raidenftpd.com/kb/it-kb000000010.html

Reti di Elaboratori

ftp - wireshark



137

Reti di Elaboratori

Protocolli di livello applicativo

- ✓ **FTP: file transfer protocol**
- ✓ **TFTP: Trivial File Transfer Protocol**
- ✓ **Posta elettronica**
  - ➢ **smtp**
  - ➢ **pop3 - imap**
- ✓ **DNS: Domain Name System**
- ✓ **SNMP: Simple Network Management Protocol**

138

Reti di Elaboratori

TFTP (Trivial File Transfer Protocol)

TFTP è un protocollo molto semplice utilizzato per trasferire file tra host utilizzando UDP.

definito nella **RFC 1350**

protocollo di trasporto **UDP** (*User Datagram Protocol*).

La porta sulla quale è in ascolto il server TFTP server è la **69**.

- molto semplice
- meno funzioni rispetto a FTP
- **non** può **leggere directory**
- **non** è provvisto di **autenticazione**
- utilizzo è limitato

TFTP viene solitamente usato per trasferire file tra un computer ed un altro dispositivo come router o switch in ambito LAN.

Reti di Elaboratori

TFTP (Trivial File Transfer Protocol)

- TFTP trasmette pacchetti con una **lunghezza fissa di 512 byte**.
- Un pacchetto avente una dimensione inferiore rappresenta l'ultimo pacchetto trasmesso.
- I pacchetti dati inviati vengono memorizzati in un buffer fino alla ricezione della avvenuta accettazione da parte dell'host remoto.
- In caso di mancata conferma della ricezione entro un determinato tempo di un pacchetto, quest'ultimo viene ritrasmesso.

La modalità di trasferimento dati di TFTP è di due tipi:
- **NETASCII** per i file di testo;
- **OCTET** per i file binari.

I pacchetti utilizzati durante una sessione TFTP sono di cinque tipi:
- **RR**: Read Request (*Richiesta di lettura*);
- **WR**: Write Request (*Richiesta di scrittura*);
- **DATA**: Dati;
- **ACK**: Acknowledgment (*Accettazione*);
- **ERR**: Errore;

Le fasi di una sessione TFTP:
1. Il client contatta il server inviando una pacchetto di tipo **RR** (*richiesta di lettura*) o **WR** (*richiesta di scrittura*);
2. Il server, se accetta la connessione, risponde inviando/ricevendo pacchetti **DATA** di 512 byte.
Per ogni pacchetto inviato/ricevuto regolarmente viene inviato/ricevuto un **ACK** altrimenti un **ERROR**;
3. I pacchetti vengono trasferiti finche la loro lunghezza non è inferiore a 512 byte;
4. Termine della connessione;

esercitazioni

internet reti sicurezza

Esercitazioni

141

Protocolli di livello applicativo

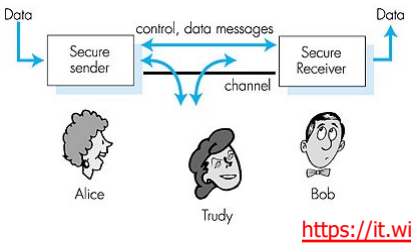
- ✓ **FTP: file transfer protocol**
- ✓ **TFTP: Trivial File Transfer Protocol**
- ✓ **Posta elettronica**
  - ➢ **smtp**
  - ➢ **pop3 - imap**
- ✓ **DNS: Domain Name System**
- ✓ **SNMP: Simple Network Management Protocol**
- ✓ **Remote login: telnet - ssh**

142

143

Alice, Bob e Trudy

- ✓ Bob e Alice (amanti!) vogliono comunicare “in modo sicuro”
- ✓ Trudy (l’intrusa) potrebbe intercettare, cancellare, aggiungere messaggi



https://it.wikipedia.org/wiki/Alice_e_Bob

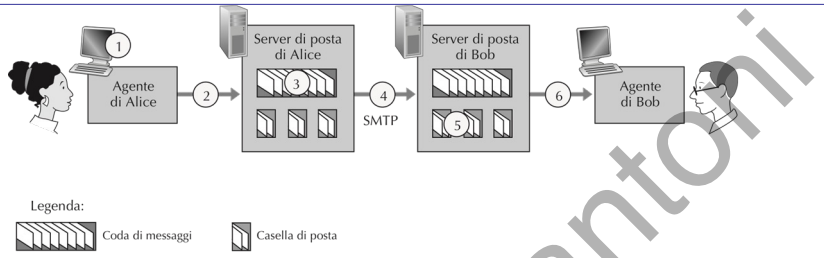
144

Reti di Elaboratori

Posta elettronica

Tre componenti principali:
- User agent
- Server di posta
- Simple Mail Transfer Protocol: SMTP

User Agent
- "Lettore di posta"
- Composizione e lettura di messaggi di posta
- Es.: Outlook, Thunderbird, Mailbird, Newton Mail, eM Client, Zimbra Desktop, Claws Mail
- Gestione dei messaggi in ingresso/uscita memorizzati sul server



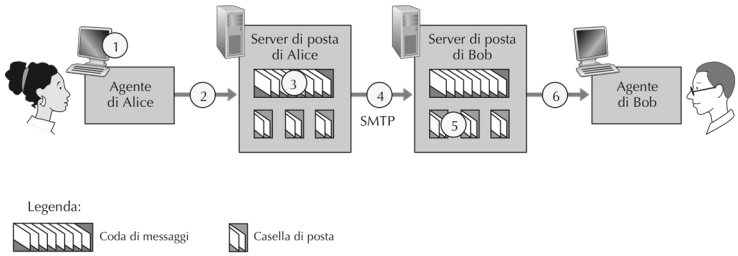
145

Reti di Elaboratori

Posta elettronica: mail server

Mail Server
- Mailbox contenente messaggi dell'utente (*non ancora letti*)
- Coda di messaggi in uscita (*non ancora spediti*)
- Protocol SMTP tra i mail server per il recapito dei messaggi
  - client: il server che invia il messaggio
  - server: server che riceve il messaggio

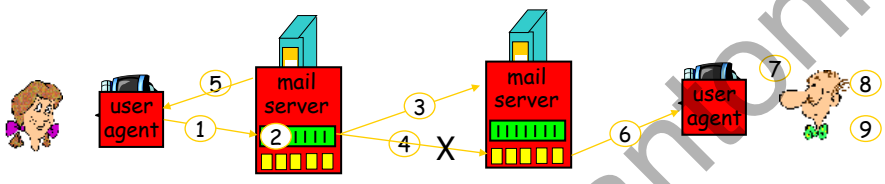
Nota: solo una distinzione di ruoli



146

Recapito di un messaggio di posta elettronica

1. Alice compone un messaggio e lo inoltra al suo Mail Server
2. Mail Server dispone il messaggio nella coda di messaggi in uscita
3. Mail Server di Alice apre una connessione SMTP con il Mail Server di Bob ed inoltra il messaggio
4. Se il contatto fallisce, l'invio è ripetuto ogni "n" Δtime (minuti)
5. Se l'invio fallisce dopo "m" Δtime (giorni), una mail di notifica inviata ad Alice
6. Mail Server di Bob riceve il messaggio dal Mail Server di Alice e lo salva nella Mailbox di Bob
7. Bob accede la propria Mailbox specificando Username e Password
8. Messaggi possono essere trasferiti dalla Mailbox all'host da cui Bob ha acceduto e/o lasciati sul server
9. Bob legge il messaggio di Alice



Posta elettronica: SMTP [RFC 821] (1982!)

- Usa TCP per il trasferimento affidabile dei messagi da client a server, porta 25
- Trasferimento diretto: da server a server, non si usano server intermedi di posta
- Tre fasi
  - Handshaking (saluto)
  - Trasferimento di uno o più messaggi (connessione permanente)
  - Chiusura
- Interazione mediante comandi/risposte
  - Comando: testo ASCII
  - Risposta: codice di stato e frase
- Attenzione: I messaggi devono essere comunque riportati in formato ASCII a 7 bit, anche dati multimediali

Esempio di interazione SMTP

```
S: 220 hamburger.edu
C: HELO crepes.fr
S: 250  Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Do you like ketchup?
C:   How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection
```


SMTP: considerazioni

- Connessioni TCP persistenti
- Richiede che il messaggio (header & corpo) sia in formato ascii 7-bit
- Alcune sequenze di caratteri non consentite (es., `CRLF.CRLF`). Conseguenza: il messaggio deve essere codificato
- Il server smtp usa `CRLF.CRLF` per determinare la fine del messaggio

Confronto con http

- http: pull protocol
- email: push protocol
- Entrambi usano un'interazione mediante comandi/risposta in testo ASCII e codici di stato
- http: ogni oggetto incapsulato nel messaggio di risposta
- smtp: tutti gli oggetti sono inviati mediante un unico messaggio (eventualmente in più parti)

Reti di Elaboratori

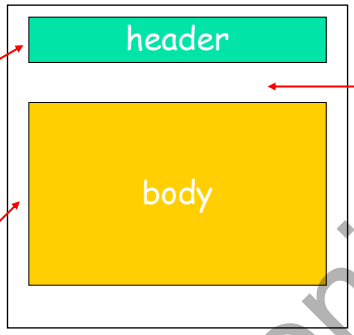
Formato dei messaggi

smtp: protocollo per lo scambio di messaggi di posta

RFC 822: standard per il formato dei messaggi inviati:

- header, es.,
  - To:
  - From:
  - Subject:

  Diversi dai comandi smtp!
- body
  - Il "messaggio" vero e proprio, solo caratteri ASCII



Reti di Elaboratori

Formato: estensioni multimediali

- MIME: multipupose internet mail extension, RFC 2045, 2056.
  - Dati Multimediali e di specifiche applicazioni
- Righe addizionali dell'header specificano il tipo del contenuto MIME

Versione MIME

Metodo di codifica

Tipo di dato multimediale, sottotipo, dichiarazione di parametri

Dati codificati

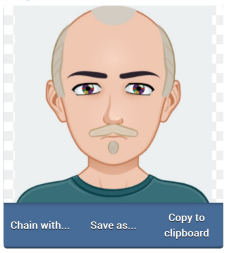
```
From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Picture of yummy crepe.
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Type: image/jpeg

base64 encoded data .....
.........................
......base64 encoded data
```

http://toolset.mrwebmaster.it/dev/base64-encoder-decoder.html

Image <-> base64

base64

png



https://onlinepngtools.com/convert-base64-to-png

https://www.base64-image.de/

153

Tipi MIME

`Content-Type: type/subtype; parameters`

Text
- Esempi di sottotipi:
  `plain, html`

Image
- Esempi di sottotipi :
  `jpeg, gif`

Audio
- Esempi di sottotipi :
  `basic (8-bit mu-law encoded)`
  `32kadpcm (32 kbps coding)`

Video
- Esempi di sottotipi :
  `mpeg, quicktime`

Application
- Dati che devono essere processati da un' applicazione prima di essere "visibili"
- Esempi di sottotipi :
  `msword, octet-stream`

https://tools.ietf.org/html/rfc1521

Elenco dei più comuni MIME (Multipurpose Internet Mail Extensions) types associati alle corrispondenti estensioni dei file.

