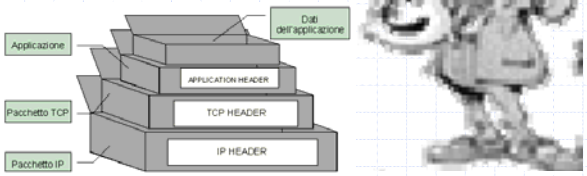
Analizzare il livello applicativo dell'architettura TCP/IP

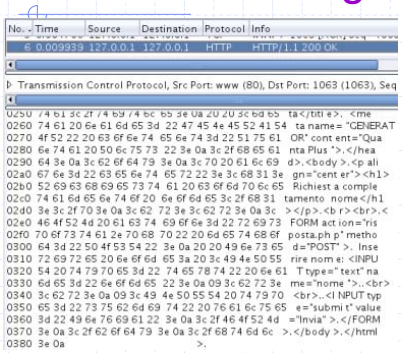
Dott. Libanore Luca

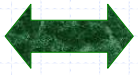
Livello 5: finalmente arriva l'Utente!



Vedremo ***come***, appoggiandosi sul livello di trasporto, e poi su tutti i livelli sottostanti della pila di protocolli, le ***applicazioni*** *interagiscono* per fornire i servizi all'**utente**

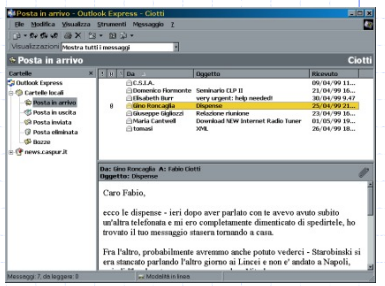
I protocolli e i servizi associati: uso il WWW grazie a HTTP





Vedremo come funzionano i **protocolli** (Es. HTTP) associati ai principali **servizi** (Es. WWW) utilizzati dagli utenti

Cos'è un servizio?



Un ***servizio*** è:

"qualsiasi prestazione fornita da ..... (ente pubblico, impresa privata, etc.) che serva a **soddisfare** un'esigenza della collettività (o di un gruppo di utenti)"

Cos'è un protocollo?

I protocolli

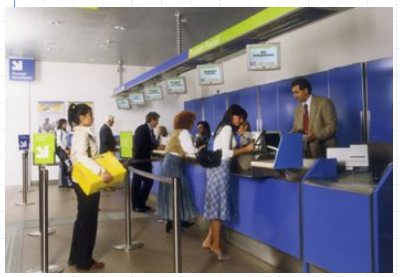
forniscono le **regole** per la comunicazione

1. Alzare il braccio per chiedere di parlare
2. Non si parla mai contemporaneamente
3. .....

contengono i dettagli dei **formati** dei **messaggi**

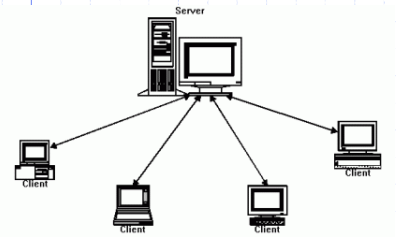
La grammatica della lingua (italiano, inglese, arabo, etc.) utilizzata per comunicare

Alcune regole dei protocolli di posta ordinaria



1. Le lettere vanno consegnate all'ufficio postale o in un'apposita buca delle lettere
2. Se il destinatario della lettera è un utente dell'ufficio postale allora la consegna avviene direttamente
3. Se il destinatario NON è un utente dell'ufficio postale allora la lettera viene inoltrata all'ufficio postale competente
4. .....

Alcune regole dei protocolli di posta elettronica

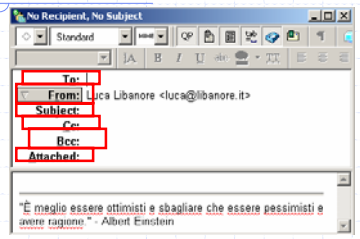


1. Le e-mail vanno spedite ad un server SMTP
2. Se il destinatario dell'e-mail possiede uno spazio su disco (mailbox) presso il medesimo server SMTP allora la consegna è immediata
3. Se il destinatario NON possiede una mailbox sul server SMTP allora l'e-mail viene inoltrata al server SMTP competente
4. .....

Alcuni aspetti del formato dei messaggi del protocollo di posta ordinaria

DIMENSIONI (*)	PICCOLO	MEDIO
Altezza (mm)	120	250
Lunghezza (mm)	235	353
Spessore (mm)	5	25
Peso (g)	50	2000

Alcuni aspetti del formato dei messaggi del protocollo di posta elettronica



"Lines in a message MUST be a maximum of 998 characters excluding the CRLF, but it is RECOMMENDED that lines be limited to 78 characters excluding the CRLF."

Tratto dal RFC 2822

"La dimensione massima di una e-mail che può essere inviata utilizzando una casella di posta registrata su un contratto Tin.it Free non può superare i 3 MB"

Tratto da http://help.virgilio.it/assistenza/index.jsp?id=6080

Quindi il protocollo come...

Insieme del

come comportarsi (regole)

e del

come devono essere i messaggi che ci si scambia (formato)

I servizi non sono realizzati con tecniche particolarmente difficili!

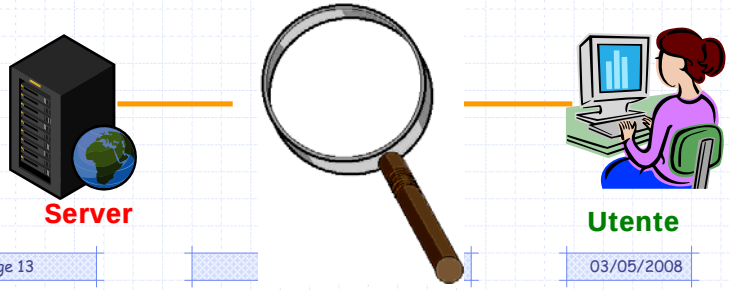
I protocolli che andremo a vedere sono prevalentemente dei protocolli ASCII → gli applicativi, appoggiandosi su una connesione gestita dal livello di trasporto, si scambiano dei **messaggi/comandi testuali**

Che protocolli/servizi andremo a vedere?

- Protocollo DNS → Servizio: risoluzione di nomi di host in indirizzi IP
- Protocolli SMTP, POP3, IMAP → Servizio: Posta elettronica
- Protocollo FTP → Servizio: trasferimento file
- Protocollo HTTP → Servizio: il WWW
- FTP vs. HTTP → Servizio: il download di file
- Streaming vs. Podcast

Come studieremo questi protocolli?

Utilizzando un analizzatore di pacchetti, ovvero uno strumento di ”sniffing” che ci permette di “catturare” il traffico che viaggia su una rete, memorizzandolo in un file

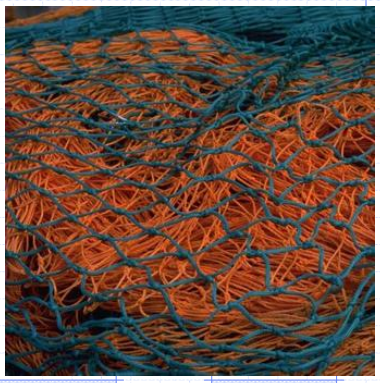


Come utilizzeremo Wireshark?

- Catturando delle interazioni fra un client e un server
- I file sono in formato tcpdump (nota utility di origine Unix)
- L’analizzatore che utilizzeremo è Wireshark (www.wireshark.com)

A cosa serve e come funziona un network sniffer?

- Come detto è un programma che serve a *catturare* pacchetti entranti/uscenti dall'interfaccia di rete di una workstation
- Lo scopo di questo strumento e ***analizzare il traffico, valutare le prestazioni di una rete, ed individuare possibili anomalie***



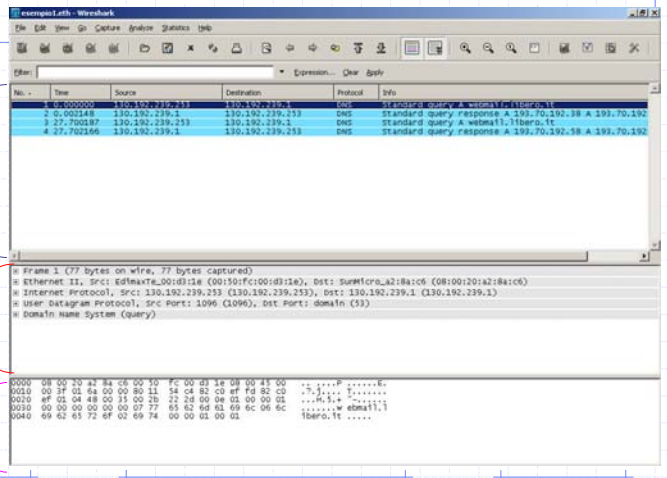
Sniffer e sicurezza

- È possibile, utilizzando uno sniffer, *analizzare il traffico in modo del tutto invisibile* agli altri calcolatori → gli altri utenti, in presenza di uno sniffer, non si accorgono di questo
- Con lo *sniffer, è possibile effettuare la cattura di informazioni sensibili*, tipo password (es. POP3), contenuto in chiaro dei messaggi di posta elettronica → SOLUZIONE: usare servizi di rete che fanno uso di tecnologia sicura, come IPSec (livello 3) o SSL (livello 4)

Le funzionalità di Wireshark

- Wireshark è uno degli sniffer più utilizzati (prima si chiamava Ethereal)
- E' uno strumento di tipo Open Source
- Permette la cattura del traffico su interfacce di tipo ethernet ma non solo
- E' possibile anche effettuare la cattura su qualunque interfaccia di rete disponibile su un calcolatore (es. Adattatore USB Wireless)

Interfaccia di Wireshark



Partiamo subito!

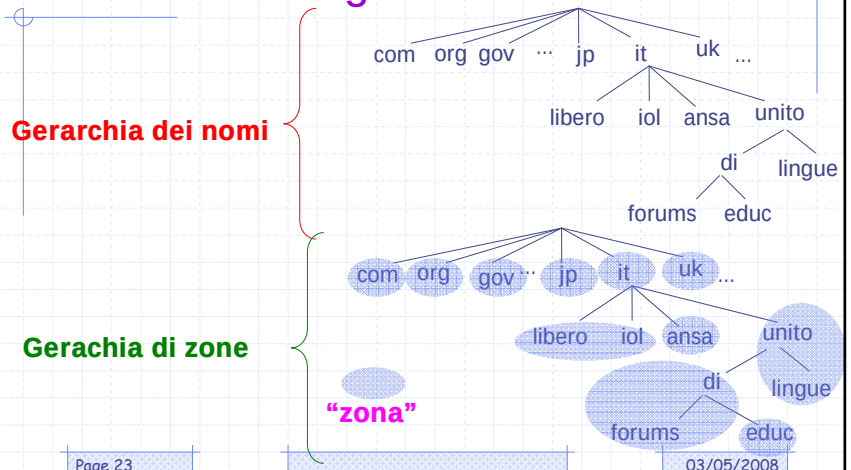
Il Domain Name System (DNS)

Il DNS permette di...

utilizzare delle ***stringhe di caratteri ASCII*** di tipo mnemonico, più facili da ricordare e più utili **PER L'UTENTE**

Come funziona il DNS?

