Servizi di rete passo a passo

Appunti sulla installazione e configurazione dei servizi di rete in ambiete Gnu/Linx

Author: Andrea Manni

Version: 0.92

Questa guida tratta la configurazione di base dei principali servizi di rete utilizzati con sistemi operativi *Unix* e derivati come *GNU/Linux* per gli studenti dei corsi per amministratori di rete in ambiente *GNU/Linux*. In particolare questa guida e' stata scritta usando come riferimento la distribuzione *Debian GNU/Linux*. Gli esempi presentati dovrebbero quindi essere direttamente utilizzabili anche su distribuzioni derivate da Debian come Ubuntu, per sistemi operativi diversi si presti attenzione ai percorsi dei file citati.

Indice degli argomenti

Generato il 2010-02-04 con: http://docutils.sourceforge.net/rst.html

1 Configurazione sistema

1.1 Solo per uso interno

Impostazioni di base per la configurazione del sistema operativo e della rete nel laboratorio 208 facente parte della rete piffa.net .

Sono qui riportati i parametri della rete locale per comodita' degli studenti, gli altri lettori possono considerarli come riferimento per comprendere i valori espressi nei vari file di configurazione. Ad esempio: quando leggerete `10.10.208.248:3128` saprete che si tratta del nostro *proxy http* in ascolto sulla porta *3128*, stara' quindi a voi sostituire i dati con gli equivalenti *IP* della vostra rete.

1.2 Rete

Parametri della rete attualmente in uso:

Parametri della rete	
rete	10.10.208.0/24
netmask	255.255.255.0
broadcast	10.10.208.255
gateway	10.10.208.248
DNS	10.10.208.248

Dal server locale degli studeti, **Bender**, corrispondente all'IP `10.10.208.248`, vengono erogati i servizi DHCP, DNS, gateway (con NAT), proxy http e mirror della distribuzione Debian (http://debian.piffa.net). Altri servizi in esecuzione sul server:

- Rsync server e altri software di aggiornamento / installazione di massa
- Server imap / pop3 / webmail / smtp
- Server ssh per i test degli studenti
- File server Samba, NFS e controller di qualsiasi altro FS distribuito
- Print server per la gestione delle stampanti

- DNS server
- Mirror locale delle *.iso dei sistemi operativi e dei vari software usati durante le lezioni
- Spazi web con PHP, MySQL, Postgresql (altri DB o framework vengono attivati a seconda dei corsi attivi)

Durante il corso delle lezioni e' opportuno che le macchine degli studenti si appoggino al server Bender (ottetto finale `248`), nel caso questo non fosse raggiungibile (ad esempio per permettere impostazioni di DHCP / routing diverse) sara' comunque disponibile il `10.10.208.254` come gateway | DNS per la rete `10.10.208.10`.

Non e' piu' possibile raggiungere Bender tramite l'IP pubblico `212.22.136.248` o *qualsiasi altro ip* della classe `C 212.22.136.0/24` precedentemente disponibile.

Il computer del docente con il server VNC e' sempre configurato con l'ottetto finale: `177` della rete utilizzata durante le lezioni (quindi generalmente la VNC sara' disponibile sul `10.10.208.177:1`.

Gli studenti sono pregati di non impedire l'accesso SSH alla propria macchina dal computer del docente, e non modificare la password dell'utente `root` del sistema operativo *pre-istallato* (ad es: *Diurno*).

1.2.1 interfaces

Segue un esempio del file di configurazione della scheda di rete con configurazione statica:

/etc/network/interfaces:

```
# /etc/network/interfaces -- configuration file for ifup(8), ifdown(8)

# The loopback interface
iface lo inet loopback

# La prima scheda di rete (se si chiama eth0)
iface etho inet static
  # esempio con dhcp:
  # iface etho inet dhcp
address 10.10.208.101
netmask 255.255.255.0
network 10.10.208.0
broadcast 10.10.208.255
gateway 10.10.208.254

# Quali interfaccie devono partire automaticamente:
auto lo eth0
```

Controllare il nome della propria scheda di rete: a volte *udev* rinomina la prima scheda a `eth1`, oppure potreste avere piu' di una scheda di rete (anche un'interfaccia *firewire* puo' essere automaticamente abilitata come scheda di rete).