154

Tipo Multipart

```
From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Picture of yummy crepe.
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary=98766789

--98766789
Content-Transfer-Encoding: quoted-printable
Content-Type: text/plain

Dear Bob,
Please find a picture of a crepe.
--98766789
Content-Transfer-Encoding: base64
Content-Type: image/jpeg

base64 encoded data .....
.........................
......base64 encoded data
--98766789--
```

limite

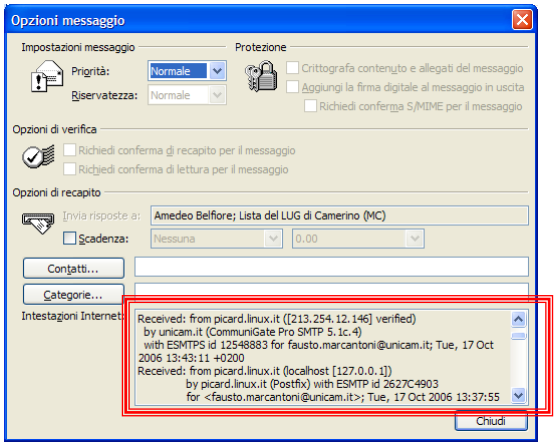
- E-mail contenenti più oggetti
- Boundary character: delimitano i messaggi
- per ogni oggetto
  - Content-Transfer-Encoding
  - Content-Type

Messaggio ricevuto

```
Received: from hamburger.edu by sushi.ij 6 Oct 2003
Received: from crepes.fr by hamburger.edu; 6 Oct 2003
From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Picture of yummy crepe.
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Type: image/jpeg
```

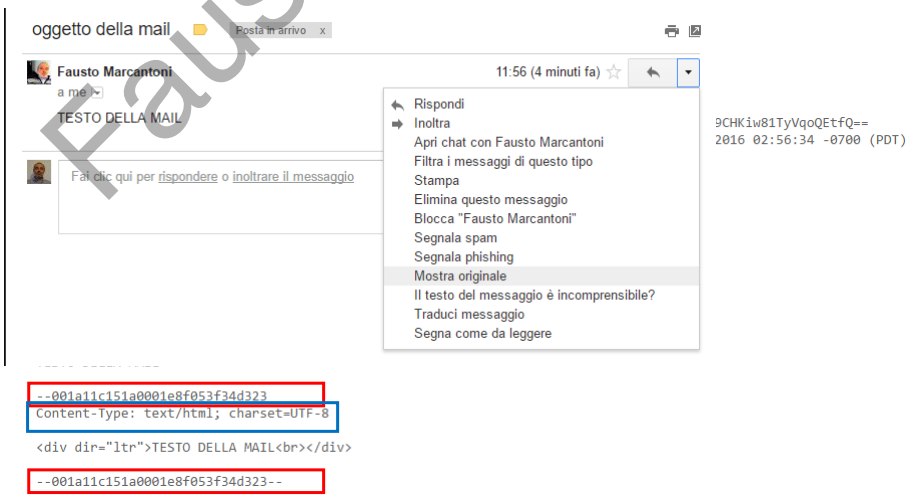
- **Received indica i Mail Server che hanno recapitato il messaggio**
- **Più linee "Received" se il messaggio è stato inoltrato da più server SMTP lungo il percorso da mittente a destinatario**

Esempio opzioni messaggio



157

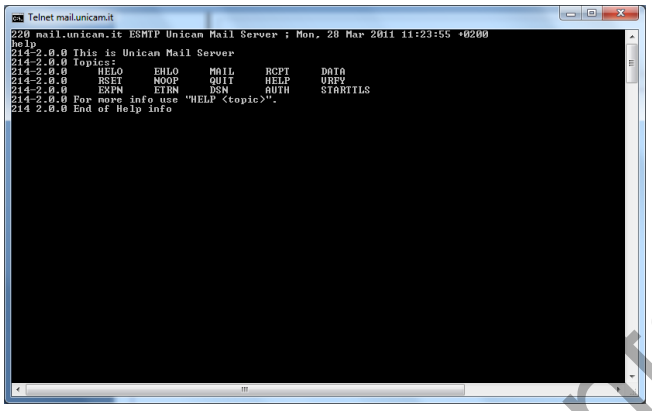
Gmail - Opzioni - Mostra originale



158

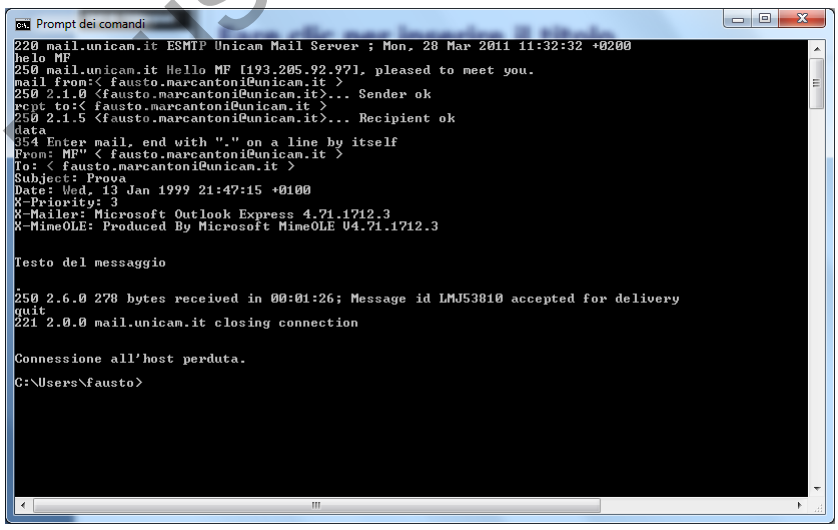
Esempio mail

telnet 90.147.42.137 25



159

Esempio mail



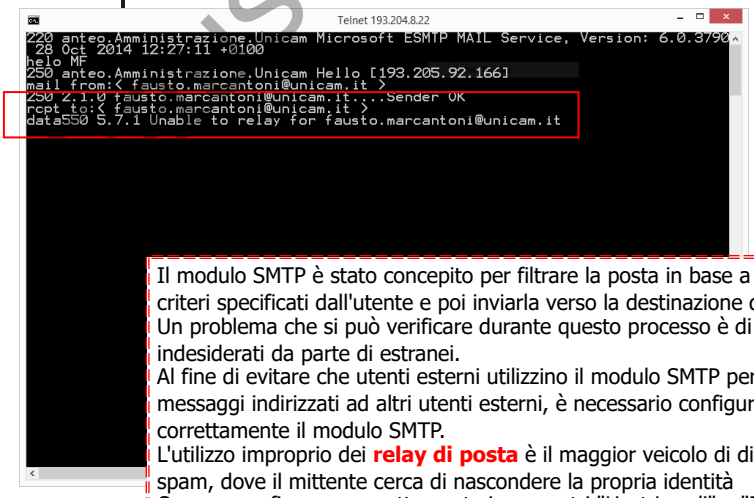
160

Fake mail

```
telnet mail.unicam.it 25

220 mail.unicam.it ESMTP Unicam Mail Server ; Mon, 28 Mar 2011 11:32:32 +0200
helo MF
250 mail.unicam.it Hello MF [193.205.92.97] pleased to meet you.
mail from:< fausto.marcantoni@unicam.it >
250 2.1.0 <fausto.marcantoni@unicam.it>... Sender ok
rcpt to:< fausto.marcantoni@unicam.it >
250 2.1.5 <fausto.marcantoni@unicam.it>... Recipient ok
data
354 Enter mail, end with "." on a line by itself
From: MF" < fausto.marcantoni@unicam.it >
To: < fausto.marcantoni@unicam.it >
Subject: Prova
Date: Mon, 28 Mar 2011 11:32:32 +0200
X-Priority: 3
X-Mailer: Microsoft Outlook Express 4.71.1712.3
X-MimeOLE: Produced By Microsoft MimeOLE V4.71.1712.3


Testo del messaggio

.
250 2.6.0 278 bytes received in 00:01:26; Message id LMJ53810 accepted for delivery
```

161

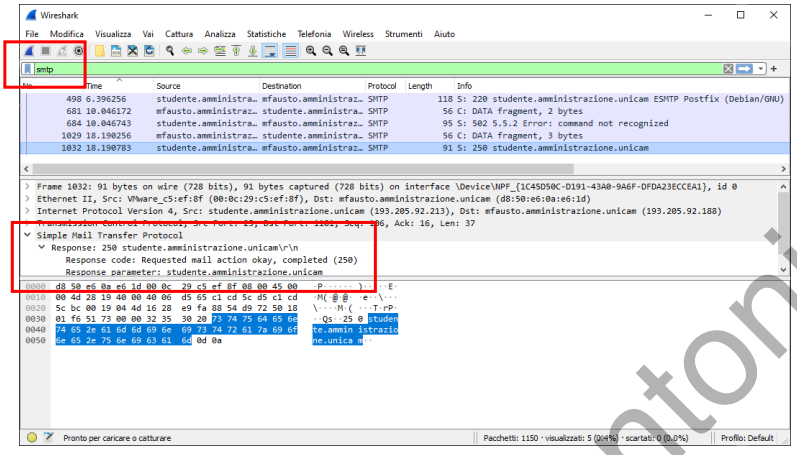
Mail relay



Il modulo SMTP è stato concepito per filtrare la posta in base a determinati criteri specificati dall'utente e poi inviarla verso la destinazione desiderata.
Un problema che si può verificare durante questo processo è di evitare utilizzi indesiderati da parte di estranei.
Al fine di evitare che utenti esterni utilizzino il modulo SMTP per spedire messaggi indirizzati ad altri utenti esterni, è necessario configurare correttamente il modulo SMTP.
L'utilizzo improprio dei **relay di posta** è il maggior veicolo di diffusione dello spam, dove il mittente cerca di nascondere la propria identità
Occorre configurare correttamente i parametri "Host Locali" e "Relay Domain".

162

smtp <-> wireshark



163

ESMTP command

```
faustomarcantoni — -zsh — 80×24
faustomarcantoni@MacBook-Pro-di-Fausto ~ % telnet smtp.libero.it 25
Trying 213.209.1.144...
Connected to smtp.libero.it.
Escape character is '^]'.
220 smtp-17.iol.local smtp-17.iol.local ESMTP server ready
HELP
214-This is smtp-17.iol.local ESMTP service help
214-To contact postmaster send email to postmaster@italiaonline.it.
214-For local information send email to Postmaster at your site.
214 end of HELP info
quit
221 smtp-17.iol.local smtp-17.iol.local closing connection
Connection closed by foreign host.
faustomarcantoni@MacBook-Pro-di-Fausto ~ %
```

Essential SMTP commands in the order they may be used

- HELO/EHLO
- MAIL FROM
- RCPT TO
- DATA
- NOOP
- HELP
- VRFY and EXPN
- RSET
- QUIT

Extended SMTP commands that some SMTP servers may support

- STARTTLS
- AUTH
- ATRN
- BDAT
- ETRN

https://mailtrap.io/blog/smtp-commands-and-responses/

https://www.samlogic.net/articles/smtp-commands-reference.htm

164

Fake Mailer

https://emkei.cz/

http://anonymailer.net/

mailsend
Send mail via SMTP protocol from command line

https://code.google.com/p/mailsend/
https://github.com/muquit/mailsend

Google fake mailer

165

Test http://anonymailer.net

test zorro Posta in arrivo x

zorro <zorro@yyyy.com>
a me

inglese › italiano Traduci messaggio

This email was sent via Anonymous email service for free.
YOU CAN REMOVE THIS TEXT MESSAGE BY BEING A PAID MEMBER FOR $19/year.
CLICK HERE =>
IP address of the sender:193.205.92.194 Message ID= 183555

test zorro

This email was sent via Anonymous email service for free.
YOU CAN REMOVE THIS TEXT MESSAGE BY BEING A PAID MEMBER FOR $19/year.
CLICK HERE =>
Message ID= 183555

Rispondi Inoltra

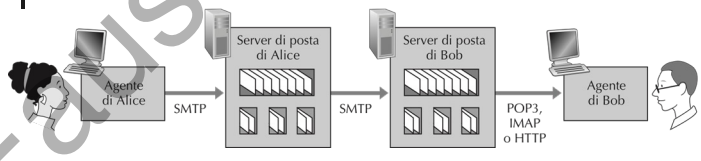
166

Protocolli di accesso alla posta

- Soluzione tradizionale: utente legge direttamente la posta sul Mail Server (shell, telnet/ssh+shell, …) archeologia
- L'host su cui è disposto il Mail Server deve essere sempre attivo
- Agenti di posta permettono di trasferire la posta dal Mail Server all'host locale del ricevente
- Possibilità visualizzare file multimediali e di specifiche applicazioni
- Occorre un protocollo "Push" per accedere alla Mailbox collocata sul Mail Server

167

Protocolli di accesso alla posta



SMTP: consegna al/memorizza nel server di posta del ricevente
Protocollo di accesso: recupero della posta dal server locale

- POP: Post Office Protocol [RFC 5034] https://tools.ietf.org/html/rfc5034
  - Autenticazione (agent <-->server) e scaricamento
- IMAP: Internet Mail Access Protocol [RFC 3501] https://tools.ietf.org/html/rfc3501
  - Più possibilità (più complesso)
  - Manipolazione dei messaggi memorizzati sul server
- HTTP: Gmail, Hotmail, Yahoo! Mail, Libero, ecc.

168

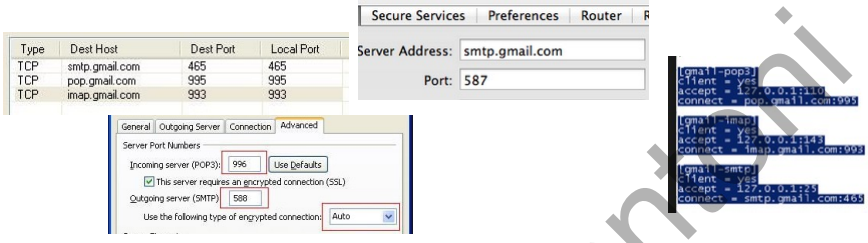
POP3 – IMAP : porte usate

By default, the ***POP3*** protocol works on two ports:

110 - this is the default POP3 non-encrypted port
995 - this is the port you need to use if you want to connect using POP3 securely

By default, the ***IMAP*** protocol works on two ports:

143 - this is the default IMAP non-encrypted port
993 - this is the port you need to use if you want to connect using IMAP securely



169

Protocollo POP3

Fase di autorizzazione
- Comandi del client:
  - `user`: nome utente
  - `pass`: password
- Risposte del server
  - `+OK`
  - `-ERR`

Fase di transazione, client:
- `list`: lista numeri e dim. msg
- `retr`: scarica messaggio in base al numero
- `dele`: cancella
- `quit`

```
S: +OK POP3 server ready
C: user alice
S: +OK
C: pass hungry
S: +OK user successfully logged on
```

```
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: <message 1 contents>
S: .
C: dele 1
C: retr 2
S: <message 1 contents>
S: .
C: dele 2
C: quit
S: +OK POP3 server signing off
```


170

Reti di Elaboratori

Il protocollo IMAP

IMAP
Internet Message Access Protocol
Interactive Mail Access Protocol
è un protocollo di comunicazione per la ricezione di e-mail.