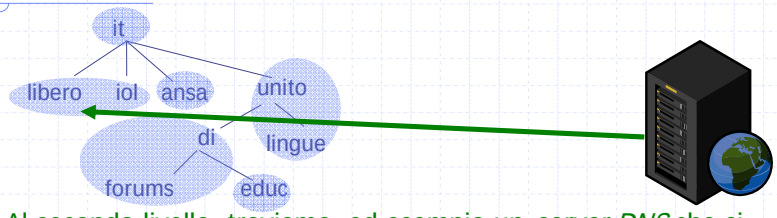
La struttura gerarchica di zone



Esistono le zone DNS per una questione di responsabilità

- ***Una zona DNS è una parte dello spazio dei nomi,*** che è sotto una ***stessa gestione amministrativa*** e quindi è gestita da uno o più server
- ***Una zona DNS è costituita da un dominio e dai suoi sottodomini che non sono a loro volta delegati*** (potrebbero esserci sottodomini esclusi dalla zona perché delegati/affidati ad un'altra zona, vedi l'esempio di educ!)

Vediamo da vicino come i server DNS sfruttano la gerarchia a zone

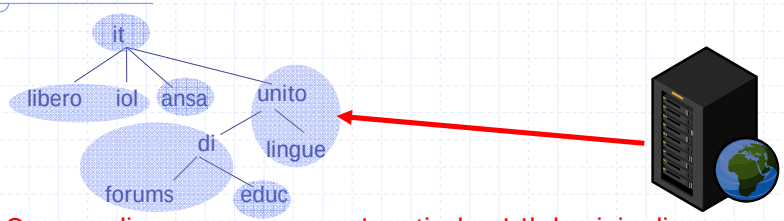


Al secondo livello, troviamo, ad esempio un *server DNS* che si occupa di tutti i *resource record* riguardanti sia *libero* che *iol*; il server DNS è lo stesso per entrambi i domini; se al server DNS viene richiesta un'informazione riguardante libero o iol risponderà direttamente lui, se invece la richiesta riguarda, ad esempio, unito allora inoltrerà la query al server di competenza, dato che unito non fa parte della sua zona

Ma i server che sono competenti per una zona come si chiamano? AUTORITATIVI

- Si dice che il ***server DNS è AUTORITATIVO per una zona, quando contiene i resource record ad essa relativi***
- Quando ad un server DNS viene fatta una richiesta per un nome di cui NON è AUTORITATIVO, si connetterà con altri DNS di altre zone per reperire informazioni relative a nodi remoti

E per unito cosa succede?



Ora guardiamo un caso un po' particolare! Il dominio di secondo livello **unito** ha due sottodomini (in realtà ne ha molti di più!): ***lingue***, che fa parte della stessa zona di unito, e ***di*** che fa parte di un'altra zona; ci sarà quindi un server DNS che contiene i resource record di ***unito*** e del suo sottodominio ***lingue*** MA NON del sottodominio ***di***, che sarà di competenza di un altro server DNS

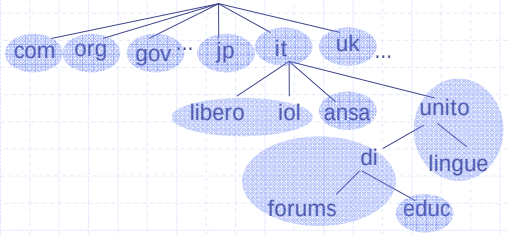
Un esempio pratico sul funzionamento del DNS

Supponiamo che io sia un utente di Libero

Questo vuol dire che Libero, oltre a fornirmi un collegamento ad Internet mi offrirà anche altri servizi, tra cui:

- Una casella di posta elettronica
- Un server DNS di riferimento a cui rivolgere le mie query DNS
- Etc....

Io sono un cliente di Libero e voglio vedere il sito www.educ.di.unito.it



Vediamo i vari passi che ci portano alla risoluzione della Query DNS

Primo passo: interrogo il server DNS di Libero

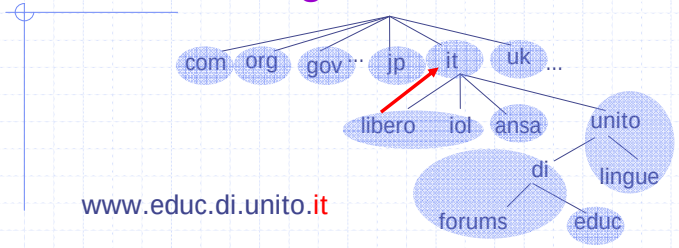
Server DNS di Libero

"Qual è l'indirizzo IP di www.educ.di.unito.it?"

Il server DNS di Libero non è competente/autoritativo/di fiducia per l'indirizzo www.educ.di.unito.it

Se però conosce l'associazione NOME/INDIRIZZO IP (c'è l'ha in cache) ci risponde, dicendo che la sua non è una risposta di fiducia... altrimenti?

Secondo passo: inizia la risalita della gerarchia!

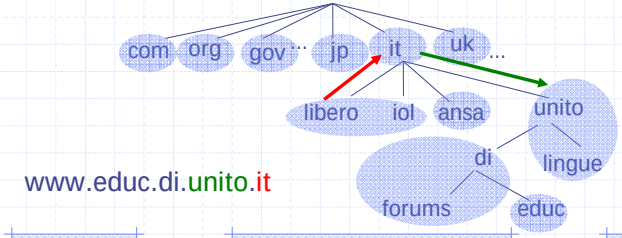


Se il server DNS di Libero non ha l'associazione nel suo database inizia ad analizzare il nome e nota che questo termina con .it

Dopo aver fatto questa osservazione, a sua volta, ***va a interrogare*** un server DNS che risponde per gli indirizzi .it

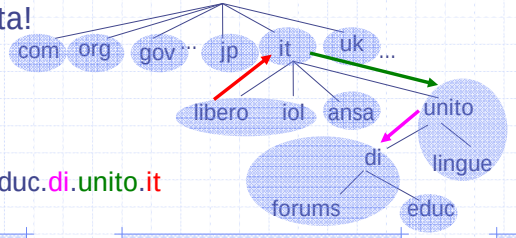
Terzo passo: il server .it conosce il nome ma non è competente per rispondere!

Il server .it conosce il nome e sa che esiste, però nota anche lui non è competente a rispondere, perché ha delegato il compito a qualcun altro! Ma proprio perché ha delegato il compito, sa a chi l'ha delegato e gli ***inoltra*** la richiesta!



Quarto passo: il server di unito.it conosce il nome ma non è competente per rispondere!

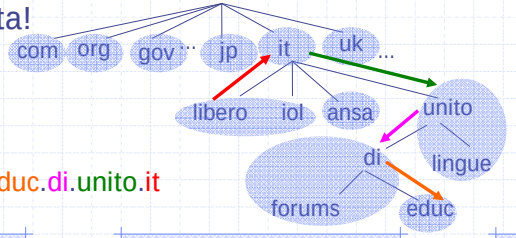
Il server unito.it conosce a sua volta il nome e sa che esiste, però nota anche lui non è competente a rispondere, perché ha delegato il compito a qualcun altro! Ma proprio perché ha delegato il compito, sa a chi l'ha delegato e gli ***inoltra*** la richiesta!



www.educ.di.unito.it

Quinto passo: il server di di.unito.it conosce il nome ma non è competente per rispondere!

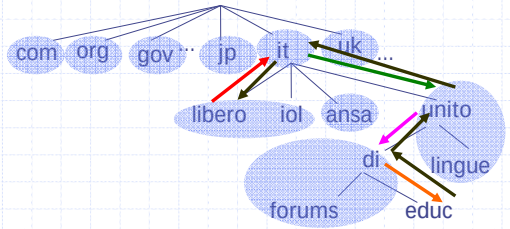
Il server di.unito.it conosce a sua volta il nome e sa che esiste, però nota anche lui non è competente a rispondere, perché ha delegato il compito a qualcun altro! Ma proprio perché ha delegato il compito, sa a chi l'ha delegato e gli ***inoltra*** la richiesta!



www.educ.di.unito.it

Sesto passo: finalmente il server autoritativo/competente

Il server educ.di.unito.it conosce il nome ed è il server competente per rispondere! Guarda nel suo database e *inoltra la risposta all'indietro*!



Come funziona una risoluzione di nome

In generale, per ottenere la risoluzione di un nome è necessario partire dalla radice, (1) interrogare uno dei root servers nel dominio di primo livello, ottenere il server che lo gestisce, (2) interrogarlo nel dominio di secondo livello, fino a (3) raggiungere il server autoritativo per il nome desiderato

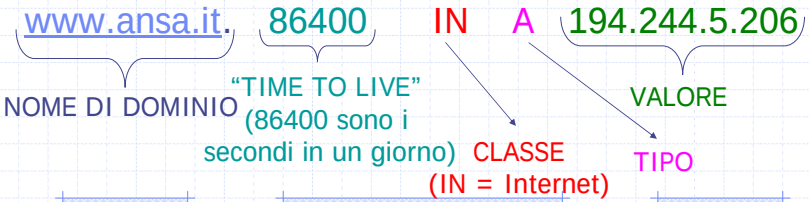
Questa tecnica è detta "***ricorsione***"

Piccoli aiuti per migliorare il sistema: il caching dei record DNS

- Alcuni ***server*** si prestano ad effettuare query ricorsive per conto di alcuni client
- Una volta che hanno ***ottenuto*** una ***risposta, memorizzano in una cache*** tutte le informazioni che hanno imparato, fino alla loro scadenza
- In questo modo diminuisce il traffico su rete e soprattutto la risposta è più rapida!

Ma il DNS, alla fine della fiera, cosa e come memorizza?

Il DNS mette in associazione questi nomi con degli indirizzi, utilizzando una struttura di registrazione delle informazioni (chiamato *descrittore di risorsa* o RESOURCE RECORD) che ha questo aspetto:



I descrittori, però, possono essere di diversi tipi, tra cui...