Se si usano *schede di rete virtuali* (eth0:1 , eth0:1 , ...) ricordarsi che queste dipendono dalla scheda fisica a cui sono associate: abbattere con `ifconfig down eth0` la scheda principale fara' cadere anche queste. Tornando ad attivare la scada principale con `ifconfig eth0 up` la virtuale tornera' attiva: nel caso voleste disabilitarla dovrete quindi sempre abbattere manualmente la scheda virtuale *prima* della scheda reale.

I DNS vanno indicati nel file `/etc/resolv.conf` , la cui sintassi e' spiegata al punto 4.6 . Come DNS si *deve* usare il server Bender, alcuni parametri dei software di installazione, risoluzione dei mirror, vengono opportunamente modificatia da questo DNS.

1.3 Bash completion

Il completamento automatico della shell (che si attiva premendo il tasto tab una o due volte mentre si sta scrivendo un termine) permette di comporre automaticamente i nomi dei comandi e i percorsi dei file, sopratutto la composizione automatica dei percorsi dei file e' di grande importanza.

Bash_completion permette di integrare il completamento automatico con i nomi dei pacchetti e oggetti dei comandi: ad es. volendo digitare `apt-get inst[TAB] xtigh[TAB]` ora verra' completato automaticamente sia la parola `install` che il nome del pacchetto `xtightvncviewer`.

Abilitare /etc/bash_completion nel file `/etc/bash.bashrc` oppure includerlo nel proprio `~/.bashrc` (che sarebbe il file *nascosto*, quindi con un punto all'inizio del nome del file, di configurazione della shell bash per ogni utente, presente nella propria *home directory*):

```
echo ". /etc/bash_completion" >> ~/.bashrc
```

Esempio di ~/.bahsrc

```
# ~/.bashrc: executed by bash(1) for non-login shells.

export PS1='\h:\w\$ '
umask 022

# De-commentare le seguenti righe per abilitare la colorazione dei
# nomi dei file:
 export LS_OPTIONS='--color=auto'
 eval "`dircolors`"
 alias ls='ls $LS_OPTIONS'
 alias ll='ls $LS_OPTIONS -l'
 alias l='ls $LS_OPTIONS -lA'

# Abilitare i seguenti alias per impostare la conferma per cancellare file
# alias rm='rm -i'
# alias cp='cp -i'
# alias mv='mv -i'

# questo abilita bash completion
. /etc/bash_completion
```

Il file `/etc/bash_completion` deve essere presente nel sistema, in caso contrario installare il pacchetto: `bash-completion`. Generalmente l'utente `root` ha un file `.bashrc` preimpostato analogo a quello citato sopra, a differenza dei normali utenti di sistema.

Links:

- An introduction to bash completion
- Working more productively with bash 2.x/3.x
- UNIX / Linux Shell Scripting Tutorial: http://steve-parker.org/sh/sh.shtml

1.4 Vim

Vim e' l'editor di testo preferito dai sistemisti, quindi sara' conveniente impostare fin da subito alcune impostazioni per renderlo piu' comodo.

Assicurarsi che sia installata nel sistema la versione completa dell'editor installando il pacchetto `vim`:

```
# apt-get install vim
```

Modificare poi il file di configurazione generale `/etc/vim/vimrc`

```
" All system-wide defaults are set in $VIMRUNTIME/debian.vim (usually just
" /usr/share/vim/vimcurrent/debian.vim) and sourced by the call to :runtime
" you can find below.  If you wish to change any of those settings, you should
" do it in this file (/etc/vim/vimrc), since debian.vim will be overwritten
" everytime an upgrade of the vim packages is performed.  It is recommended to
" make changes after sourcing debian.vim since it alters the value of the
" 'compatible' option.

" This line should not be removed as it ensures that various options are
" properly set to work with the Vim-related packages available in Debian.
runtime! debian.vim

" Uncomment the next line to make Vim more Vi-compatible
" NOTE: debian.vim sets 'nocompatible'.  Setting 'compatible' changes numerous
" options, so any other options should be set AFTER setting 'compatible'.
"set compatible

" Vim5 and later versions support syntax highlighting. Uncommenting the next
" line enables syntax highlighting by default.
syntax on

" If using a dark background within the editing area and syntax highlighting
" turn on this option as well.
set background=dark

" Uncomment the following to have Vim jump to the last position when
" reopening a file

if has("autocmd")
  au BufReadPost * if line("'\"") > 0 && line("'\"") <= line("$")
    \| exe "normal! g'\"" | endif
endif

" Uncomment the following to have Vim load indentation rules and plugins
" according to the detected filetype.
" This is not recommanded if you often copy and paste into vim,
" as it messes all the indentation.
if has("autocmd")
  filetype plugin indent on
endif

" This goes for comments folding: use co to expnad and zc to compress,
" zi to toggle on/off
set fdm=expr
set fde=getline(v:lnum)=~'^\\s*#'?1:getline(prevnonblank(v:lnum))=~'^\\s*#'?1:getline(nextnonblank(v:lnum))=~'^\\s*#'?1:0

" The following are commented out as they cause vim to behave a lot
" differently from regular Vi. They are highly recommended though.
set showcmd             " Show (partial) command in status line.
"set showmatch          " Show matching brackets.
# Ignorecase is quite userfull
set ignorecase          " Do case insensitive matching
"set smartcase          " Do smart case matching
"set incsearch          " Incremental search
set autowrite           " Automatically save before commands like :next and :make
"set hidden             " Hide buffers when they are abandoned
"set mouse=a            " Enable mouse usage (all modes) in terminals

" Source a global configuration file if available
" XXX Deprecated, please move your changes here in /etc/vim/vimrc
if filereadable("/etc/vim/vimrc.local")
  source /etc/vim/vimrc.local
endif
```