Il significato "Interactive Mail Access Protocol" è stato valido fino alla versione 3, dalla quarta in poi è cambiato in "Internet Message Access Protocol"

Il protocollo è stato inventato da Mark Crispin nel 1986 come alternativa più moderna all'utilizzatissimo POP.

171

Reti di Elaboratori

Il protocollo IMAP

Accesso alla posta sia online che off-line
Con POP3 il client si connette, scarica le email sul computer, le cancella dal server e poi si disconnette. Con l'IMAP il client rimane connesso e sostanzialmente si limita a leggere il contenuto del server senza trasferire nulla; questo permette di risparmiare tempo se ci sono messaggi di grandi dimensioni.

Più utenti possono utilizzare la stessa casella di posta
Il protocollo POP assume che un solo client (utente) sia connesso ad una determinata mailbox (casella di posta), quella che gli è stata assegnata. Al contrario l'IMAP permette connessioni simultanee alla stessa mailbox, fornendo meccanismi per controllare i cambiamenti apportati da ogni utente.

Accesso a molteplici caselle di posta sul server
Alcuni utenti, con il protocollo IMAP, possono creare, modificare o cancellare mailbox (di solito associate a cartelle) sul server. Inoltre, questa gestione delle mailbox, permette di avere cartelle condivise tra utenti diversi.

Possibilità di fare ricerche sul server
L'IMAP permette al client di chiedere al server quali messaggi soddisfano un certo criterio, per fare, per esempio, delle ricerche sui messaggi senza doverli scaricare tutti.

Password criptate
Con il protocollo POP le password vengono solitamente inviate in testo, rendendo facile, con una intercettazione, l'individuazione della password. Con l'IMAP è possibile criptare la password.

172

Protocollo POP3 e IMAP

2 modalità di utilizzo

- Scarica ed elimina:
  1. User Agent elimina la posta dalla Mailbox dopo averla scaricata
  2. Un utente disperde la posta sui diversi host da cui accede la Mailbox
  3. User Agent permette di creare cartelle, spostare messaggi, effettuare ricerce nei messaggi
- Scarica e conserva
  1. User Agent conserva la posta sulla Mailbox
  2. Utente può leggere i messaggi da macchine diverse
- Permette di gestire cartelle di posta remote come se fossero locali
- IMAP deve mantenere una gerarchia di cartelle per ogni utente

Vantaggi di IMAP

Vantaggi di IMAP

- ***Sicurezza.*** I messaggi stanno sul server per cui non rischiano di essere persi se il computer si dovesse rompere.
- ***Semplicità.*** Se l'utente cambia il computer o cambia client di posta elettronica, non è necessario trasferire tutte le email sul nuovo computer o sul nuovo programma.
- ***Velocità.*** Dato che le email non devono essere trasferite dal server al singolo computer, la velocità è massima anche in presenza di messaggi con allegati di grosse dimensioni. Il server IMAP inoltre scarica in continuo le email su di lui e tutti i client accedono alle email in rete locale.
- ***Accessibilità.*** Se di solito scaricate le email sul vostro computer in ufficio e, da casa o da fuori ufficio, dovete controllare un'email ricevuta, dovete per forza recarvi in ufficio. Con un server IMAP invece potete accedere alla vostra casella di posta da qualsiasi computer connesso a internet senza nessun problema e con la massima velocità.

Esercitazione

internet reti sicurezza

Esercitazioni

175

Protocolli di livello applicativo

- ✓ **FTP: file transfer protocol**
- ✓ **TFTP: Trivial File Transfer Protocol**
- ✓ **Posta elettronica**
  - ➢ **smtp**
  - ➢ **pop3 - imap**
- ✓ **DNS: Domain Name System**
- ✓ **SNMP: Simple Network Management Protocol**
- ✓ **Remote login: telnet - ssh**

176

DNS: Domain Name System

Come identifico le persone:
- molte mezzi di identificazione: CF, nome, # Passaporto

Come identifico computer, host, router, … :
- Indirizzi IP (32 bit) – usati per indirizzare i datagrammi IP
- "Nome", es., dida.cs.unicam.it – usati dagli utenti

Risposta: corrispondenza tra indirizzo IP e nome?

Domain Name System
- ***Database distribuito*** implementato come una gerarchia di molti *name server*
- ***Protocollo applicativo*** usato da host, router, name server per comunicare allo scopo di ***risolvere*** (tradurre) i nomi in indirizzi IP
  - Funzione di base di Internet implementata come protocollo applicativo
  - La complessità trasferita al "bordo" della rete

Name server DNS

Perché non un server DNS centralizzato?
- ★ Minor tolleranza ai guasti
- ★ Traffico eccessivo
- ★ Database centrale troppo distante in molti casi
- ★ Scarsa scalabilità!

- Nessun **name server** contiene tutte le associazioni nome simbolico/indirizzo IP

Name server locali :
- Ogni ISP o compagnia ha un ***name server locale (default)***
- La richiesta di traduzione (mapping) di un host è prima rivolta al name server locale

Name server di riferimento (assoluto **authoritative**):
- Per un host: per definizione è quello che è sempre in grado di eseguire la traduzione (mapping) nome simbolico/indirizzo IP dell' host

Reti di Elaboratori

DNS Primario - Secondario

Ogni dominio deve avere almeno due DNS Server per funzionare:
Primario (master)
Secondario (slave) e se fosse possibile un altro Secondario esterno alla rete del dominio che gestisce.

Il Primario ha i dati effettivi, mentre i **Secondari ne ha una copia** (Transfer Zone) ricevuta dal Primario.

Gli Amministratori di Rete, cambiano/aggiornano/modificano i dati solo nel DNS Primario, i Secondari verranno aggiornati automaticamente.

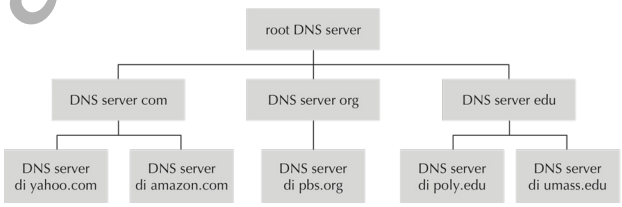
Ogni DNS contiene le informazioni relative al proprio dominio (zona) e sa come ottenere le informazioni di quelli sottostanti (in quanto ne ha delegato la gestione ad altri DNS, oppure li gestisce direttamente).

Quindi il DNS del dominio ***.it*** sa chi detiene le informazioni del dominio ***libero.it*** e chi detiene le informazioni del dominio ***unicam.it*** ma non sa chi detiene le informazioni per il dominio ***mat.unicam.it***

179

Reti di Elaboratori

Gerarchia di server DNS



180

Reti di Elaboratori

Root name server

I **root "name" servers** sono dei server DNS che hanno tutte le informazioni relative ai domini di 1 livello (top-level) ed indirizzano i vari DNS nelle traduzioni da nomi di domini in indirizzi IP e viceversa.

I DNS server di un dominio (zona) sono detti **autoritativi** per quel dominio, mentre risulteranno **non autoritativi** per tutti gli altri domini che non gestiscono direttamente.

I DNS del dominio ***.it*** non sono autoritativi per il dominio ***libero.it***.
Ogni client di una rete ha installato automaticamente nel sistema operativo un **DNS client (resolver)** che manda le richieste ai DNS server configurati nelle proprietà di rete.

E' **FQDN (Full Qualified Domain Name)** un nome (di un host) comprensivo del dominio di appartenenza: ***www.mat.unicam.it*** è un FQDN, ***www.libero.it*** è un FQDN, ***www*** non è un FQDN.

Il DNS server usa il protocollo **UDP ed ascolta sulla porta 53** per le richieste da parte dei client (che usano invece una porta alta, sulla quale risponderà il DNS), usa invece **protocollo TCP e porta 53** per i trasferimenti di zona con i DNS Secondari.

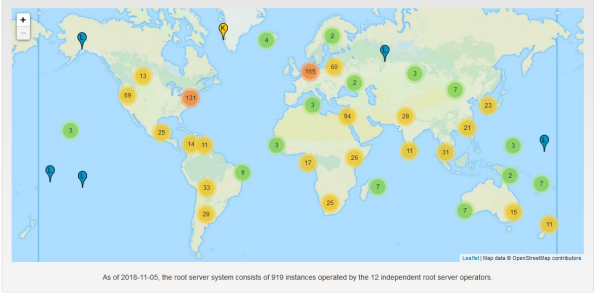
181

Reti di Elaboratori

DNS: Root name servers

- Contattato dal name server locale che non riesce a risolvere un nome
- root name server:
  - Contatta il name server di riferimento (authoritative) se la traduzione non è nota
  - Ottiene la traduzione
  - Restituisce la traduzione al name server locale
- ~ una dozzina di root name server nel mondo

Verisign USC-ISI Cogent UMD NASA Ames ISC DISA DoD NIC ARL Netnod RIPE NCC ICANN WIDE



http://www.root-servers.org/

182

Reti di Elaboratori

Root Zone Database

iana
Internet Assigned Numbers Authority

DOMAIN	TYPE	TLD MANAGER
.aaa	generic	American Automobile Association, Inc.
.aarp	generic	AARP
.abarth	generic	Fiat Chrysler Automobiles N.V.
.abb	generic	ABB Ltd
.abbott	generic	Abbott Laboratories, Inc.
.abbvie	generic	AbbVie Inc.
.abc	generic	Disney Enterprises, Inc.
.able	generic	Able Inc.
.abogado	generic	Minds + Machines Group Limited
.abudhabi	generic	Abu Dhabi Systems and Information Centre
.ac	country-code	Network Information Center (AC Domain Registry) c/o Cable and Wireless (Ascension Island)

https://www.iana.org/domains/root/db

183

Reti di Elaboratori

Root Zone Database - whois

iana
Internet Assigned Numbers Authority

IANA WHOIS Service

The IANA WHOIS Service is provided using the WHOIS protocol on port 43. This web gateway will query thi
query arguments are domain names, IP addresses and AS numbers.

unicam.it Submit

https://www.iana.org/whois

184

Whois - italia

https://www.nic.it/it

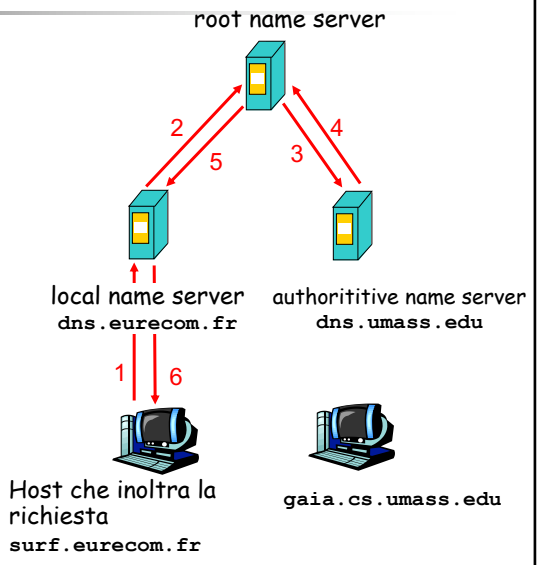
https://web-whois.nic.it/

185

Esempio (1)

L' host `surf.eurecom.fr` vuole l'indirizzo IP di `gaia.cs.umass.edu`

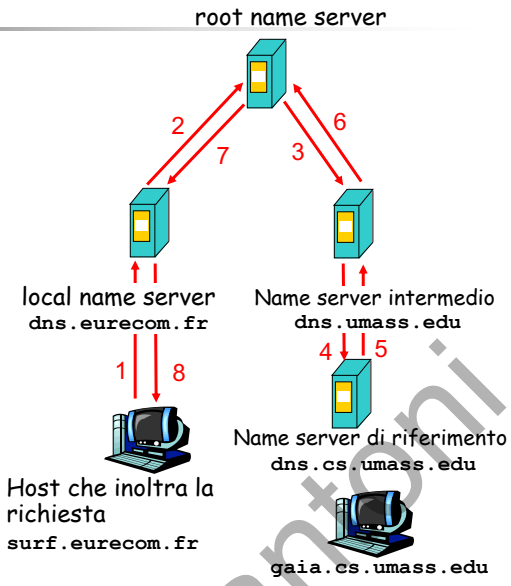
1. Contatta il server DNS locale, `dns.eurecom.fr`
2. `dns.eurecom.fr` contatta il root name server, se necessario
3. Il root name server contatta il name server di riferimento, `dns.umass.edu`, se necessario



186

Esempio (2)

Root name server:
- Può non essere a conoscenza di un name server di riferimento
- Può tuttavia conoscere un *name server intermedio* che contatta per raggiungere quello di riferimento

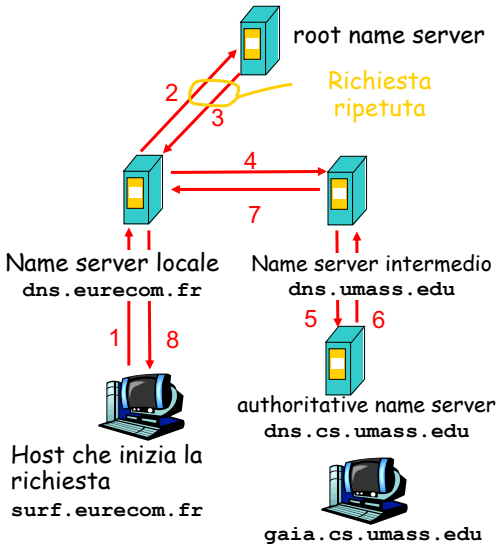


187

DNS: richieste ripetute (iterated queries)

Richieste ricorsive (recursive query):
- Trasferisce il carico della traduzione al name server contattato
- Carico eccessivo?