NS	Nome del server	Quando si vuol sapere solo il Nome del server di dominio e non tutti gli altri parametri di zona si usa questo campo al posto di SOA
A	Address	Indirizzo IP dell'host
MX	Mail exchange	La priorità e il nome con cui il dominio desidera accettare la posta elettronica
CNAME	Canonical Name	Utilizzato per creare alias di nomi di dominio

Proviamo a capire meglio quali informazioni contengono alcuni di questi tipi

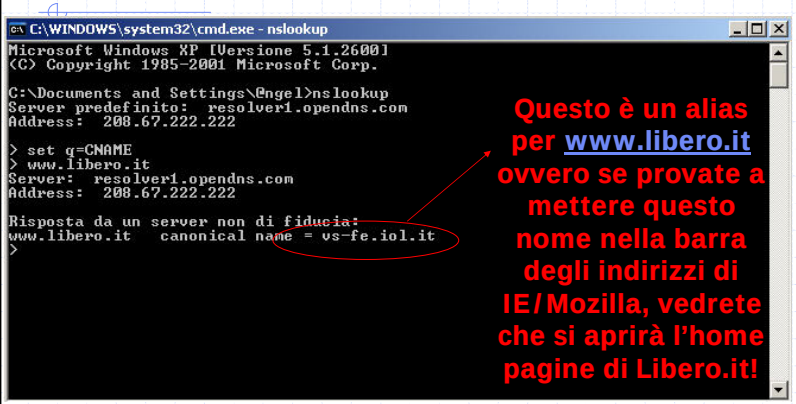
- Il tipo **NS** contiene il *nome del server di dominio ed eventualmente il suo indirizzo di rete*
- Il tipo MX contiene informazioni sul dominio di posta elettronica, quindi *come deve essere instradata la posta elettronica* (in particolare con la *possibilità* di avere *più alternative* nel caso in cui un server di posta elettronica non si accessibile)
- Il tipo CNAME, contiene degli alias, ovvero permette di avere nomi diversi che si riferiscono allo stesso servizio/server (Es. Il CNAME di www.fbi.gov è fbi.edgesuite.net e sono entrambi lo stesso server WEB); non tutti i nomi di domini hanno associato un CNAME;

Nslookup: il client DNS di Windows!

- Proviamo ad utilizzare un client DNS, quello messo a disposione da Windows
- E' un client testuale
- Per usarlo:
- Digitare nslookup da interfaccia MS-DOS
- Comandi da utilizzare:
  - Set q=TIPO
  - Server x.x.x.x (indirizzo di rete del server DNS che si vuol interrogare, altrimenti usa il server che è indicato nella impostazioni di rete di Windows)
- Lo stesso client c'è anche in Linux e si usa quasi nello stesso modo (provare per credere!)

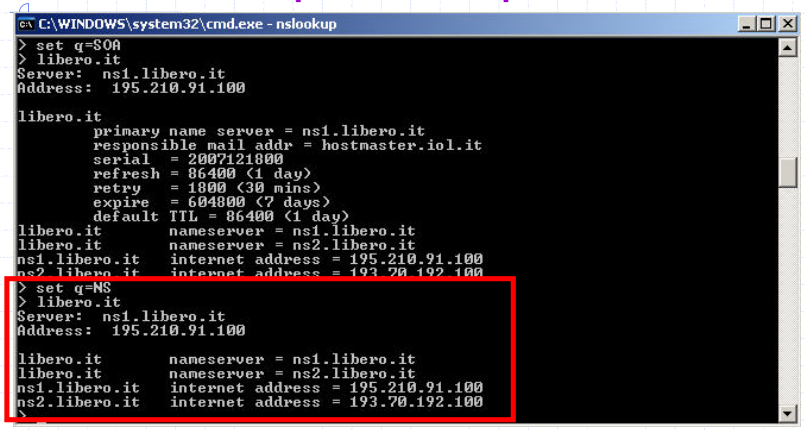
Vediamo un'interfaccia d'esempio di nslookup

Esempio 1: vediamo un esempio di CNAME



Questo è un alias per www.libero.it ovvero se provate a mettere questo nome nella barra degli indirizzi di IE/Mozilla, vedrete che si aprirà l'home pagine di Libero.it!

Esempio 2: il tipo NS



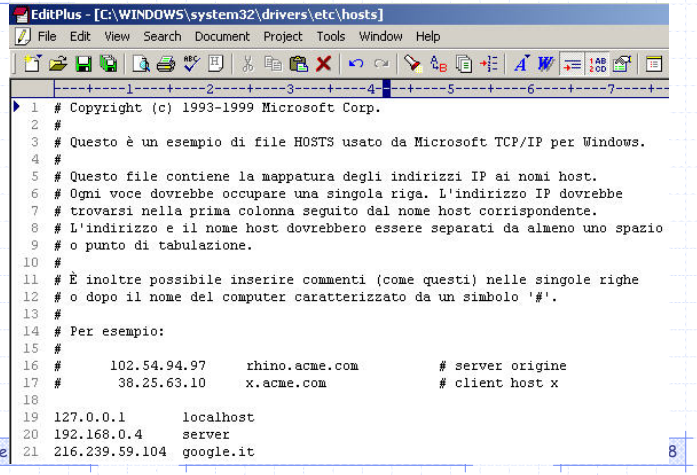
Ma, noi, come utilizziamo i campi CNAME, A e MX?

Facciamo alcune prove!

Provate a risolvere le seguenti query DNS:

- Chiedere a che indirizzo corrisponde il nome www.google.it
- Chiedere qual è l'alias di www.libero.it
- Chiedere qual è l'alias di www.iol.it
- Chiedere qual è il mail server di gmail.com

Curiosità dal DNS: il file hosts in Windows XP SP2



Date un'occhiata al sito **hpHosts**

- http://www.hosts-file.net/
- E' interessante perché propone dei file ***hosts*** di diverso genere
- Sono file ***hosts*** utili per ambiente Windows/MacOS/Linux
- Proviamo a vedere insieme il contenuto di uno di questi file ***hosts***! Secondo voi, in questo caso, qual è l'utilità di questo file?
- Normalmente i file hosts presenti nei nostri PC servono per definizioni locali e caching

Provate ora a risolvere le seguenti query

Attivate Wireshark e configuratelo in modalità cattura

Ora con nslookup sottoponiamo al nostro server DNS di riferimento le seguenti interrogazioni:

- A quale indirizzo corrisponde il nome www.google.it
- A quale indirizzo corrisponde il nome www.libanore.it
- Facciamo un po' di osservazioni..

esempio1.eth: Ricerca di indirizzi per webmail.libero.it

- In questo file si può notare che vengono ritornati più di un tipo A
- Il DNS usato ritorna anche i nomi dei DNS server autorevoli, ma non i relativi indirizzi; questo comportamento non è ottimo, perché costringe il client a fare un'altra query se vuole contattare i server autorevoli
- Se si esegue una seconda query, l'ordine con cui sono ritornati gli indirizzi è diverso; come mai?

esempio2.eth: e se contatto direttamente il DNS server di libero.it?

- Se invece interroghiamo il DNS di libero.it possiamo avere l'indirizzo (Additional Records) di ambedue i DNS di libero.it

esempio3.eth: Ricerca del Name Server di libero.it effettuata sul name server di .it

- Un DNS del dominio di .it risponde con gli indirizzi dei due DNS di libero.it, ma dice anche che la risposta non è autorevole, perché dal punto di vista di principio gli unici autorevoli per il dominio libero.it sono appunto i DNS del dominio
- Però in questo modo chi cerca i DNS di un dominio li può scoprire e può contattarli per avere informazioni autorevoli

Un pò di esercizi...

Utilizzando nslookup come client DNS e 151.99.125.2 come server iniziale (vedi slide 42):

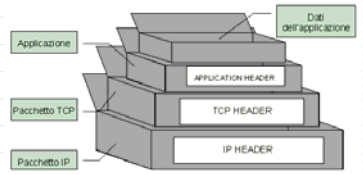
1. Provare a cercare un nome non esistente; che risposta viene fornita dal server DNS? Analizzatela molto attentamente!
2. Ricerca del Name Server di libero.it effettuata sul name server di libero.it
3. Ricerca del Name Server di libero.it effettuata sul name server di .it. (Come si fa a scoprire chi è?)
   - Che differenza c'è tra la risposta dell'esercizio 2 e 3?

Catturare ed analizzare tutto il traffico generato utilizzando wireshark!

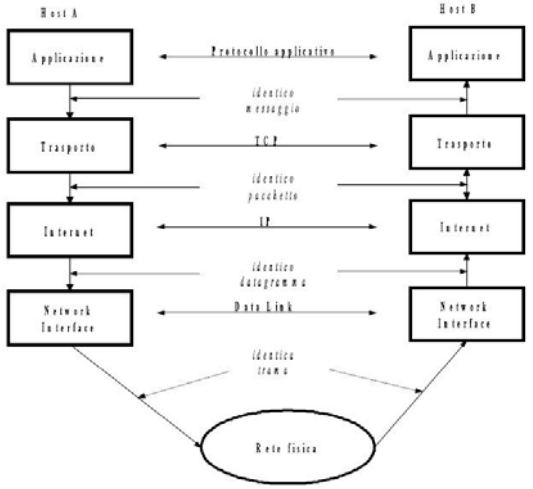
Osservazioni sugli esercizi 2 e 3

- Come si vede un DNS del dominio di .it risponde con gli indirizzi dei due DNS di libero.it, ma dice anche che la risposta non è autorevole, perché dal punto di vista di principio gli unici autorevoli per il dominio libero.it sono appunto i DNS del dominio
- Però in questo modo chi cerca i DNS di un dominio li può scoprire e può contattarli per avere informazioni autorevoli
- ***I DNS del dominio "superiore" devono conoscere il nome e l'indirizzo dei DNS di domini sottostanti che sono sotto la loro autorità.***

E gli altri livelli della pila TCP/IP?



Analizziamo lo stack TCP/IP



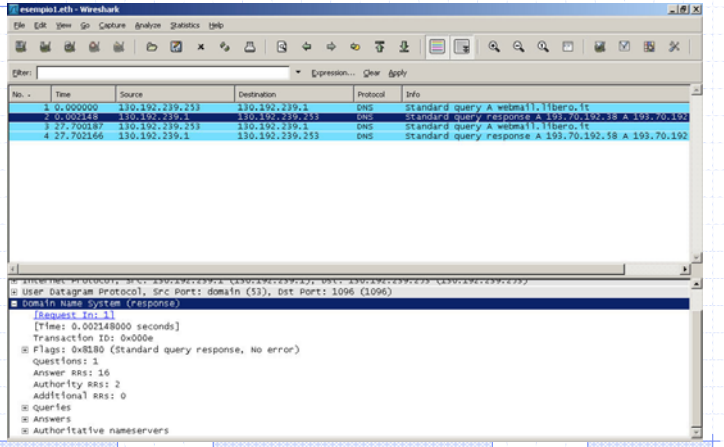
Partiamo dal DNS

0	16	31
IDENTIFICATION	PARAMETER	
NUMBER OF QUESTIONS	NUMBER OF ANSWERS	
NUMBER OF AUTHORITY	NUMBER OF ADDITIONAL	
QUESTION SECTION ...		
ANSWER SECTION ...		
AUTHORITY SECTION ...		
ADDITIONAL INFORMATION SECTION ...		