I principianti faranno bene ad esercitarsi con `vimtutor it`.

Altri link per VIM:

- Vim Introduction and Tutorial: http://blog.interlinked.org/tutorials/vim_tutorial.html
- http://blog.smr.co.in/category/vim/
- http://vimdoc.sourceforge.net/

1.5 VNC

I Virtual Network Computing (o VNC) sono software di controllo remoto e servono per amministrare un computer a distanzai. Nel nostro caso la VNC sara' utilizzata per visualizzare la sessione di lavoro di un altro computer sul proprio a scopo didattico.

Per collegarvi al server del docente usate lo script `guarda.sh` che dovrebbe gia essere disponibili sui sistemi preconfigurati, oppure potete invocare direttamente il collegamento con:

```
xtightvncviewer -viewonly 10.10.208.177:1
```

Se il comando non fosse disponibile installate il pacchetto `xtightvncviewer`. Potete anche scaricare lo script `guarda.sh` e renderlo eseguibile, ed eventualmente creare una voce nel menu di KDE per richiamarlo.

Procedura:

```
su root
cd /usr/local/bin
wget http://bender/guarda.sh
chmod +x guarda.sh
exit
```

Per eseguire lo script digitare semplicemente `guarda.sh`, oppure creare un link / collegamento sul Desktop allo script `/usr/local/bin/guarda.sh`.

Le impostazioni del server VNC sono:

Parametro	Valore
IP	10.10.208.177:1
Server grafico	:1
password	password

Si noti che non e' possibile lanciare un applicativo sul server grafico di un utente da una shell in cui si sta lavorando come altro utente, anche se root. E' quindi necessario essere l'utente di sistema che si e' loggato inizialmente nella sessione grafica per poter lanciare lo script guarda.sh da una shell.

Controllare con `whoami` di essere l'utente normale (es `utente | studente | proprio nome`), in caso si sia assunta una altra `id` si apra un altra shell o si esca da quella attuale con `exit` .

1.6 Lista dei pacchetti di base

I pacchetti installati generalmente [1] per poter seguire le lezioni sono:

```
kde-core kdm kde-i18n-it xorg vim less xtightvncviewer
```

1.7 Apt configurazione

Vediamo i due file principali di apt:

- `/etc/apt/sources.list`
- `/etc/apt/apt.conf`

1.7.1 sources.list

Questo file contiene i sorgenti da cui *apt* preleva i pacchetti da installare tramite *dpkg*, vengono quindi precisati i metodi (ad es. http / ftp / cdrom / file), la release che si vuole tracciare (es `stable, testing, unstable` oppure i corrispondenti release name es: `Lenny, Squeeze, Sid`), i rami di interesse (es: `main` che e' l'archivio principale, `non-free` per il software non libero, `contrib` per i pacchetti non realizzati dai manutentori ufficiali).

Gli archivi sono generalmente:

- `deb` per pacchetti Debian binari, pronti per l'installazione.
- `deb-src` per i pacchetti sorgenti (quindi da compilare, come il kernel) degli stessi pacchetti binari. In genere se non compilate spesso potete evitare di tracciare i sorgenti per risparmiare tempo e banda.