Richieste ripetute (iterated query):
- Il name server contattato risponde con l'indirizzo del prossimo name server da contattare
- "Non conosco questo nome, ma prova a rivolgerti a quest'altro server"

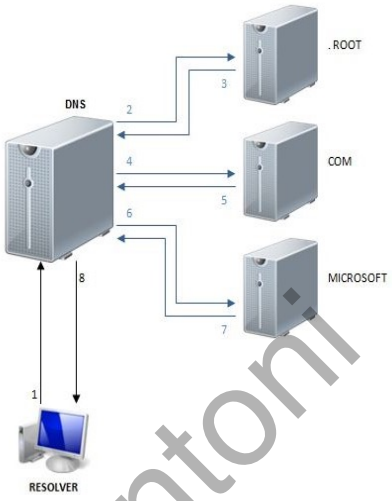


188

Esempio: query dns

Il resolver ha bisogno dell'indirizzo IP di microsoft.com

1. Il client contatta il DNS con una RECURSIVE QUERY : il DNS dovrà rispondere o con l'IP o con il messaggio di errore
2. Il DNS non trova il nome ne' nella propria cache ne' nelle proprie zone, quindi contatta il DNS autoritativo ROOT con una ITERATIVE QUERY
3. Il DNS non conosce microsoft.com quindi restituisce l'indirizzo del referente per il dominio COM
4. Il DNS contatta il server autoritativo per il dominio COM con una ITERATIVE QUERY
5. Il DNS autoritativo per COM non conosce microsoft.com quindi restituisce l'indirizzo del referente per il dominio Microsoft.com
6. Il DNS contatta il server autoritativo per il dominio MICROSOFT.COM con una ITERATIVE QUERY
7. Il DNS autoritativo per MICROSOFT.COM restituisce l'IP address richiesto.
8. Il DNS contattato dal resolver gli restituisce l'IP address richiesto.



189

MODALITA' ITERATIVA - RICORSIVA

MODALITA' ITERATIVA

Il server DNS invia al client DNS (Resolver) la risposta alla richiesta di risoluzione o l'indirizzo di un server che secondo lui è in grado di risolvere il nome

MODALITA' RICORSIVA

Il client si aspetta dal server DNS la risposta alla sua richiesta. Il server DNS, se è responsabile del dominio, risolve l'indirizzo altrimenti trasmette la richiesta ad un server DNS di livello superiore e aspetta la risposta per il client

In ogni caso, il resolver genera un'interrogazione contenente:
- nome da risolvere
- dichiarazione della classe del nome
- tipo di risposta desiderata
- modalità

190

Reti di Elaboratori

DNS riepilogo

Quando un server DNS riceve un'interrogazione, **verifica se il nome è relativo alla propria zona di autorità**:

1. se sì, **converte il nome in un indirizzo** in base alle informazioni del proprio database, inserisce l'indirizzo nella risposta e la spedisce al client

2. se non può risolvere completamente il nome, **esamina la modalità specificata nell'interrogazione**:
   A. Se è **ricorsiva**, contatta un server di livello superiore che possa risolvere il nome, e riporta al client la risposta ottenuta
   B. Se è **iterativa**, si limita a comunicare al client il nome del server a cui rivolgersi

191

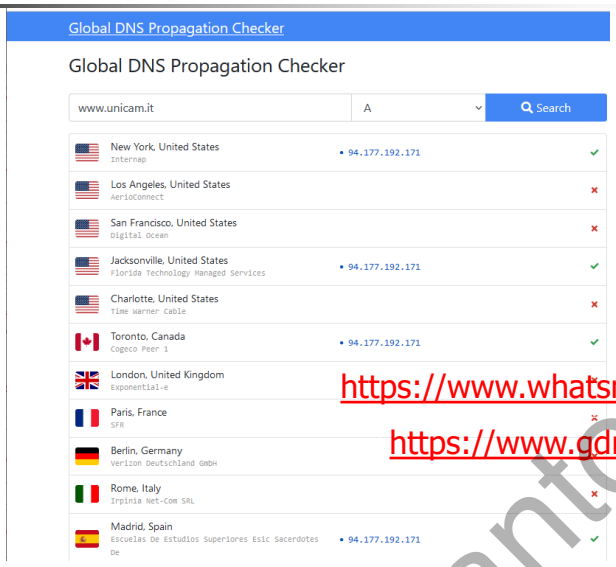
Reti di Elaboratori

DNS: caching e aggiornamento

- Quando un qualsiasi name server apprende una traduzione la memorizza localmente **(caching)**
  - Le traduzioni memorizzate nella cache **(cache entries)** scadono **(timeout)** dopo un certo tempo **(di solito un paio di giorni)**
- Se possibile, richieste successive vengono servite usando la traduzione presente in cache
- I meccanismi di aggiornamento/modifica in studio da parte dell' IETF
  - RFC 2136
  - http://www.ietf.org/html.charters/dnsind-charter.html

192

DNS Propagation Checker



https://www.whatsmydns.net

https://www.gdnspc.com/

193

cache DNS

Come è possibile visualizzare il contenuto della cache DNS?

ipconfig /displaydns

Come è possibile cancellare il contenuto della cache DNS?

ipconfig /flushdns

rndc
resolvectl status

sudo dscacheutil -flushcache;sudo killall -HUP mDNSResponder

La cache DNS è mantenuta dal servizio **mDNSResponder**.
Se il tuo macOS è una versione molto recente, potrebbe non mostrare tutte le voci DNS in modo dettagliato, poiché alcune informazioni sono gestite internamente dal sistema.

194

Recursive/Iterative Queries in DNS

Recursive/Iterative Queries in DNS

https://media.pearsoncmg.com/curriculum/intl/it/lab/9788891902559_KUROSE_ROSS/studente/interactive%20animations/recursive-iterative-queries-in-dns/index.html

195

Record DNS

DNS: database distribuito che memorizza Resource Record (RR)

Formato RR: `(nome, valore, tipo, ttl)`

- Tipo=A
  - **`nome`** è il nome dell' host
  - **`valore`** è l' indirizzo IP
- Tipo=CNAME
  - **`nome`** è un alias di qualche nome reale ("canonico")
  - **`valore`** è il nome canonico
- Tipo=NS
  - **`nome`** è il dominio (es. foo.com)
  - **`valore`** è l'indirizzo IP del name server di riferimento per questo dominio
- Tipo=MX
  - **`valore`** è il nome di un mailserver associato a **`nome`**

196

Reti di Elaboratori

Record DNS – Esempio Master Zone

Edit Master Zone

193.205.92

Reverse Address (18) | Name Server (1) | Name Alias (0) | All Record Types (19)

Edit Records File | Edit Zone Parameters | Edit Zone Options | Record Generators

webmin

http://www.webmin.com/

197

Reti di Elaboratori

Record DNS – Esempio - All Record

All Records

In 193.205.92

Name	Type	TTL	Values
92.205.193.in-addr.arpa.	NS	Default	nameserver.cs.unicam.it.
1.92.205.193.in-addr.arpa.	PTR	Default	nameserver.cs.unicam.it.
28.92.205.193.in-addr.arpa.	PTR	Default	giasone.cs.unicam.it.
33.92.205.193.in-addr.arpa.	PTR	Default	geronaccess.cs.unicam.it.
3.92.205.193.in-addr.arpa.	PTR	Default	mail.cs.unicam.it.
28.92.205.193.in-addr.arpa.	PTR	Default	cvs.cs.unicam.it.
5.92.205.193.in-addr.arpa.	PTR	Default	www.cs.unicam.it.
32.92.205.193.in-addr.arpa.	PTR	Default	ibfit.cs.unicam.it.
5.92.205.193.in-addr.arpa.	PTR	Default	intranet.cs.unicam.it.
46.92.205.193.in-addr.arpa.	PTR	Default	sibilla.cs.unicam.it.

Name	Type	TTL	Values
48.92.205.193.in-addr.arpa.	PTR	Default	carteraffaello.cs.unicam.it.
54.92.205.193.in-addr.arpa.	PTR	Default	cosy.cs.unicam.it.
53.92.205.193.in-addr.arpa.	PTR	Default	geronacces.cs.unicam.it.
56.92.205.193.in-addr.arpa.	PTR	Default	dida.cs.unicam.it.
55.92.205.193.in-addr.arpa.	PTR	Default	sicetsinpliciter.cs.unicam.it.
59.92.205.193.in-addr.arpa.	PTR	Default	shib.cs.unicam.it.
60.92.205.193.in-addr.arpa.	PTR	Default	guernica.cs.unicam.it.
42.92.205.193.in-addr.arpa.	PTR	Default	usr.cs.unicam.it.
22.92.205.193.in-addr.arpa.	PTR	Default	iceproject.cs.unicam.it.

198

Record DNS – Esempio – Reverse Address Record

Reverse Address Records

In 193.205.92

Add Reverse Address Record

Address 193.205.92. Time-To-Live Default seconds

Hostname

Update forward? Yes No Create

Address	TTL	Hostname	Address	TTL	Hostname
193.205.92.1	Default	nameserver.cs.unicam.it.	193.205.92.48	Default	carteraffaello.cs.unicam.it.
193.205.92.28	Default	giasone.cs.unicam.it.	193.205.92.54	Default	cosy.cs.unicam.it.
193.205.92.33	Default	geronaccess.cs.unicam.it.	193.205.92.53	Default	geronacces.cs.unicam.it.
193.205.92.3	Default	mail.cs.unicam.it.	193.205.92.56	Default	dida.cs.unicam.it.
193.205.92.28	Default	cvs.cs.unicam.it.	193.205.92.55	Default	sicetsinpliciter.cs.unicam.it.
193.205.92.5	Default	www.cs.unicam.it.	193.205.92.59	Default	shib.cs.unicam.it.
193.205.92.32	Default	ibfit.cs.unicam.it.	193.205.92.60	Default	guernica.cs.unicam.it.
193.205.92.5	Default	intranet.cs.unicam.it.	193.205.92.42	Default	usr.cs.unicam.it.
193.205.92.46	Default	sibilla.cs.unicam.it.	193.205.92.22	Default	iceproject.cs.unicam.it.