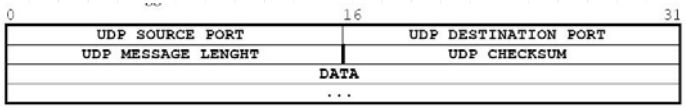
Quali sono i bit del campo PARAMETER

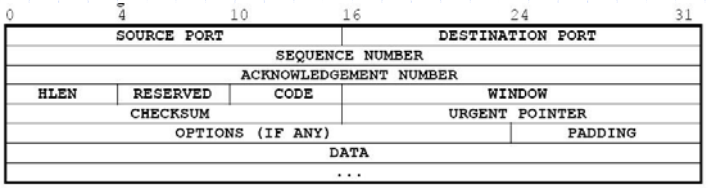
Bit of PARAMETER field	Meaning
0	Operation: 0 Query 1 Response
1-4	Query Type: 0 Standard 1 Inverse 2 Completion 1 (now obsolete) 3 Completion 2 (now obsolete)
5	Set if answer authoritative
6	Set if message truncated
7	Set if recursion desired
8	Set if recursion available
9-11	Reserved
12-15	Response Type: 0 No error 1 Format error in query 2 Server failure 3 Name does not exist

esempio1.eth: il livello applicativo

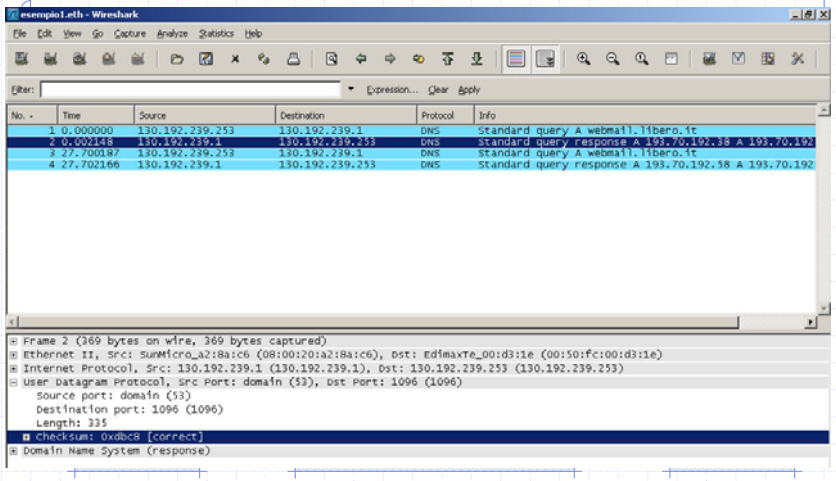


Scendiamo al livello di trasporto: UDP e TCP

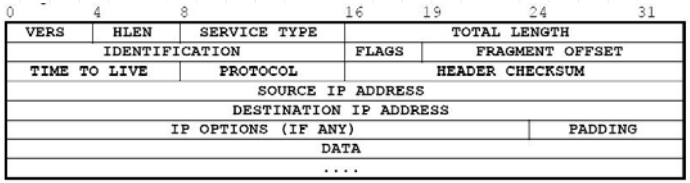




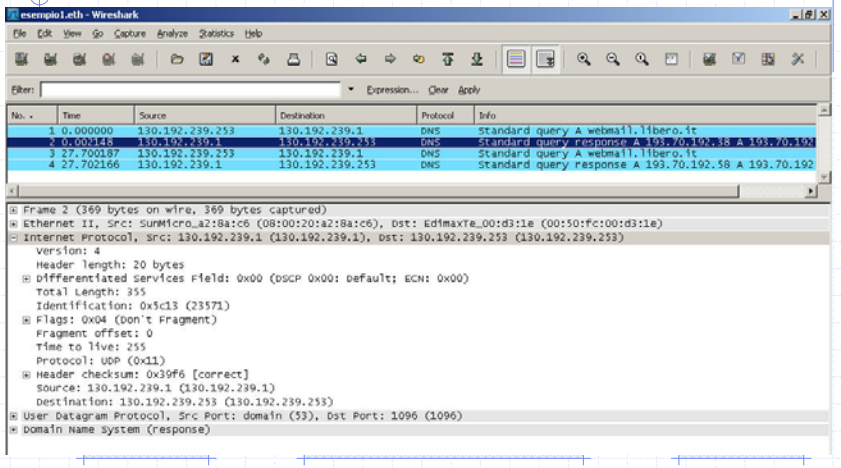
esempio1.eth: il livello di trasporto



Livello Internet: IP



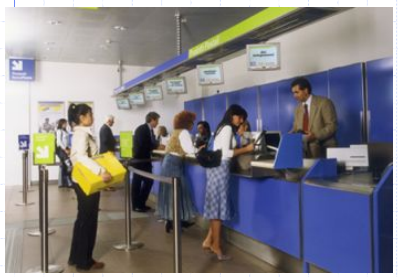
esempio1.eth: il livello Internet



Il sistema di posta elettronica

Dott. Libanore Luca

Alcune regole dei protocolli di posta ordinaria



1. Le lettere vanno consegnate all'ufficio postale o in un'apposita buca delle lettere
2. Se il destinatario della lettera è un utente dell'ufficio postale allora la consegna avviene direttamente
3. Se il destinatario NON è un utente dell'ufficio postale allora la lettera viene inoltrata all'ufficio postale competente
4. .....

Come avviene il trasferimento dei messaggi dal mio computer a quello di un mio amico?

Il trasferimento dei messaggi di posta elettronica si realizza:

1. Utilizzando SMTP (Simple Mail Transfer Protocol) per il trasferimento dei messaggi tra i server di posta elettronica
2. Utilizzando POP3 (Post Office Protocol versione 3) o IMAP4 (Internet Message Access Protocol) per il trasferimento dei messaggi dal server di posta al client

Vediamo un pò di caratteristiche del protocollo SMTP

Il protocollo SMTP risponde sulla porta 25.

È un protocollo di tipo ASCII, ovvero il ***dialogo fra i vari server*** di posta elettronica avviene ***tramite comandi di tipo ASCII***

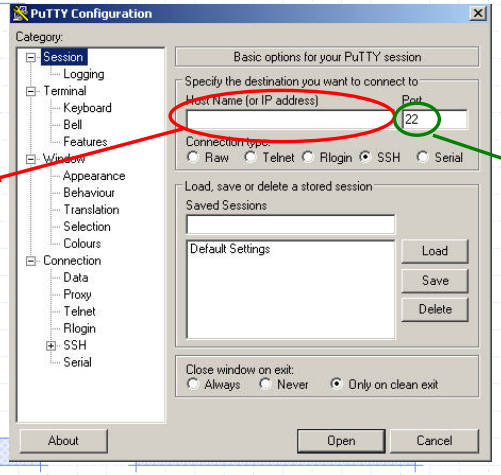
Ecco alcuni dei comandi ASCII che si scambiano i server SMTP

HELO	Dal client al server seguito dall'indirizzo DNS del client (è un abbreviazione dell'HELLO)
MAIL FROM:	Nella composizione di un nuovo messaggio indica il mittente (user@mailserver)
RCPT TO:	"recipient to": destinatario del messaggio (una linea RCPT TO per ogni destinatario)
DATA	Precede il messaggio vero e proprio, comprensivo di busta, intestazione e corpo
QUIT	Chiude la connessione

Vediamo ora come si comporta l'SMTP per la spedizione di un messaggio di posta elettronica

- Per provare il funzionamento di questo protocollo ASCII utilizzeremo l'applicativo TELNET, chiamando però il server non sulla normale porta telnet ma sulla porta 25
- A tal fine utilizzeremo PuTTY, un client telnet/ssh grafico gratuito, utilizzabile sia in piattaforma Windows che Unix (www.chiark.greenend.org.uk/~sgtatham/**putty**/)

Com'è fatto PuTTY?

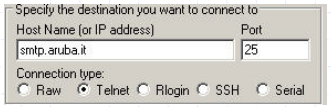


In questo campo si inserisce il nome, o indirizzo IP, del server da contattare

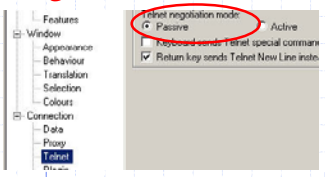
In questo campo si inserisce la porta del server; PuTTY può essere utilizzato con tutti i protocolli ASCII (SMTP, POP3, etc.)

Inseriamo le impostazioni in PuTTY

Server SMTP: smtp.aruba.it su porta TCP 25

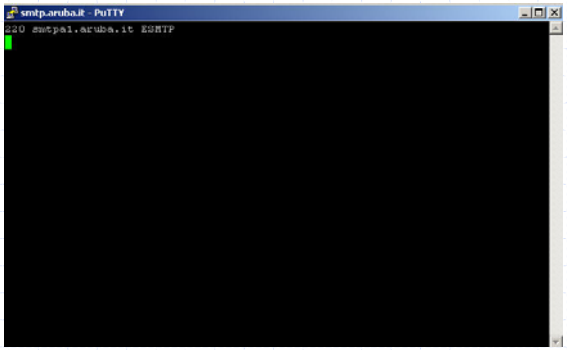


Piccola raccomandazione: configurare come passiva la *negoziazione* Telnet

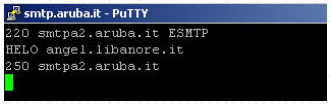


E una volta stabilita la connesione...

- Il server SMTP si mette in attesa!

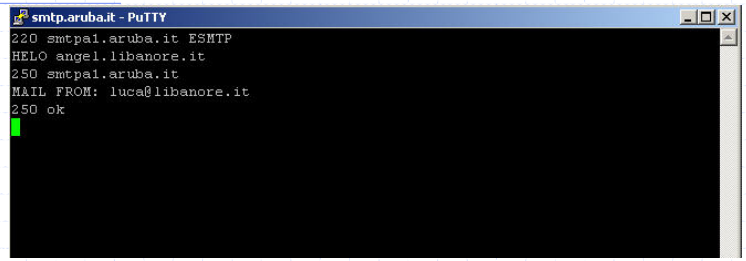


Noi lo salutiamo dicendogli il nome della nostra macchina (o quantomeno l'indirizzo di rete!)



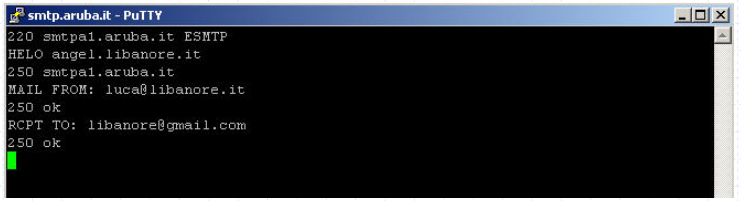
- Questo è il primo comando da fornire al server SMTP per iniziare una sessione di posta elettronica
- Una volta riconosciuti (alcuni server SMTP non permettono di continuare la sessione se non riconoscono la macchina o IP) possiamo creare un nuovo messaggio

Il primo comando per creare un nuovo messaggio è MAIL FROM:



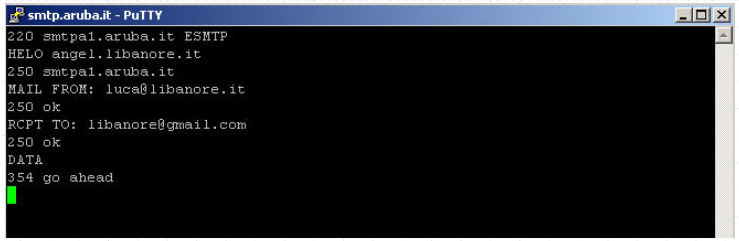
In questo modo ho specificato il mittente del messaggio, ora andiamo a inserire il destinatario

Il comando RCPT TO:



Molti server SMTP non accettano la creazione di un nuovo messaggio di posta elettronica se il destinatario non è un loro cliente! Perchè? Lo vedremo dopo!