`/etc/apt/sources.list`

```
# esempio di accesso a un CDROM:
# cdrom:[Debian GNU/Linux 5.0.1 _Lenny_ - Official i386 kde-CD Binary-1 20090$

# ftp.it.debian.org viene rediretto su un mirror interno
# quando vi trovate nella rete interna piffa.net
deb http://ftp.it.debian.org/debian/ lenny main
# Sono disponibili anche i rami non-free contrib
# deb http://ftp.it.debian.org/debian/ lenny  non-free contrib
# Sono disponibili anche le release unstable e testing
# deb http://ftp.it.debian.org/debian/ testing  main non-free contrib
# deb http://ftp.it.debian.org/debian/ sid  main non-free contrib

# Sorgenti dei pacchetti:
# deb-src http://ftp.bononia.it/debian/ lenny main

# Mirror da kernel.org europa da usare a casa:
deb http://mirrors.eu.kernel.org/debian/ lenny main

# Security dal sito principale
deb http://security.debian.org/ lenny/updates main
# deb-src http://security.debian.org/ lenny/updates main

# Debian volatile per i pacchetti soggetti a frequanti cambiamenti
# non legati a dinamiche di sicurezza
deb http://volatile.debian.org/debian-volatile Lenny/volatile main
# deb-src http://volatile.debian.org/debian-volatile Lenny/volatile main

# Esempio di accesso a un file system locale contenente i pacchetti:
# Potete scaricarei in aula con debmirror  da debian.piffa.net
# un mirror locale da usare poi a casa anche senza internet
# deb file:/mnt/mirror sid main non-free contrib
```

1.7.2 /etc/apt/apt.conf

Questo file contiene le opzioni di apt, come ad esempio il proxy:

```
Acquire::http::Proxy "http://10.10.208.248:3128";
```

Si tenga conto che se si imposta un proxy per apt sul proprio portatile e tornati a casa propria si vuole scaricare nuovi pacchetti si dovra' disabilitare il proxy commentando la riga con ";" ("punto-e-virgola"). Su un portatile vi conviene tracciare il mirror `ftp.it.debian.org` senza impostare il proxy: in aula verra' rediretto al mirror locale e a casa vi appoggerete al mirror ufficiale.

E' consigliabile impostare il proxy per apt anche in presenza di un proxy-http *trasparente*.

2 Squid

Squid e' un proxy cache http (ma anche FTP e https) robusto e strutturato, puo' essere usato sia in situazioni relativamente semplici che in scenari piu' complessi grazie alla possibilita' di gestirne in modo granulare le risorse. Si partira' dalle configurazioni piu' semplici per la semplice *condivisione della navigazione* internet all'interno di una rete locale, per poi poter negoziare la gestione degli accessi, il filtraggio dei contenuti (Squid e' una applicazione che si muove nel 4' livello del modello TCP/IP a differenza di un *ipfilter* limitato al 2'), nel bilanciamento del carico tra piu' server proxy.

Inoltre Squid svolge la funzione di *anonymizer*:

Nasconde i client http alla rete internet: e' solo il server proxy ad accedere ai server web frequentati dai client: questi non sono percepiti ed esposti all'esterno della rete locale ma si relazionano solo con il server proxy. Dal punto di vista della sicurezza della rete locale questo e' preferibile ad un approccio alla navigazione basato su *NAT*.

Cosa a volte sottovalutata, Squid permette la navigazione web a una rete basata su *indirizzi IP privati* (es una 192.168.0.0/24). E se la rete privata deve *solo navigare* in internet, non serve introdurre nella rete un *NAT* (si veda la sezione sui firewall) per condividere la connessione: basta il solo Squid. Per altro non servira' neanche un servizio DNS accessibile dai clients dato che *sara' il solo Squid a risolvere i nomi di dominio* per i suoi client http.

Squid ascolta di default sulla porta 3128, per impostare *apt* per utilizzarlo si aggiunga ad `/etc/apt/apt.conf`

```
Acquire::http::Proxy "10.10.208.254:3128";
```

Per installare Squid si usino i pacchetti:

```
squid3
```

2.1 Configurazione: squid.conf

Segue un estratto delle direttive principali viste in aula presenti nel file di configurazione `/etc/squid3/squid.conf` .