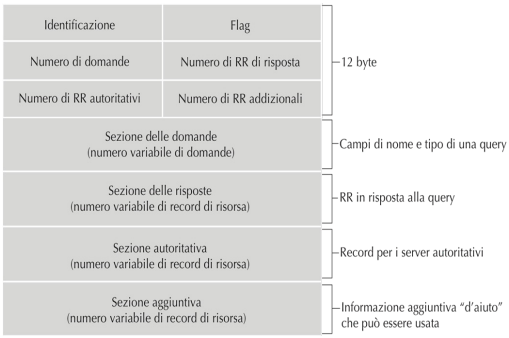
Protocollo DNS, messaggi (1/2)

Protocollo DNS : messaggi di *richiesta (query)* e *risposta (reply)*, *client-server*

Header di messaggio

- identification: numero a 16 bit per la richiesta, la risposta usa lo stesso numero
- flags:
  - Richiesta o risposta
  - Bit di competenza
  - Richiesta di ricorsione (Q)
  - Ricorsione disponibile (R)
  - Il server che risponde è di riferimento per la richiesta (R)

Nota: richiesta e risposta hanno lo stesso formato



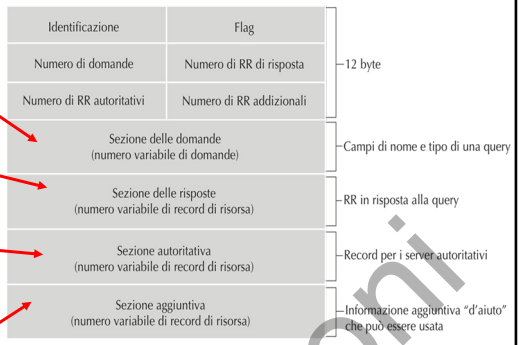
Protocollo DNS, messaggi (2/2)

Nome, campi tipo per una richiesta

RR in risposta a una richiesta

Record per server di riferimento

Informazioni addizionali
Es.: RR di tipo A contenente indirizzo IP di un mail server il cui nome canonico è contenuto nella answer section



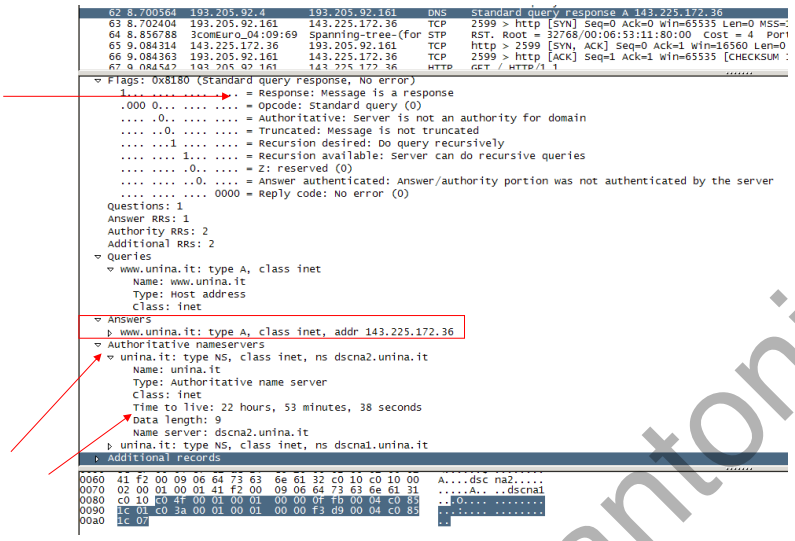
201

DNS – Esempio query

```
    61 8.697688  193.205.92.161     193.205.92.4       DNS    Standard query A www.unina.it
    62 8.700564  193.205.92.4       193.205.92.161     DNS    Standard query response A 143.225.172.36
    63 8.702404  193.205.92.161     143.225.172.36     TCP    2599 > http [SYN] Seq=0 Ack=0 Win=65535 Le
    64 8.856788  3ComEuro_04:09:69  Spanning-tree-(for STP    RST. Root = 32768/00:06:53:11:80:00  Cost
    65 9.084314  143.225.172.36     193.205.92.161     TCP    http > 2599 [SYN, ACK] Seq=0 Ack=1 Win=165
    66 9.084363  193.205.92.161     143.225.172.36     TCP    2599 > http [ACK] Seq=1 Ack=1 Win=65535 [C
    67 9.084542  193.205.92.161     143.225.172.36     HTTP   GET / HTTP/1.1
▷ Frame 61 (72 bytes on wire, 72 bytes captured)
▷ Ethernet II, Src: 00:0f:b0:44:69:10, Dst: 00:0a:5e:02:83:f2
▷ Internet Protocol, Src Addr: 193.205.92.161 (193.205.92.161), Dst Addr: 193.205.92.4 (193.205.92.4)
▷ User Datagram Protocol, Src Port: 1988 (1988), Dst Port: domain (53)
▽ Domain Name System (query)
    Transaction ID: 0xce9b
  ▽ Flags: 0x0100 (Standard query)
      0... .... .... .... = Response: Message is a query
      .000 0... .... .... = Opcode: Standard query (0)
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
      .... .... .0.. .... = Z: reserved (0)
      .... .... ...0 .... = Non-authenticated data OK: Non-authenticated data is unacceptable
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▽ Queries
    ▽ www.unina.it: type A, class inet
        Name: www.unina.it
        Type: Host address
        Class: inet

0000  00 0a 5e 02 83 f2 00 0f  b0 44 69 10 08 00 45 00   ..^..... .Di...E.
0010  00 3a 42 6f 00 00 80 11  bc 03 c1 cd 5c a1 c1 cd   .:Bo.... ....\...
0020  5c 04 07 c4 00 35 00 26  18 32 ce 9b 01 00 00 01   \....5.& .2......
0030  00 00 00 00 00 00 03 77  77 77 05 75 6e 69 6e 61   .......w ww.unina
0040  02 69 74 00 00 01 00 01                           .it.....
```


202

DNS – Esempio response



nslookup

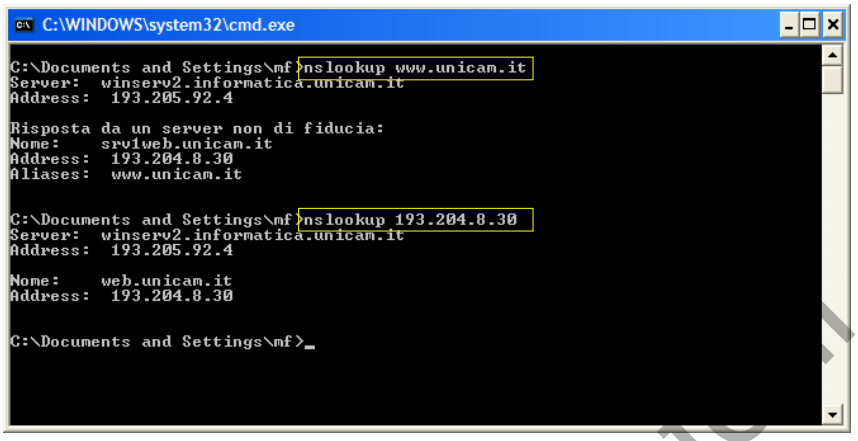
- NSLookup (Name Server Lookup) è un programma che effettua ricerche all'interno dei database del DNS
- l'utente comune non ha alcun bisogno di effettuare questo tipo di ricerche, se non per curiosità, o per qualche caccia al tesoro informatica;

nslookup

nslookup

dig

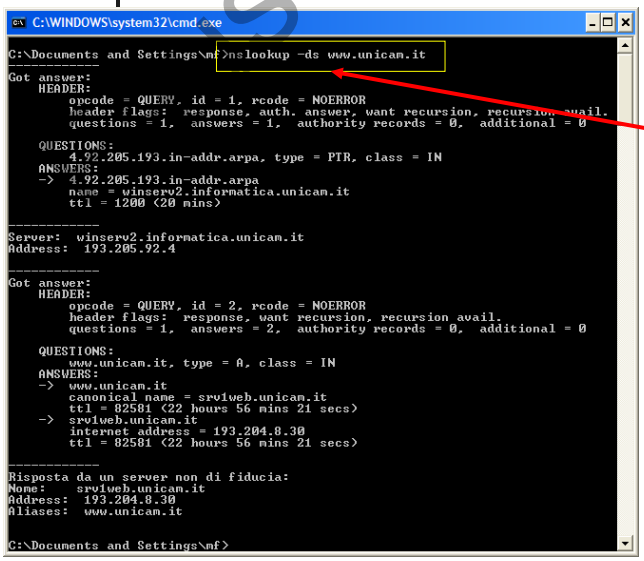
nslookup - esempio



https://technet.microsoft.com/it-it/library/cc645505.aspx

205

nslookup - esempio



206

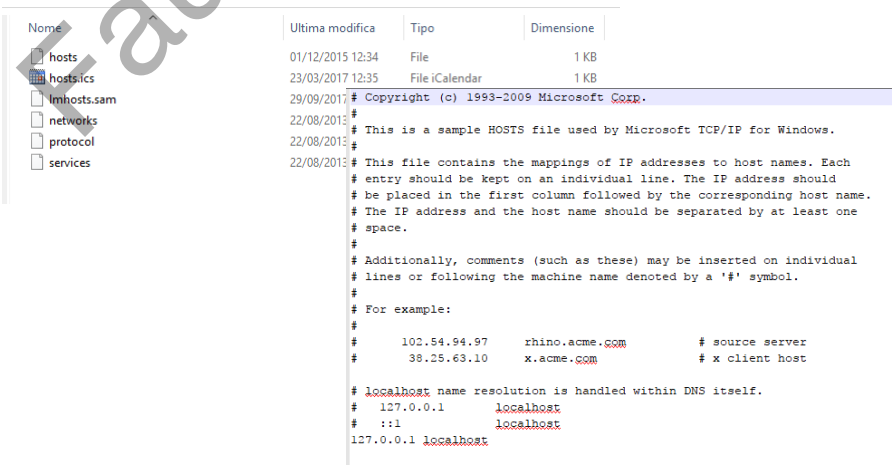
dig - esempio

```
studente@server-IRS: ~
;; MSG SIZE  rcvd: 299

studente@server-IRS:~$ dig

; <<>> DiG 9.18.1-1ubuntu1.2-Ubuntu <<>>
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 63029
;; flags: qr rd ra; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 4

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;.                              IN      NS

;; ANSWER SECTION:
.                       3146    IN      NS      d.root-servers.net.
.                       3146    IN      NS      b.root-servers.net.
.                       3146    IN      NS      a.root-servers.net.
.                       3146    IN      NS      l.root-servers.net.
.                       3146    IN      NS      k.root-servers.net.
.                       3146    IN      NS      i.root-servers.net.
.                       3146    IN      NS      f.root-servers.net.
.                       3146    IN      NS      e.root-servers.net.
.                       3146    IN      NS      h.root-servers.net.
.                       3146    IN      NS      j.root-servers.net.
.                       3146    IN      NS      m.root-servers.net.
.                       3146    IN      NS      g.root-servers.net.
.                       3146    IN      NS      c.root-servers.net.

;; ADDITIONAL SECTION:
a.root-servers.net.     3146    IN      AAAA    2001:503:ba3e::2:30
a.root-servers.net.     3146    IN      A       198.41.0.4
m.root-servers.net.     3146    IN      A       202.12.27.33

;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Tue Oct 18 11:05:37 CEST 2022
;; MSG SIZE  rcvd: 299

studente@server-IRS:~$
```

man dig

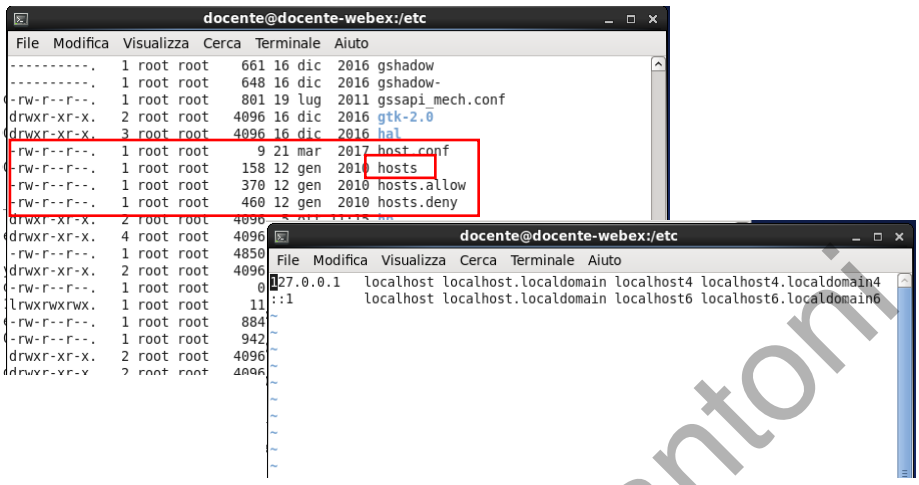
Il file hosts (windows)

C:\Windows\System32\drivers\etc\hosts



Il file hosts (linux)

/etc/hosts

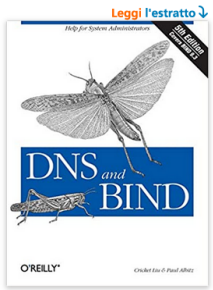


Il file hosts

quando vogliamo aprire una determinata pagina web, prima viene controllato questo file, per vedere se l'host dell'URL è contenuto al suo interno e:

- se il file hosts contiene la corrispondenza hostname -> IP, allora viene usato questo indirizzo IP per aprire la pagina desiderata.
- se il file hosts non contiene l'hostname, allora viene contattato il server DNS a cui viene chiesto di risolvereil nome dell'host dell'URL in indirizzo IP.

approfondimenti



DNS and BIND Copertina flessibile – 20 giugno 2006

Edizione Inglese | di Cricket Liu (Autore), Paul Albitz (Autore)