A questo punto possiamo utilizzare il comando DATA per creare il corpo del messaggio



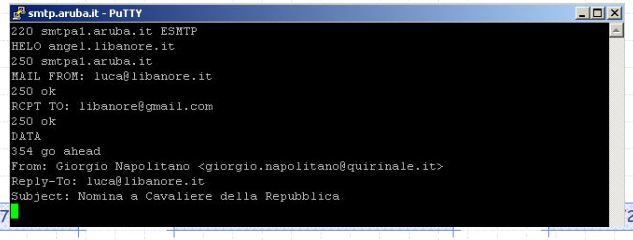
Inizieremo specificando alcuni campi, ovvero quelli presenti NEL RFC 822

Alcune osservazioni sul corpo del messaggio e sulla sua interpretazione

L'indirizzo a cui verrà spedito il messaggio è ormai già stato costruito nel comando RCPT TO: ma ***adesso avremo una serie di campi che verranno interpretati dal client*** che riceverà il messaggio, per spiegare all'utente da dove arriva il messaggio, qual è il soggetto e così via.

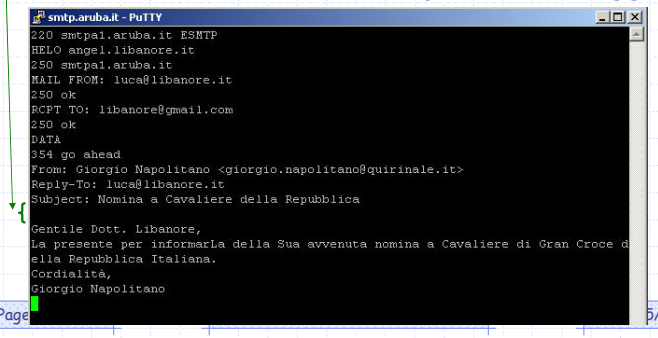
Iniziamo a creare il messaggio, partendo dalle intestazioni

- Per prima cosa l'indirizzo del mittente: per esempio scriveremo che il nostro indirizzo mittente sarà sergio.napolitano@quirinale.it bleffando per ingannare il destinatario!
- Poi possiamo dare un subject che ci ritorna utile (la nostra famosa nomina!)

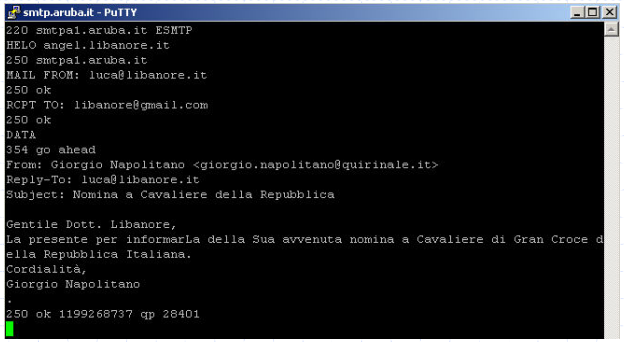


Ora concentriamoci sul corpo del messaggio vero e proprio!

Ora passiamo a scrivere alcune righe del messaggio, ricordandoci però che dobbiamo inserire ***una riga vuota*** tra le intestazioni e il corpo del messaggio!



Concludiamo il messaggio con un . e via!



Il server ci dice che il messaggio è stato accettato e lo invierà!

Vediamo ora il nostro messaggio arrivato a destinazione!

Delivered-To: libanore@gmail.com
Authentication-Results: mx.google.com; spf=neutral (google.com: 62.149.128.211 is neither permitted nor denied by best guess record for domain of luca@libanore.it) smtp.mail=luca@libanore.it
Date: Wed, 02 Jan 2008 02:12:20 -0800 (PST)
From: Giorgio Napolitano <giorgio.napolitano@quirinale.it>
Reply-To: luca@libanore.it
Subject: Nomina a Cavaliere della Repubblica
X-Spam-Rating: smtpa1.aruba.it 1.6.2 0/1000/N

Gentile Dott. Libanore,
La presente per informarLa della Sua avvenuta nomina a Cavaliere di Gran Croce della Repubblica Italiana.
Cordialità,
Giorgio Napolitano

Come vediamo il mittente, per il client di posta, è proprio quello che abbiamo creato noi!

Vediamo una serie di intestazioni: *alcune sono quelle che abbiamo creato noi* (Reply-To, From e Subject) mentre *altre sono state aggiunte dai vari mailer attraversati* (*come lo stesso server di Aruba*)

Ma questa tecnica funziona con tutti i server SMTP? Purtroppo NO, anzi..

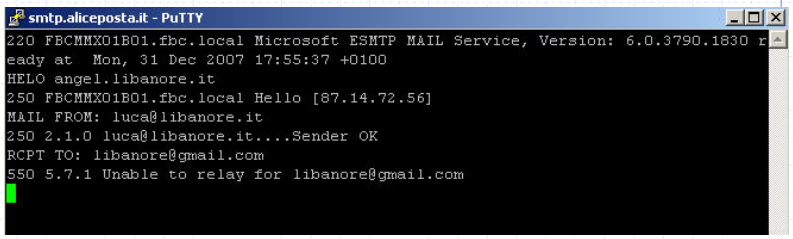
Le risposte dei server SMTP potrebbero essere le più svariate:

- Non posso perché il tuo indirizzo IP non è uno di quello autorizzati ad usare i miei servizi
- Non posso perché il ***Relaying*** (cos'è?) non è concesso!

Tutti questi problemi sono sorti da quando esiste il fenomeno dello *spamming*

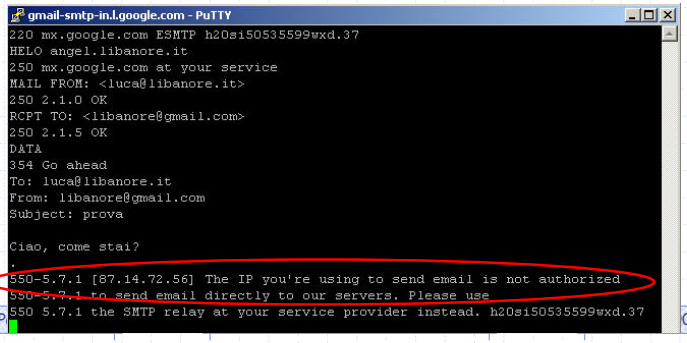
Al giorno d'oggi i server SMTP prima di accettare una e-mail da consegnare effettuano molti controlli (per evitare di finire in blacklist!)

Esempio di Relaying non permesso utilizzando il server SMTP smtp.aliceposta.it



Non posso offrirti il servizio perché il tuo indirizzo IP NON LO CONOSCO

E se proviamo a spedire un'email utilizzando un server SMTP di gmail? (ad esempio gmail-smtp-in.l.google.com)



I server SMTP di libero (MA NON SOLO) sono proprio maleducati!

Provate a fare un tentativo con un server SMTP di libero (ad esempio mxlibero1.libero.it): se non siete clienti di libero vi chiude direttamente la connessione in faccia!

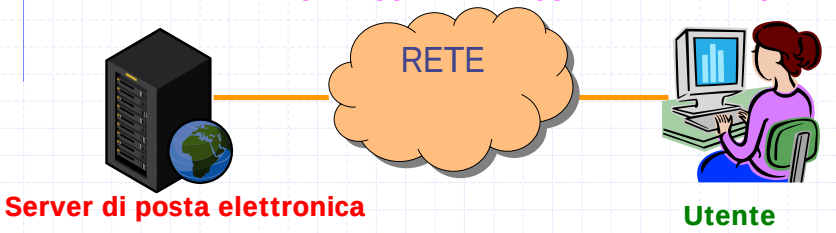
Come avviene il trasferimento dei messaggi dal server al client?

Esiste un'esigenza degli utenti di utilizzare il sistema di posta elettronica in questo modo:

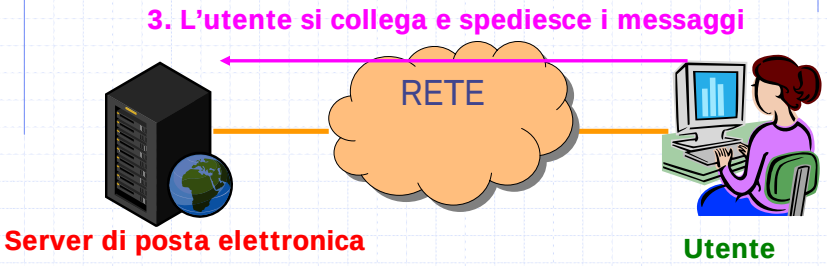
1. L'utente si collega e "scarica" i messaggi ricevuti

Dopo "aver scaricato" i messaggi

2. L'utente si scollega, legge i messaggi e scrive le risposte



Infine l'utente potrà ricollegarsi per spedire i nuovi messaggi



Esiste ancora l'esigenza di lavorare in locale?

- La necessità per gli utenti di scollegarsi e lavorare in locale, deriva dal ***1) non voler impegnare la linea telefonica*** (nel caso di collegamenti con linee tradizionali, quindi non ISDN, ADSL o Fibbra ottica) e ***2) di pagare di meno, visto che il costo del collegamento è a tempo***
- Queste esigenze sono ancora molto sentite dagli utenti standard, che non possiedono abbonamenti di tipo FLAT o non hanno effettuato il passaggio a linee a banda larga

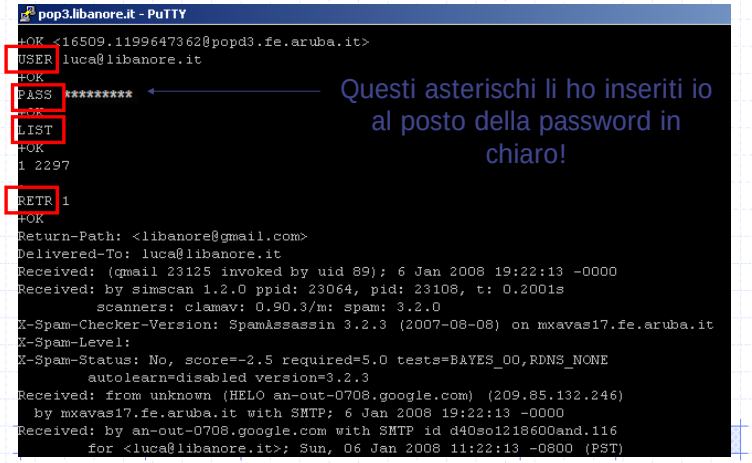
Quali protocolli vengono utilizzati per trasferire i messaggi dal server al client?