95 voti

Visualizza tutti i formati ed edizioni

Formato Kindle 26,61 €

Copertina flessibile 41,60 €

Leggilo con la nostra App gratuita

1 Usato da 19,03 €
6 Nuovo da 41,60 €

Spedizione GRATUITA con consegna presso **punti di ritiro (se disponibile per il tuo ordine**

Questo articolo è acquistabile con il Bonus Cultura e con il Bonus Carta del Docente e spedito direttamente da Amazon. Sono esclusi prodotti di Venditori terzi sul Marke
Verifica i termini e condizioni dell' iniziativa Bonus Cultura 18app e di Carta del Doce

DNS and BIND tells you everything you need to work with one of the Internet's fundamental buildi

https://www.mrwebmaster.it/hosting/dns-come-funziona_11718.html

https://www.extraordy.com/dns-e-bind-concetti-e-configurazione-caching-dns/

https://www.apogeonline.com/contrib/uploads/linux-server-amministratore-di-rete-capitolo4.pdf

Enumerate DNS

https://dnsdumpster.com/

https://github.com/fwaeytens/dnsenum

sudo apt install dnsenum
dnsenum --enum <*domain*>

https://github.com/darkoperator/dnsrecon

sudo apt install dnsrecon
dnsrecon -d <*domain*>
dnsrecon -d <*domain*> -t axfr
dnsrecon –r <*ip_range*>

Laboratorio

internet reti sicurezza

Esercitazioni

213

- ✓ **FTP: file transfer protocol**
- ✓ **TFTP: Trivial File Transfer Protocol**
- ✓ **Posta elettronica**
  - ➢ **smtp**
  - ➢ **pop3 - imap**
- ✓ **DNS: Domain Name System**
- ✓ **SNMP: Simple Network Management Protocol**
- ✓ **Remote login: telnet - ssh**

214

Reti di Elaboratori

Simple Network Management Proocol

Gestione delle reti: SNMP

RFC 1157
https://datatracker.ietf.org/doc/html/rfc1157

- ✓ Simple Network Management Proocol
- ✓ disponibile in praticamente tutti i dispositivi di rete maneggiabili, comprese le stampanti, i printer server, i fax server e, spesso in modo nascosto, anche i sistemi operativi.
- ✓ Semantica del protocollo SNMP:
  - ➢ GET, GETNEXT ("WALK"), GETBULK, SET, TRAP

SNMP Agent

SNMP Manager

Reti di Elaboratori

SNMP

SNMP è usato diffusamente per monitorare e configurare dispositivi di rete e anche computer o software

- ❑ SNMP version 1, 2c e 3
- ❑ Uso di ACL
- ❑ Specificare SNMP per porta
- ❑ Blocco di SNMP al firewall

UDP port 161

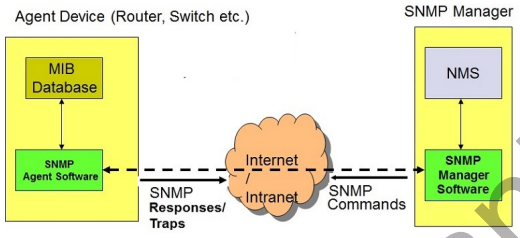
Ogni stazione di gestione o agente in una rete gestita da SNMP mantiene un database locale di informazioni rilevanti per la gestione della rete, noto come **Management Information Base** (MIB)

SNMP Agent - Station

Modello client-server

- SNMP Station: modulo che emette una richiesta
  - **Get-Request** per richiedere informazioni all'Agent
  - **Get-Next-Request** per richiedere righe successive di tabelle dell'Agent (ritorna errore a fine tabella)
  - **Set-Request** per configuarare parametri dell'Agent
- SNMP Agent: modulo che formula la risposta
  - **Get-Response** per fornire informazioni alla Station
  - **Trap** per riportare un evento a una Station predefinita

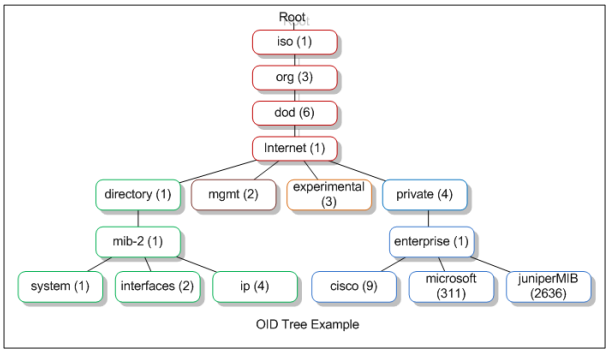
SNMP Architecture



217

Albero OID

- Ogni oggetto nel MIB ha un identificatore di oggetto (OID), che la stazione di gestione usa per richiedere il valore dell'oggetto dall'agente.
- Un OID è una sequenza di numeri interi che identifica univocamente un oggetto gestito definendo un percorso verso quell'oggetto attraverso una struttura ad albero chiamata albero OID o albero di registrazione.
- Quando un agente SNMP ha bisogno di accedere a uno specifico oggetto gestito, attraversa l'albero OID per trovare l'oggetto.



218

identificatore di oggetto (OID)

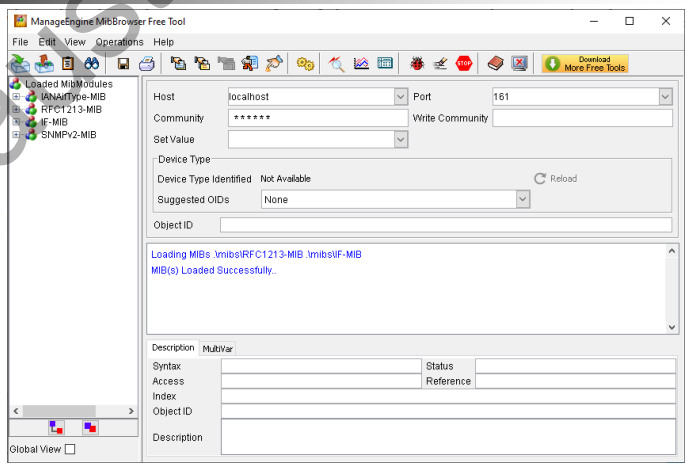
Per ottenere il tempo di accensione del sistema di un dispositivo gestito, è possibile eseguire il polling di questo

```
OID -1.3.6.1.6.3.10.2.1.3
```

e l'apparato restituirà il numero di secondi dall'ultima volta che il motore SNMP è stato attivato.

```
sysDescr (1.3.6.1.2.1.1.1)
sysObjectID (1.3.6.1.2.1.1.2)
sysUpTime (1.3.6.1.2.1.1.3)
sysContact (1.3.6.1.2.1.1.4)
sysName (1.3.6.1.2.1.1.5)
sysLocation (1.3.6.1.2.1.1.6)
sysServices (1.3.6.1.2.1.1.7)
```


Tools SNMP

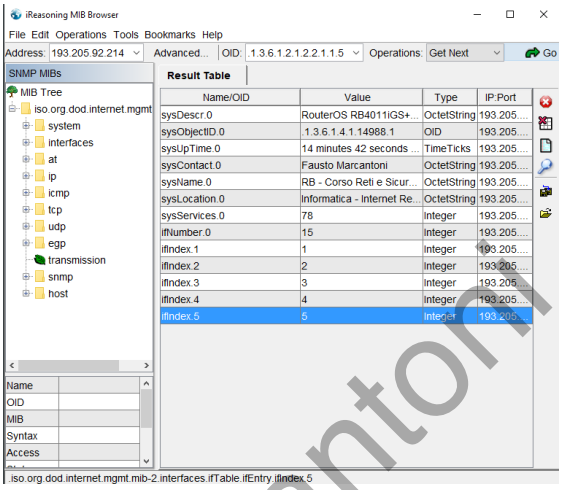


ManageEngine

https://www.manageengine.com/products/mibbrowser-free-tool/download-confirm.html

Tools SNMP

http://www.ireasoning.com/



221

Come installare e configurare SNMP su Windows 10

Come abilitare SNMP utilizzando PowerShell

Per abilitare SNMP tramite PowerShell, assicurati innanzitutto che il tuo computer abbia accesso a Internet. In tal caso, esegui un PowerShell con privilegi elevati premendo Win + X e selezionando Windows PowerShell (Admin) . Eseguire il seguente comando per installare i server SNMP dai server Microsoft:

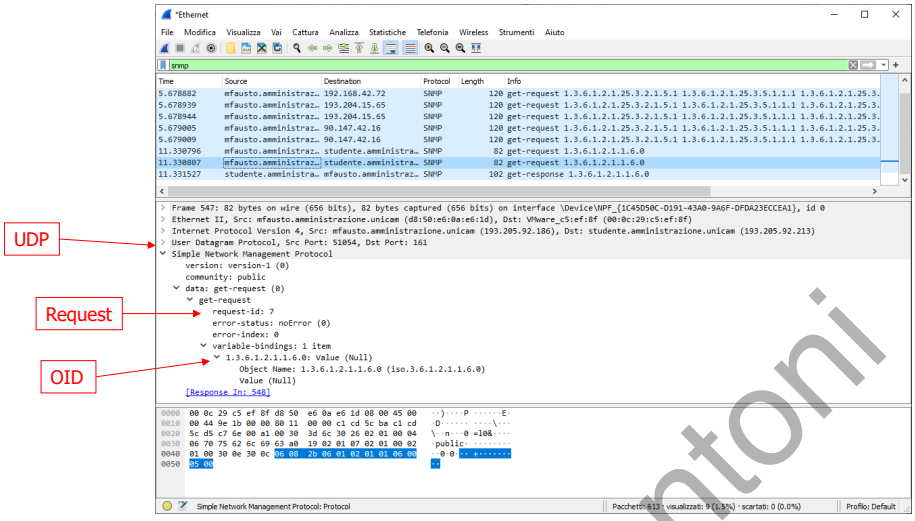
```
Add-WindowsCapability -Online -Name "SNMP.Client----0.0.1.0"
```

https://www.tecnobabele.com/come-installare-e-configurare-snmp-su-windows-10/2021-08-27/

222

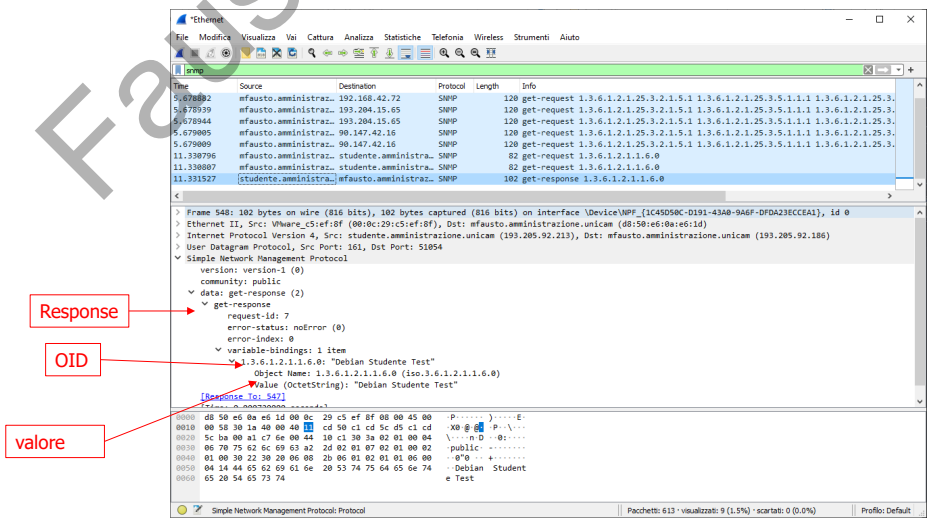
Reti di Elaboratori

snmp <-> wireshark



Reti di Elaboratori

snmp <-> wireshark



- ✓ **FTP: file transfer protocol**
- ✓ **TFTP: Trivial File Transfer Protocol**
- ✓ **Posta elettronica**
  - ➢ **smtp**
  - ➢ **pop3 - imap**
- ✓ **DNS: Domain Name System**
- ✓ **SNMP: Simple Network Management Protocol**
- ✓ **Remote login: telnet - ssh**

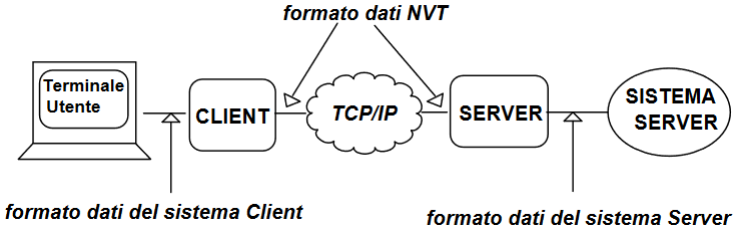
225

Remote Login - telnet

Il protocollo **telnet** permette di accedere a un servizio interattivo con un server remoto in modalità di console a caratteri.

Ciò significa che l'host client diventa un terminale remoto del server.