- Il primo, e più conosciuto, è POP3 (Post Office Protocol)
- ***POP3*** permette di effettuare queste ***operazioni*** con una serie di ***comandi*** sempre di tipo ***ASCII***, abbastanza simili a quelli del SMTP (USER, PASS, LIST, RETR, DELE, QUIT)
- POP3 rimane in attesa sulla porta TCP 110
- Sebbene sia il protocollo più diffuso per il trasferimento dei messaggi dal server di posta al client non è l'unico: esiste anche IMAP4

Ecco alcuni dei comandi ASCII che si scambiano i server POP3

USER	Nome dell'account FTP (oppure anonymous)
PASS	Password dell'account FTP (oppure indirizzo posta elettronica per account anonymous)
LIST	Chiede al server di elencare il contenuto della casella di posta elettronica
RETR	Serve per recuperare i messaggi
DELE	Contrassegna un messaggio per l'eliminazione
QUIT	Chiude la connessione

Un esempio del funzionamento del protocollo POP3 (utilizzando telnet)



Esercizi per casa

Controllate se il vostro provider (alice, libero, fastweb, etc.) oltre al server POP3 vi fornisce anche un server IMAP4

In caso affermativo, provate ad eseguire le seguenti prove, catturando il traffico con Wireshark, e annotate ciò che ritenete rilevante

Utilizzando POP3:

- provate ad accedere con due connessioni contemporanee alla stessa casella di posta elettronica

Utilizzando IMAP4:

- provate ad accedere contemporaneamente con IMAP4 alla stessa casella di posta elettronica
- In fase di autenticazione, come viene gestita la password?

Trovate che sia migliore POP3 o IMAP4? Sotto quali aspetti (performance in locale e/o remoto, sicurezza, gestione, ..)

Alcune annotazioni dagli esercizi

Caratteristiche di IMAP4 (ma non di POP3):

- *IMAP: mantiene i messaggi presso il server*
- **Accesso a molteplici caselle di posta sul server**: con il protocollo IMAP4 si possono creare, modificare o cancellare mailbox (di solito associate a cartelle) sul server.
- **Password criptate**: Con il protocollo POP le password vengono solitamente inviate in testo, rendendo facile, con una intercettazione, l'individuazione della password
- Porta TCP utilizzata: 143

Il comando AUTHENTICATE

DAL RFC 2060: "AUTHENTICATE *mechanism*: This command requests a special authentication mechanism with an argument from the server. If the server does not support that mechanism, the server sends an error message. Valid mechanisms, include:

- KERBEROS_V4
- GSSAPI
- SKEY

LOGIN vs AUTHENTICATE

```
C: a001 LOGIN SMITH SESAME
S: a001 OK LOGIN completed
```

contro

```
S: * OK KerberosV4 IMAP4revl Server
C: A001 AUTHENTICATE KERBEROS_V4
S: + AmFYig==
C: BAcAQU5EUkVXLkNNVS5FRFUAOCAsho84kLN3/IJmrMG+25a4DT
   +nZImJjnTNHJUtxAA+o0KPKfHEcAFs9a3CL5Oebe/ydHJUwYFd
   WwuQlMWiy6IesKvjL5rL9WjXUb9MwT9bpObYLGOKilQh
S: + or//EoAADZI=
C: DiAF5A4gA+oOIALuBkAAmw==
S: A001 OK Kerberos V4 authentication successful
```


Tutto il sistema di posta elettronica

E-mail da valentina@alice.it verso laura@libero.it

Valentina compone un messaggio e preme il tasto Invia del suo client di posta elettronica

"Qual è l'indirizzo IP di smtp.alice.it?"

Il client di posta di Valentina interroga il DNS per sapere l'indirizzo IP del server di posta di Alice (*da notare che Valentina conosce il nome del server SMTP, perché gli è stato comunicato ma non l'indirizzo IP*)

Server DNS di Alice.it

Tutto il sistema di posta elettronica: secondo atto

"Questi sono i destinatari e il messaggio"

Ora il client di posta di Valentina conosce l'indirizzo IP del server SMTP e può procedere

Sempre il client di posta di Valentina apre una connessione TCP ed effettua una conversazione SMTP (tramite i comandi visti prima) con il server SMTP in esecuzione sul server smtp.alice.it, per mezzo della quale gli consegna il messaggio;

Server SMTP di Alice.it

Tutto il sistema di posta elettronica: terzo atto

"Chi è il tuo Mail server e qual è il suo indirizzo IP?"

Il Server SMTP di Alice.it deve sa che il destinatario dell'email ha l'indirizzo laura@libero.it

Il server SMTP di Alice.it interroga un server DNS (utilizzando il meccanismo ricorsivo che abbiamo visto in precedenza) *per scoprire due informazioni: qual è il nome del mail server* (MX) *che accetta la posta per il dominio Libero.it e successivamente qual è l'indirizzo IP (A) di tale server*

Server DNS di Libero.it

Tutto il sistema di posta elettronica: quarto atto

"Questi sono i destinatari e il messaggio"

Il Server SMTP di Alice.it ora conosce il mail server di Libero e può inoltrargli l'email

Il server SMTP di Alice.it apre una connessione TCP e poi una conversazione SMTP (utilizzando sempre gli stessi comandi visti prima) con il mail server mxlibero1.libero.it e gli consegna il messaggio scritto da Valentina

Mail Server di Libero.it

Tutto il sistema di posta elettronica: quinto atto – le complicazioni!

A questo punto il messaggio di posta elettronica è giunto al mail server di libero, mxlibero1.libero.it

Cosa se ne fa il mail server di questo messaggio?

DIPENDE

1. Il Mail server è esso stesso il server POP3/IMAP4; mantiene l'e-mail nei suoi hard disk fin quando un client (POP3 o IMAP4) si occuperà di venirlo a prelevare, a seguito di una richiesta da parte di un utente
2. Inoltra il messaggio al server POP3/IMAP4 di competenza;

A volte mail server e server POP3/IMAP4 sono lo stesso host, la maggior parte delle volte sono host diversi!

Tutto il sistema di posta elettronica: sesto atto (facoltativo, dipende dalle scelte sistemistiche)

Il Mail Server di Libero.it

Questo è il messaggio che ho ricevuto per laura@libero.it

Il mail server di Libero.it trasferisce il messaggio di posta elettronica scritto da Valentina presso il server POP3/IMAP4 popmail.libero.it

Il server pop3/imap4 di libero

Tutto il sistema di posta elettronica: settimo atto

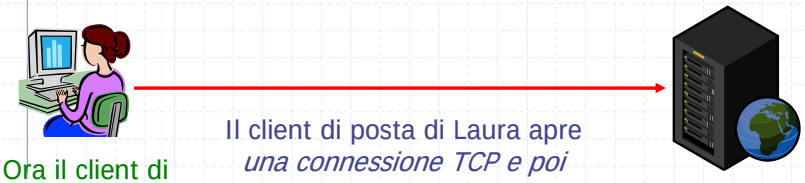
Laura, utilizzando il suo client di posta elettronica, tramite il pulsante Ricevi controlla se ha posta

"Qual è l'indirizzo IP di popmail.libero.it?"

Il client di posta di Valentina interroga il DNS per sapere l'indirizzo IP del server POP3/IMAP4 di Libero (*da notare che Valentina conosce il nome del server POP3/IMAP4, perché gli è stato comunicato ma non l'indirizzo IP*)

Server DNS di Libero.it

Tutto il sistema di posta elettronica: ultimo atto (il più difficile da prevedere!)

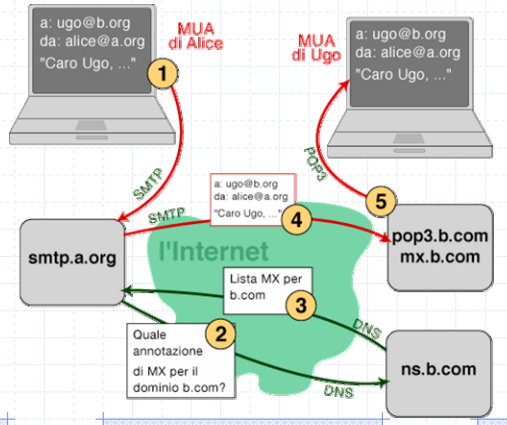


Ora il client di posta di Laura conosce l'indirizzo IP del server POP3/IMAP4 e può procedere

Il client di posta di Laura apre *una connessione TCP e poi una conversazione POP3 (o IMAP4) col server POP3/IMAP4 in esecuzione su popmail.libero.it e preleva (o consulta) il messaggio di Valentina, che viene mostrato a Laura.*

Server pop3/imap4 di Libero.it

Riassunto grafico (tratto da wikipedia.it)



Osservazioni sulle scelte sistemistiche

- A volte mail server e server POP3/IMAP4 coincidono (es. aruba.it, anche se non offre l'accesso IMAP), a volte no (fastweb.it), come mai?
- A volte server SMTP e POP3/IMAP4 coincidono (poste.it), a volte no (alice.it), come mai?
- È consigliabile far coincidere Mail server, Server SMTP e server POP3/IMAP4 tutti su uno stesso server (es. Supereva.it, anche se non offre IMAP)? Pensate ai Vantaggi/Svantaggi di questa soluzione.
- Provate, come compito per casa, a pensare/trovare informazioni su Internet sulle differenti scelte sistemistiche (pensate anche ad altri casi, come ad esempio un sistema di posta dove il server pop3 è situato su un host differente rispetto al server imap4. Perché? In quali casi serve?)

Un *"altro sistema"* per realizzare la posta elettronica: il webmail

Una Webmail è un'***applicazione web*** che permette di ***gestire*** un account di posta elettronica ***attraverso un browser***

Attraverso l'interfaccia grafica si ***stabilisce*** una normale ***connessione verso un server di posta***

Vantaggi del Webmail

- Possibilità di *leggere la propria posta ovunque* vi sia una connessione ad internet
- I *messaggi non necessitano di essere scaricati*
- Le ***caselle di posta*** possono essere ***amministrate*** (create, modificate, cancellate) molto ***facilmente***

Svantaggi del Webmail:

- È ***richiesta una connessione sia per la visualizzazione che per la composizione*** dei messaggi
- Una ***connessione lenta influenza*** la ***funzionalit***à generale della webmail
- Le ***funzionalità di composizione*** di ***messaggi*** sono generalmente ***limitate per la formattazione*** di un messaggio

Alcune considerazioni a livello di scelte sistemistiche

Come avrete potuto notare, non esiste la scelta valida per tutti i casi.

Le scelte sistemistiche dipendono da un certo numero di fattori:

- Risorse a disposizione (hw/sw/$$$)
- Qualità del servizio che si vuol offrire (servizio gratuito vs. servizio a pagamento)
- Aspetti legati alla sicurezza
- Numero di utenti
- Politiche di Load Balancing e Fault Tollerance
- .....

Quindi, dovrebbe risultarvi abbastanza chiaro che la stessa macchina potrebbe ospitare contemporaneamente un server SMTP, POP3 e ad esempio anche HTTP, FTP, etc.

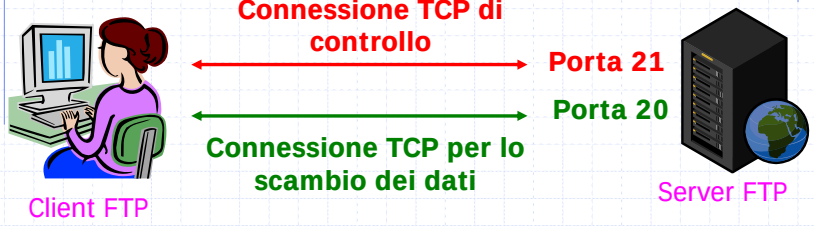
Nelle scelte sistemistiche bisogna farsi guidare dal buon senso e dai mezzi a disposizione.

Il File Transfer Protocol (FTP)

Come funziona il protocollo FTP?

- FTP è un protocollo client/server
- Permette il *trasferimento* (o meglio la copia) *di file* da una macchina remota sulla macchina locale o viceversa
- È un protocollo di livello applicativo
- È basato su dei semplici comandi ASCII

Un esempio di funzionamento di FTP



La particolarità del protocollo FTP sta proprio nell'utilizzare due diffenti connesioni TCP per comandi (21) e dati (generalmente la porta 20)

A cosa serve la connessione di controllo?

- Per scambiare comandi fra client e server
- Dura per l'intera sessione di collegamento dell'utente (a differenza della connesione dati)
- Si instaura sulla porta TCP 21

A cosa serve la connessione per lo scambio dei dati?

- Si instaura generalmente sulla porta TCP 20, man mano che serve inviare e ricevere il file
- Quindi per ogni file che viene trasferito viene aperta questa connessione dati su cui i bit relativi al file da trasferire vengono spediti

Alcune complicazioni per i server FTP

L'FTP è più complesso di altri protocolli applicativi, perché richiede che il server ***mantenga lo STATO dei comandi dell'utente*** quando questo lavora all'interno di una sessione

Quindi, per tutta la durata del collegamento tra il client e il server, il server stesso dovrà mantenere le informazioni come:

- Il **client è stato autenticato** (quindi ha ottenuto un accesso tramite un account)
- Il **client sta lavorando in una certa directory**
- In questo momento il client sta trasferendo un file
- .....

Come si comporta l'FTP dal punto di vista dei pacchetti scambiati?

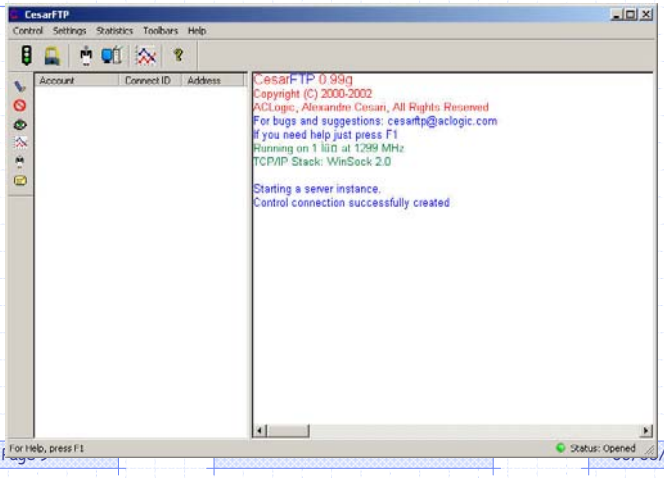
Per controllare i pacchetti scambiati durante una sessione FTP utilizzeremo Wireshark, lo sniffer visto durante le altre lezioni

Un server FTP

Per effettuare le prove che seguiranno, ho attivato un server FTP su una workstation, mettendolo in ascolto sulla porta 21, in attesa di richiesta di connessione da parte di client FTP

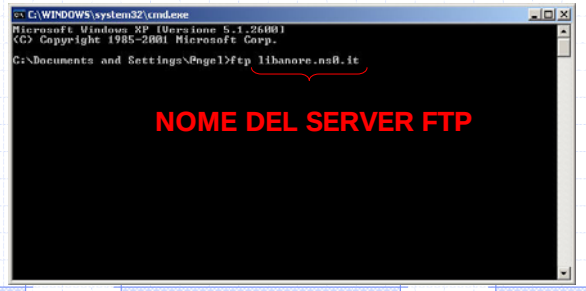
In particolare il server FTP che ho utilizzato si chiama CesarFTP, è freeware, con un'interfaccia grafica e molto semplice da utilizzare

CesarFTP: il server FTP



Il client FTP

- Per quanto riguarda il client FTP, ho utilizzato il client FTP di Windows (si lancia da riga di comando)



Vediamo un esempio di sessione

- Come prima cosa lanciamo Wireshark
- Dopo di che creiamo la sessione FTP, facendo collegare il client FTP al server e autenticandoci come utente anomimo (utente: anonymous, password: vuota)

Problemi: a scuola non ci si può collegare al server FTP

- Ora faremo qualche simulazione con un server FTP installato sul mio computer
- Successivamente analizzeremo una cattura di wireshark che io ho effettuato a casa per voi

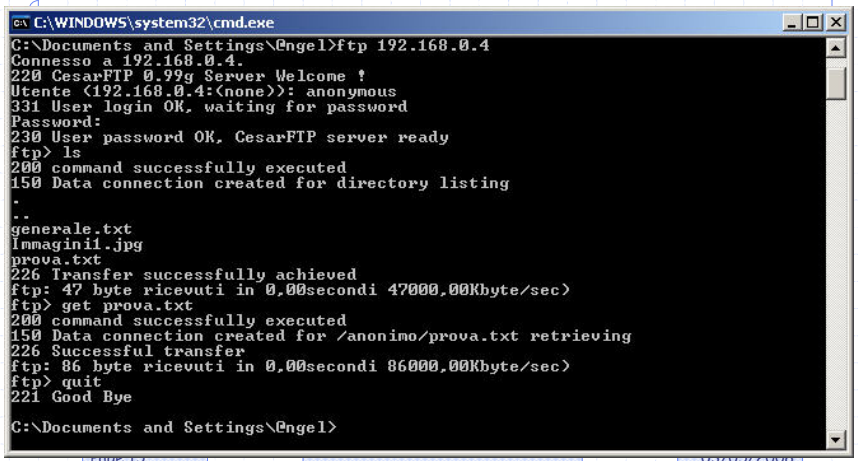
Analisi del protocollo

- Come prima cosa mi collego al server ftp, utilizzando il client ftp di Windows: ftp 192.168.0.4
- Mi autentico come utente anonymous e invio la password, che corrisponde all'indirizzo e-mail (anche se non vi è alcun tipo di controllo che questa e-mail sia valida)
- Ora posso operare con i normali comandi del client ftp
- Osservazione: i comandi che ora digiterò non sono i comandi del protocollo FTP, bensì quelli implementati dal client (ad esempio GET che serve per ricevere un file corrisponde al comando RETR FTP)

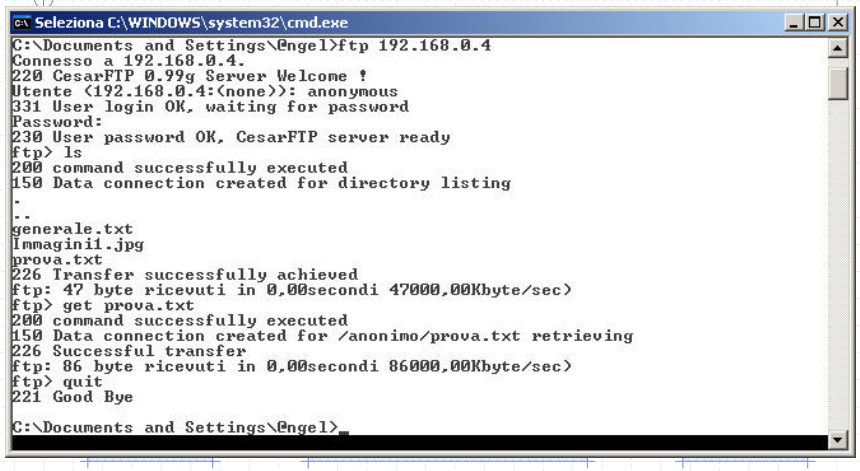
Cosa facciamo fare al nostro client?

- Il primo comando che sottoponiamo al server FTP è la lista dei file remoti (digito ***LS*** nel client che corrisponde ai comandi ***PORT + NLST***)
- Proviamo a trasferire il file prova.txt attraverso il comando GET prova.txt (digito GET nel client che corrisponde ai comandi ***PORT + RETR***)
- Chiudiamo la connessione di lavoro con il comando QUIT e vediamo cos'è stato catturato dallo sniffer, analizzando il traffico generato da questa sessione

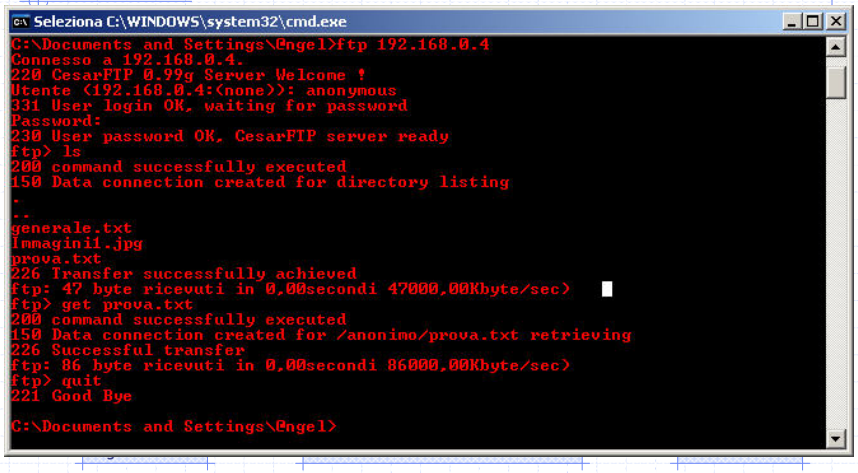
La sessione FTP



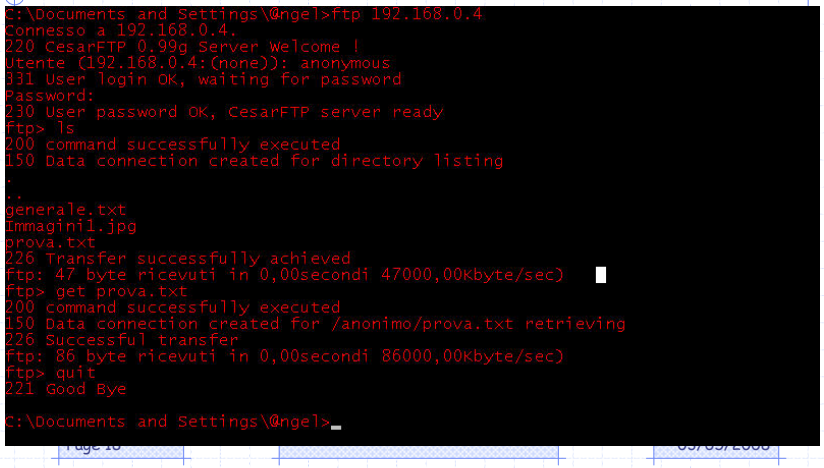
La sessione FTP (con un altro sfondo!)



La sessione FTP (penultimo tentativo!)



La sessione FTP (ultimo tentativo!)