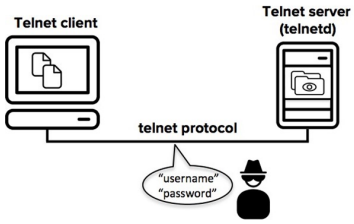
Sulla console del client si apre una shell di comandi per il server.
Più precisamente, ad entrambi i capi della comunicazione, il protocollo telnet prevede istanze paritetiche di un NVT (Network Virtual Terminal).



226

Remote Login - telnet

telnetd
È il daemon del servizio necessario per ricevere connessioni attraverso telnet.



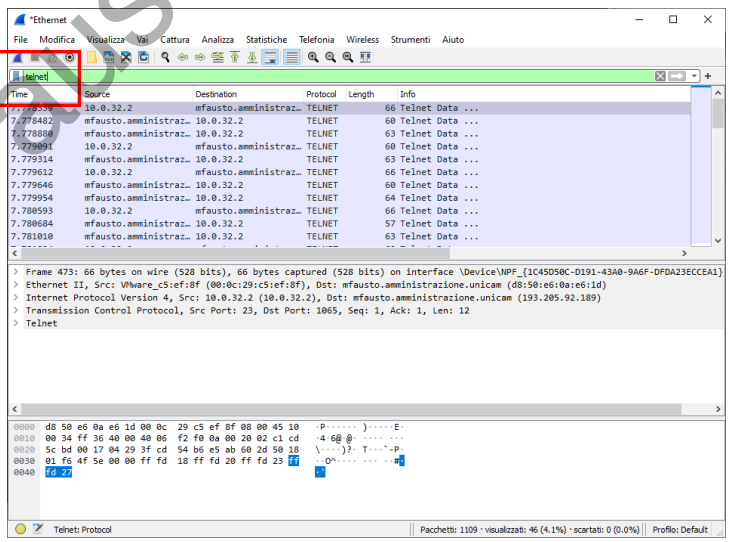
TCP porta 23

Client e Server negoziano le opzioni del collegamento (es., ASCII a 7 bit o a 8 bit)

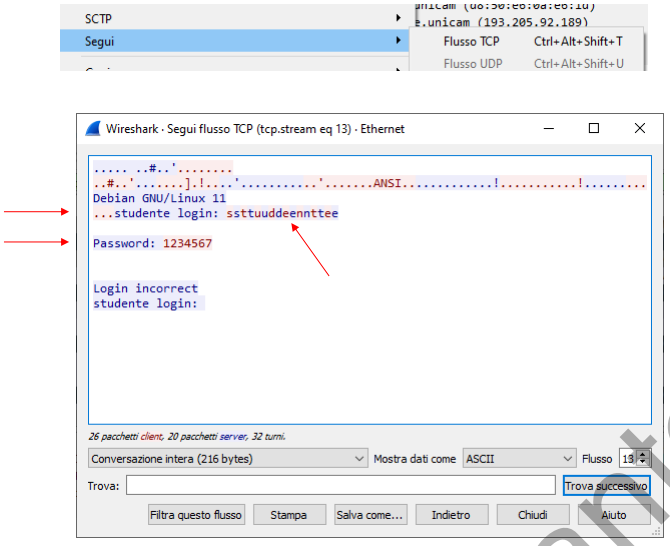
sintassi URI (Uniform Resource Identifier) per telnet (RFC4248)

telnet://<user>:<password>@<host>:<port>/

Wireshark <-> telnet



Wireshark <-> telnet



229

Remote Login - ssh

Ssh (Secure Shell)

Consente di:

- connettersi ad un altro sistema sulla rete.
- eseguire comandi su un sistema remoto.
- muovere files da un sistema all'altro sulla rete.

Ssh deve rimpiazzare completamente i comandi "r" (rlogin, rsh, rcp)

230

Remote Login - ssh

Sshd è il daemon che sta in attesa per le connessioni.

Normalmente resta in ascolto su una porta 22 TCP

fornisce un canale criptato sicuro su di una rete insicura.

Effettua:

- ✓ Autenticazione del server
- ✓ Scambio di chiavi
- ✓ Cifratura
- ✓ Protezione d'integrità

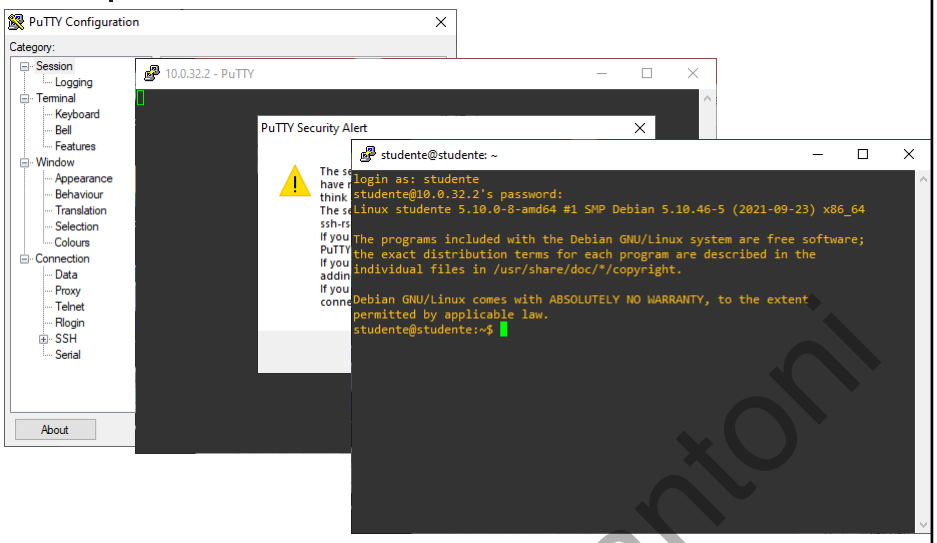
Remote Login - tools

https://www.putty.org/

https://winscp.net

Reti di Elaboratori

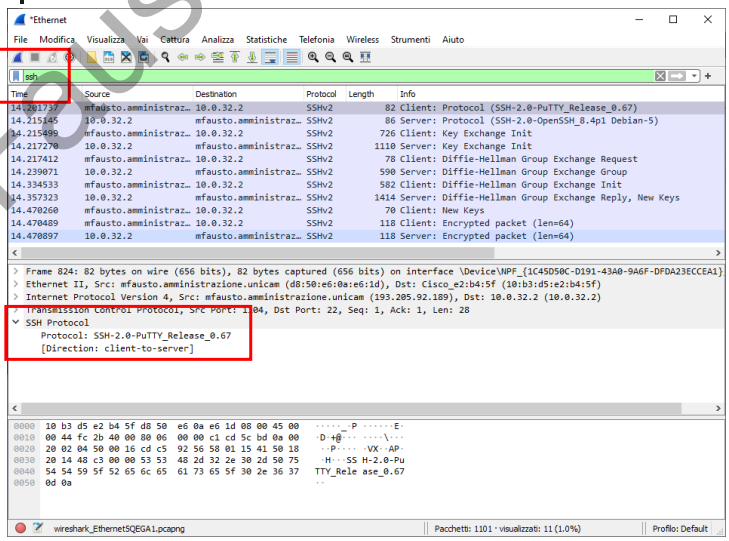
Connessione ssh



233

Reti di Elaboratori

Wireshark <-> ssh



234

Modello client - server

235

Modello client - server

Il server attende passivamente di essere contattato, mentre il client avvia il processo di comunicazione

236

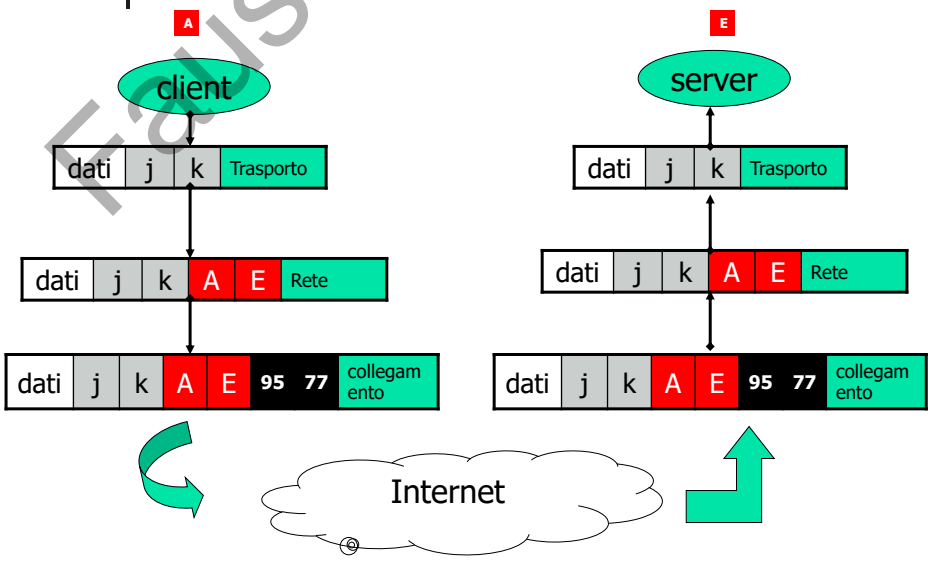
Reti di Elaboratori

Modello client - server

- Applicazione client
  - Generico programma
    - Si comporta da client quando richiede l'uso della rete
    - Esegue anche elaborazioni in locale
  - È invocata direttamente dall'utente
  - È eseguita localmente sul computer dell'utente
  - Contatta attivamente un solo server alla volta
  - Non richiede dispositivi dedicati
- Applicazione server
  - Programma specializzato
    - Dedicato all'erogazione di un servizio
    - Può servire più di un client alla volta
  - Automaticamente avviato all'avvio del sistema
  - Attende passivamente di essere contattato dai client
  - Accetta richieste da client arbitrari, ma offre un solo servizio
  - Calcolatore condiviso e macchina potente

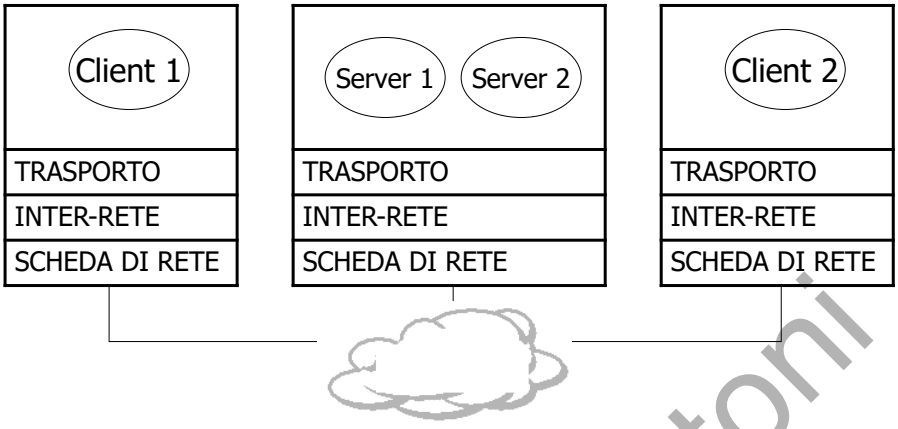
Reti di Elaboratori

Interazione tra client e server



Reti di Elaboratori

Servizi multipli



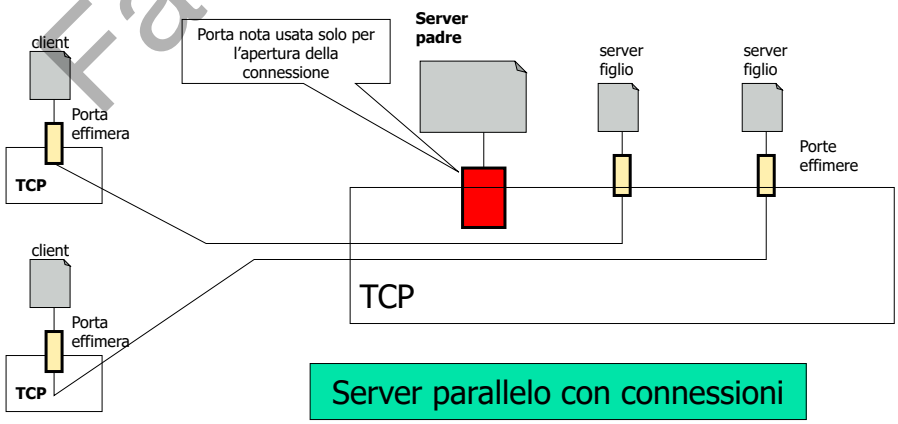
Un calcolatore di classe server può offrire contemporaneamente più servizi, ciascuno dei quali è erogato da una specifica applicazione server

239

Reti di Elaboratori

Creazione dinamica TCP (1)

Erogazione dello stesso servizio a più client senza che questi debbano mettersi in coda