Cattura completa

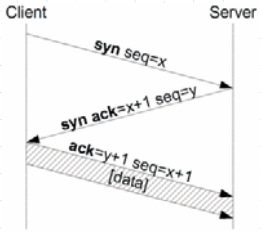
- Il nostro analizzatore di protocollo ha effettuato la cattura completa della sessione FTP e quindi potremo vedere come la connessione di controllo e le varie connessioni dati sono state utilizzate da parte del client e del server per comunicarsi le informazioni richieste

ARP request e ARP response

- La prima attività svolta (pacchetto 1) è quella di un ARP request, ovvero avendo a disposizione un indirizzo IP qual è il corrispondente MAC address dell'Host?
- Il server risponde (pacchetto 2) alla richiesta comunicando l'indirizzo MAC del destinatario

Richiesta di connessione TCP

Dopo aver ottenuto l'indirizzo MAC si stabilisce una connesione TCP (pacchetti 3-5), utilizzando la tecnica del ***Three-way handshake***



Siamo pronti per lo scambio applicativo

Dopo questi pacchetti, possiamo vedere il primo scambio di informazioni visibili da console (pacchetto 6), corrispondente alle stringhe ASCII del comando 220 di FTP che indica l'invio di una stringa ASCII da visualizzare sul client FTP

In questo caso abbiamo la visualizzazione dell'informazione relativa al nome e alla versione del programma server FTP

Vediamo ora nel dettaglio la sessione applicativa

- La successiva operazione, effettuata da parte del client FTP, è stata quella di inviare il comando USER anonymous
- Quindi il comando FTP USER (***che il client FTP nasconde***) è seguito dalla stringa anonymous che identifica l'utente anonimo
- Il server ha risposto con un'informazione di accettazione del particolare username (che non sempre è disponibile sui server FTP), insieme alla richiesta dell'invio di una password

Il comando FTP PORT

- Questo comando serve per aprire una porta data
- Il client comunica al server verso quale porta dovrà spedire i dati richiesti
- Il comando è del tipo
- PORT 192,168,0,3,19,138 che vuol dire spedisci i dati verso l'IP 192.168.0.3 porta 5002 (= 19*256+138)

La modalità passiva

- Se notate il server non utilizza la porta 20 per spedirci i dati!
- Infatti FTP prevede due modalità di funzionamento
  - Modalità attiva: il server utilizza la porta 20 per spedire i dati
  - Modalità passiva: il server non utilizza la porta 20 per spedire i dati bensì viene scelta ***dinamicamente*** una porta privata (>1024) e questa cambia ad ogni invio di dati da parte del server al client

Modalità attiva vs. modalità passiva

- Modalità attiva: è il metodo originale usato dal protocollo FTP per il trasferimento dei dati all'applicazione del client
  - Con la crescita delle reti non sicure, come ad esempio Internet, l'uso dei firewall per proteggere le macchine dei client è molto importante
  - Poichè questi firewall spesso impediscono i collegamenti in entrata provenienti dai server FTP in modalità attiva, è stato ideata la modalità passiva
- Modalità passiva: anche se la modalità passiva risolve le problematiche dovute all'interferenza dei firewall con i dati di collegamento, tale modalità può complicare la gestione dei firewall del server

La creazione/chiusura delle connessioni dati

- Notate come, in questa sessione FTP, le connessione dati effettivamente create sono state due
- La prima parte dal pacchetto 18 fino al 25 e corrisponde al comando LS, ovvero la lista dei file contenuti nella directory remota
- La seconda connessione corrisponde al campo GET e parte dal pacchetto 33 fino al 40

I veri comandi scambiati tra un client FTP e un server FTP

- ABOR - abort a file transfer
- CWD - change working directory
- DELE - delete a remote file
- LIST - list remote files
- MDTM - return the modification time of a file
- MKD - make a remote directory
- NLST - name list of remote directory
- PASS - send password
- PASV - enter passive mode
- PORT - open a data port
- PWD - print working directory

I veri comandi scambiati tra un client FTP e un server FTP (continua)

- QUIT - terminate the connection
- RETR - retrieve a remote file
- RMD - remove a remote directory
- RNFR - rename from
- RNTO - rename to
- SITE - site-specific commands
- SIZE - return the size of a file
- STOR - store a file on the remote host
- TYPE - set transfer type
- USER - send username

HyperText Transfer Protocol (HTTP)

Cos'è l'HTTP?

- HTTP è l'acronimo di HyperText Transfer Protocol (protocollo di trasferimento di un ipertesto)
- HTTP è usato come principale sistema per la trasmissione di informazioni sul web
- L'ipertesto è una struttura informativa costituita di un insieme di testi o pagine leggibili con l'ausilio di un'interfaccia elettronica, in maniera non sequenziale, per mezzo di particolari parole chiamate collegamenti ipertestuali

Come funziona l'HTTP?

- L'HTTP funziona su un meccanismo richiesta/risposta (client/server):
  - il client esegue una richiesta ed il server restituisce la risposta
- Nell'uso comune il client corrisponde al browser ed il server al sito web
- Vi sono quindi due tipi di messaggi HTTP: messaggi richiesta e messaggi risposta

Un protocollo stateless

- HTTP differisce da altri protocolli di livello 7 come FTP, per il fatto che le connessioni vengono generalmente chiuse una volta che una particolare richiesta (o una serie di richieste correlate) è stata soddisfatta
- HTTP utilizza la porta TCP 80

Trasferimento di file: FTP vs. HTTP

- Se ci pensiamo bene sia FTP che HTTP permettono il trasferimento di file
- Pensiamo ad un immagine: questa può essere trasferita sul nostro computer sia mediante un server FTP che mediante un server HTTP
- A questo punto è importante chiedersi: è meglio offrire un servizio di trasferimento file basandosi su server FTP oppure su server HTTP?

La risposta è... DIPENDE!

- Se dobbiamo rendere disponibile a tutti il download di un'immagine allora la soluzione migliore è HTTP
- Infatti, FTP ha il grosso svantaggio di utilizzare ben 2 connessioni TCP, una per i comandi e una per il trasferimento dei file!
- 2 Connessioni TCP aperte significa almeno 2 problemi:
  - Doppio uso di risorse (2 porte)
  - Due porte aperte in più all'esterno, aumentando i rischi di attacco

Ma allora di FTP cosa ce ne facciamo?

- Nel primo esempio abbiamo parlato di permettere a tutti il dowload di un'immagine
- E se volessimo permettere il download solo ad alcuni utenti (per esempio solo i nostri clienti)?
- Se pensiamo ai soli protocolli visti fino ad ora allora l'unica possibilità è l'uso di FTP, perché permette un controllo sugli accessi mediante il concetto di account

E l'upload?

- Un altro aspetto per cui è importante considerare l'uso di un server FTP è il concetto di upload, ovvero noi che trasferiamo qualcosa su un server
- L'HTTP è detto *one-way system* poiché è possibile trasferire unicamente file dal server verso il client
- Invece FTP è detto *two-way system* poiché è possibile sia l'azione di trasferire file dal server verso il client (download) sia il viceversa (upload)

Ma HTTP e FTP ci bastano?

- La risposta è semplice...NO!
- Pensiamo, ad esempio, se volessimo far vedere determinate pagine del nostro sito Internet solo ad alcuni utenti e non a tutti!
- O pensiamo ai dati che viaggiano in chiaro su FTP
- Per risolvere questi problemi esistono delle evoluzioni di HTTP e FTP

Il protocollo HTTPS

- Sintatticamente identico allo schema http:// ma con la differenza che gli accessi vengono effettuati sulla porta 443 e tra il protocollo TCP e HTTP si interpone un livello di crittografia/autenticazione
- In pratica viene creato un canale di comunicazione criptato tra il client e il server attraverso lo ***scambio di certificati***; una volta stabilito questo canale al suo interno viene utilizzato il protocollo HTTP per la comunicazione

HTTPS, ovvero come proteggere il contenuto dai curiosi

- Questo tipo di comunicazione garantisce che solamente il client e il server siano in grado di conoscere il contenuto della comunicazione
- Questo sistema è largamente usato nel World Wide Web per situazioni che richiedono particolari esigenze in ambito di sicurezza come per esempio il pagamento di ***transazioni online***

HTTPS per il controllo degli accessi

- Questa tecnologia può essere usata anche per permettere un accesso limitato ad un web server
- ***L'amministratore spesso crea dei certificati per ogni utente che vengono caricati nei loro browser contenenti informazioni come il relativo nome e indirizzo e-mail*** in modo tale da permettere al server di riconoscere l'utente nel momento in cui quest'ultimo tenti di riconnettersi senza immettere nome utente e/o password

SFTP

- SFTP è un protocollo di rete che prevede il trasferimento dei dati e funzionalità di manipolazione
- Grazie a SFTP il trasferimento dei file avviene in modo sicuro in quanto vengono ***trasmessi criptati e comprensibili solo a client e server*** che all'inizio della trasmissione si scambiamo chiavi di crittografia

Conclusioni

Durante queste lezioni abbiamo visto che....

- Esistono tanti servizi (noi ne abbiamo visto giusto qualcuno!) e ognuno serve a soddisfare un'esigenza diversa
  - Traduzione automatica nome dns/ indirizzo IP
  - Posta elettronica
  - Trasferimento File
  - Trasferimento di un ipertesto

Cosa dovremmo avere imparato?

- Anche uno stesso servizio può essere offerto in maniera diversa (un po' come andare a mangiare alla trattoria da Gigi e al ristorante Il Cambio)
- Ad esempio abbiamo visto che esistono due modi per modi per offrire il servizio di ricevimento dei messaggi di posta elettronica (POP3 e IMAP4); entrambi funzionano ma uno si adatta di più a risolvere certe situazioni, l'altro invece viene incontro ad altre esigenze

Cosa utilizzare?

- Utilizziamo POP3 o IMAP4 per ricevere la posta elettronica?
- Utilizziamo HTTP o FTP per offrire il download di file?
- Ma il DNS ci serve davvero?
- ***DIPENDE***!

Dipende da...

- Le risorse (economiche e hardware/software) che abbiamo a disposizione
- Che servizio dobbiamo offrire ai clienti
- Privilegiare aspetti di sicurezza o di performance

Se abbiamo...

- Una rete composta da due computer e vogliamo fare qualche simulazione è sufficiente caricare tutti i server su un'unica workstation
- Se dobbiamo offrire un servizio di posta elettronica efficiente (e gli altri servizi sono meno importanti) verrà privilegiato questo servizio mettendo a disposizione i computer più avanzati e magari installando il server SMTP su una macchina, il mail server su di un'altra e il server POP3/IMAP4 su un'altra macchina ancora

Inoltre...

- Se vogliamo offrire il servizio di webmail dobbiamo ricordarci che server web (ad esempio Apache) e Mail server con funzionalità di webmail devono essere messi in ascolto su porte diverse (perché entrambi si metterebbe in ascolto sulla porta 80) o su macchina diverse

Clienti Basic vs. Clienti Premium

- Se un utente lo facciamo pagare magari ci conviene offrigli un servizio migliore rispetto a quello offerto ad un cliente che non paga (altrimenti perché dovrebbe pagare)
- Quindi magari dedicandogli dei server e garantendogli un certo livello di sicurezza

Come fare allora?

- Il trucco consiste nel fare una lista delle risorse a disposizione, fare una lista dei servizi che si vuole offrire, fare una lista della tipologia di clienti
- Si mettono insieme tutte le informazioni e si inizia a progettare una soluzione ragionevole, che tenga conta di ipotesi realistiche

Buona fortuna!