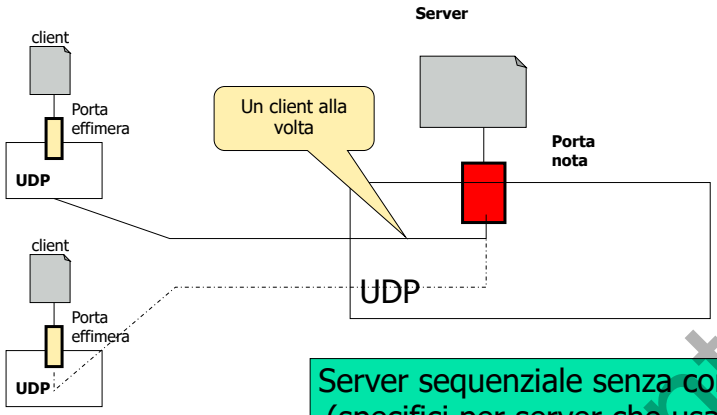
Server parallelo con connessioni

240

Reti di Elaboratori

Creazione dinamica UDP(2)

Erogazione dello stesso servizio a più client memorizzati in una coda in attesa di essere processati



Server sequenziale senza connessioni (specifici per server che usano UDP)

241

Reti di Elaboratori

Socket Programming

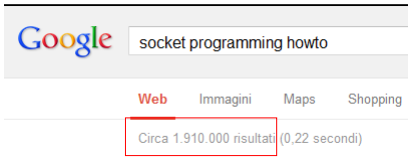
1. Le applicazioni client e server comunicano tramite i protocolli di trasporto
2. Le applicazioni debbono fornire al protocollo molte informazioni
   - API (Application Program Interface): insieme di procedure che le applicazioni possono invocare
3. API non è definita all'interno dei protocolli di comunicazione, ma parte dei sistemi operativi residenti
4. Interfaccia SOCKET standard de facto

1. Introdotto UNIX BSD 1981
2. Molti produttori aggiungono le socket ai loro sistemi
3. Librerie di procedure (Libreria delle Socket)
   - Mettono a disposizione delle applicazioni un API Socket anche se il sistema operativo sottostante non le prevede
   - Quando le applicazioni chiamano una procedura della libreria essa invoca le funzioni del proprio sistema operativo per ottenere l'effetto desiderato

242

Socket Programming

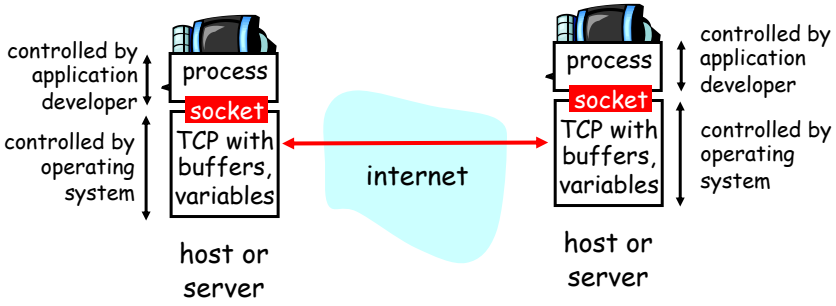
```
# crea un socket INET di tipo STREAM
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
# ora si connette al server web sulla porta 80 la normale porta http
s.connect(("www.unicam.it", 80))
```



Socket-programming using TCP

Socket: una interfaccia tra i processi delle applicazioni ed i protocolli di trasporto end to end

TCP service: trasferimento affidabile di dati tra un processo e l'altro

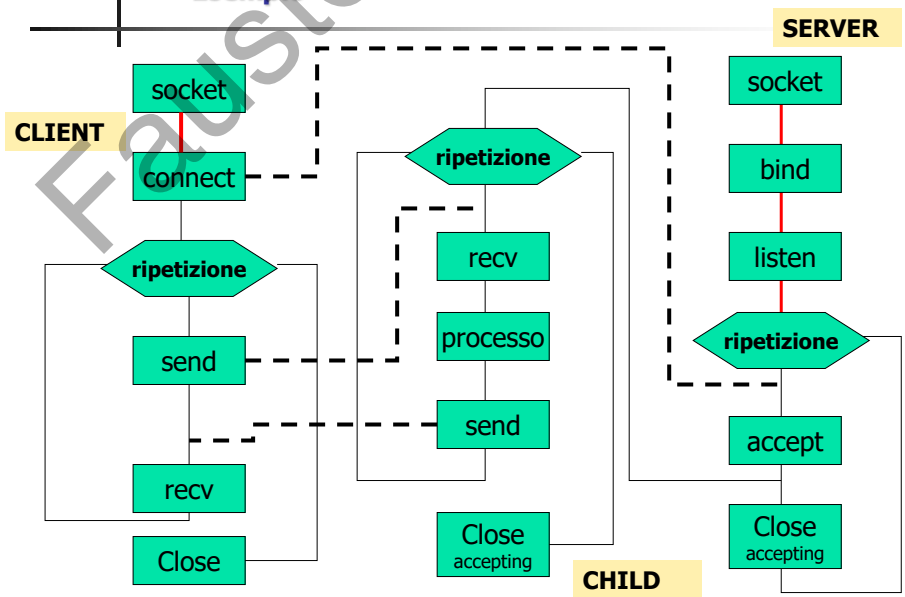


Socket programming with TCP

Client deve contattare il server
- Il processo server deve essere in esecuzione
- Il server deve avere creato il socket che è in grado di accettare il contatto del client
- Quando il client contatta il server :
  - Crea un client-local TCP socket
  - specifica IP address e port number del processo server

- Quando client crea il socket: il client TCP stabilisce una connessione con il server TCP
- Quando sarà contattato da un client, il server TCP crea un nouvo socket affinchè il processo server possa comunicare con il client
  - Questo permette al server di comunicare con multiple clients

Esempio



Reti di Elaboratori

Socket programming with UDP

UDP: connectionless tra client e server

- no handshaking
- sender explicitly attaches IP address and port of destination
- server must extract IP address, port of sender from received datagram

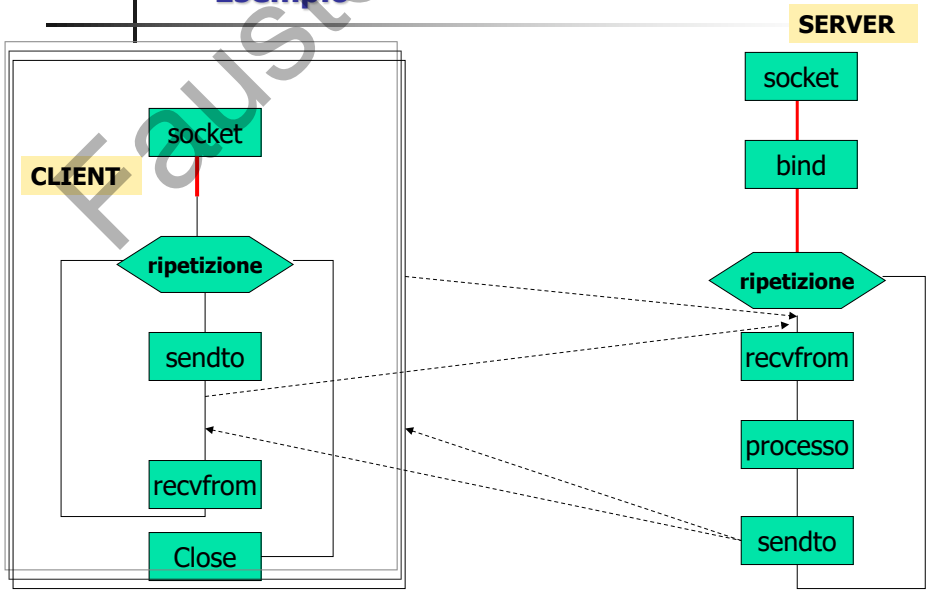
UDP: transmitted data may be received out of order, or lost

application viewpoint

*UDP provides **unreliable** transfer of groups of bytes ("datagrams") between client and server*

Reti di Elaboratori

Esempio



Applicazioni peer-to-peer

Peer: coppie di host connessi in modo intermittente, che comunicano direttamente l'uno con l'altro.

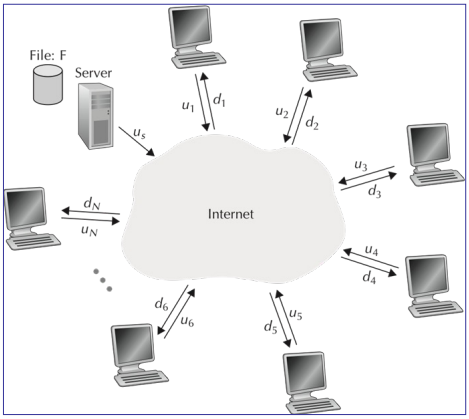
Distribuzione di file P2P (peer-to-peer)

In una distribuzione di file con P2P ciascun peer può ridistribuire agli altri qualsiasi porzione del file abbia ricevuto, aiutando in questo modo il server nel processo di distribuzione.

BitTorrent: uno tra i più diffusi protocolli di distribuzione di file tramite P2P.

Scalabilità dell'architettura P2P

Server e peer sono collegati a Internet con collegamenti di accesso.



BitTorrent

BitTorrent

L'insieme di tutti i peer che partecipano alla distribuzione di un particolare file è chiamato **torrent** (**torrente**).

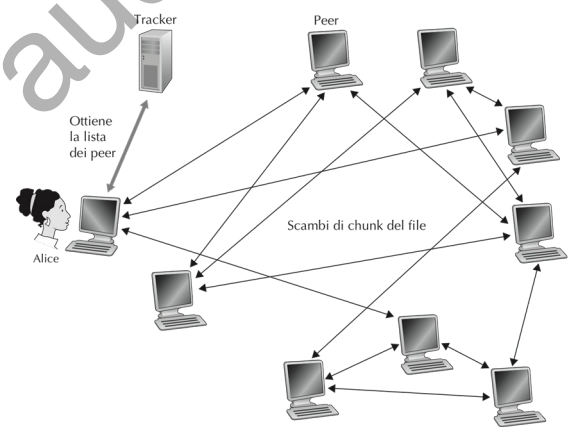
I peer in un torrent scaricano **chunk** (**parti**) del file di uguale dimensione uno dall'altro, con una dimensione tipica di 256 kbyte.

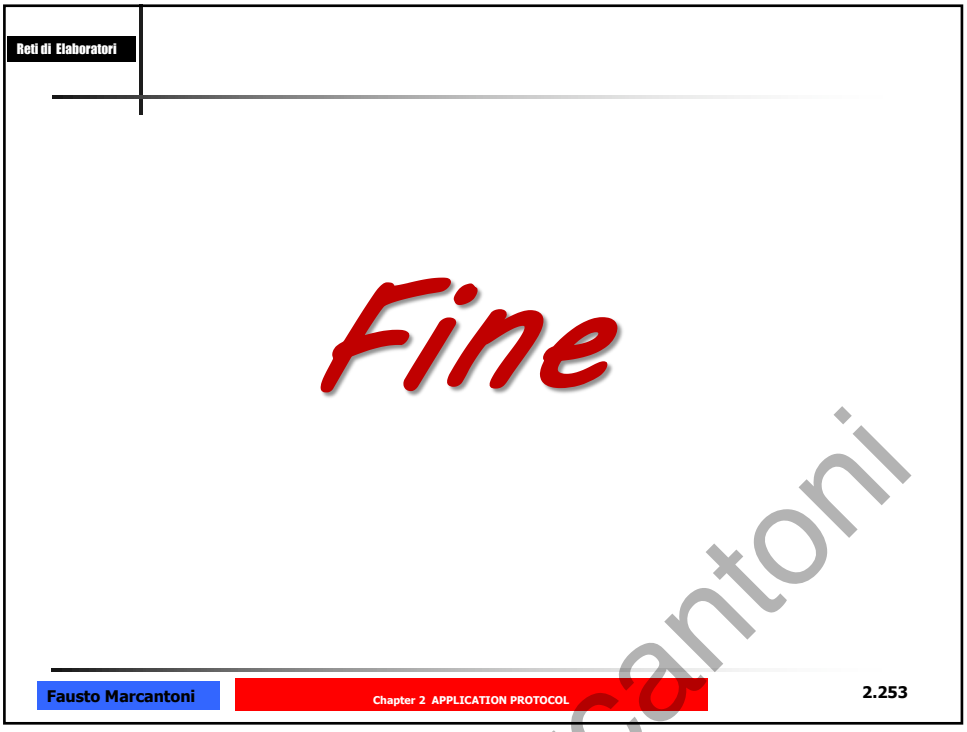
Ciascun torrent ha un nodo di infrastruttura chiamato: **tracker**.

Il meccanismo di incentivazione degli scambi descritto precedentemente viene spesso chiamato **tit-for-tat** *("***pan per focaccia***").*

L'ecosistema BitTorrent ha un grande successo con milioni di peer che si scambiano file simultaneamente in centinaia di migliaia di torrent.

BitTorrent





253