2.1.1 Cache_dir

Cache dir serve per impostare dimensione e percorso della cache creata sul supporto di storaggio. Essendo la dimensione di default della cache pari a `~100 MB` e' altamente consigliabili aumentare questo parametro se si vuole poter utilizzare la funzione di *cache* http del software.

La dimensione ovviamente dipendera' dallo spazio disponibile, dimensioni tipiche e massime degli oggetti che si vuole tenere in cache (un solo file *.iso e' circa ``700 MB``, il pacchetto *Openoffice.org* circa `150 MB`, un pacchetto debian circa `20 MB`), numero dei client.

Si presti poi attenzione alla natura dei dati che saranno salvati nella cache: sono tutti dati facilmente sostituibili (gli originali sono *on-line*) la cui perdita non arreca danni permanenti. Questo rende la cache di Squid un possibile candidato ad un *RAID stirpe* (livello 0) a ad un file system che prediliga le performance a scapito della consistenza, con vantaggi sia per le prestazioni (e la velocita' di navigazione e' uno dei motivi per cui si installa Squid) che per l'utilizzo estensivo dello spazio di storaggio.

Questo fino al momento in cui per voi non sia piu' importante *garantire la disponibilita' del servizio*, ad esempio con un RAID *mirror* o 5 (se il RAID stripe dovesse rompersi gli utenti non potrebbero piu' navigare, cosa che per natura dello stripe e' maggiormente probabile rispetto ad un *mirror* o a un filesytem *normale*).

Altra considerazione: i dati del proxy vengono slavati sul filesytem del server dietro richiesta di utenti esterni talvolta sconosciuti. Come per i servizi di file sharing o per la posta elettronica non c'e' motivo che il filesystem su cui sono ospitati questi dati abbia i privilegi di eseguibilita' o suid (in genere si puo' anche

usare *noatime* per renderlo piu' veloce, che si usi o meno il journal dipende dalle preferenze: affidabilita' oppure prestazioni):

/etc/fstab

```
...
# Filesystem per Squid http cache
/dev/md3/       /var/spool/squid/       ext2,noexec,nosuid,noatime  0 3
```

Ora possiamo impostare la cache nel file di configurazione `/etc/squid3/squid.conf`:

```
#TAG: cache_dir (riga 1628)
#       Usage:
#
#       cache_dir Type Directory-Name Fs-specific-data [options]
#
#       You can specify multiple cache_dir lines to spread the
#       cache among different disk partitions.
#       ...
#Default:
# cache_dir ufs /var/spool/squid3 100 16 256
cache_dir aufs /var/spool/squid3 300 24 256
#         |    |                 |   |  secondo livello di directory
#         |    |                 |   directory primo livello
#         |    |                 dimensione in MB
#         |    path della cache
#         algoritmo
```

Se si modifica la struttura del filesytem della cache di Squid, ad esempio variando il numero delle directory, puo' essere opportuno rigenerare la struttura della cache di squid. Tipicamente e' consigliabile cancellare (se si ha *ridotto* il numero delle directory) la vecchia cache e poi generare una nuova struttura. Se si vuole *star nel sicuro* ogni volta che si modifica l'impostazione delle directory *si svuoti la vecchia cache e se ne generi una nuova*

```
# /etc/init.d/squid3 stop
# rm -r /var/spool/squid3/??
# squid3 -z
# /etc/init.d/squid3 start
```

2.1.2 TAG: maximum_object_size

Questa direttiva imposta la dimensione massima degli oggetti che vengono salvati sul supporto di storaggio, oggetti di dimensioni superiori saranno comunque scaricati ma non tenuti in cache.

TAG: maximum_object_size (1760):

```
# TAG: maximum_object_size (1760)
#       Objects larger than this size will NOT be saved on disk.  The
#       value is specified in kilobytes, and the default is 4MB.  If
#       you wish to get a high BYTES hit ratio, you should probably
#       increase this (one 32 MB object hit counts for 3200 10KB
#       hits).  If you wish to increase speed more than your want to
#       save bandwidth you should leave this low.
#
#       NOTE: if using the LFUDA replacement policy you should increase
#       this value to maximize the byte hit rate improvement of LFUDA!
```

```
#       See replacement_policy below for a discussion of this policy.
#
#Default:
# maximum_object_size 4096 KB
maximum_object_size 150 MB
```

2.1.3 TAG: cache_mem

Cache_mem imposta quanta memoria RAM venga utilizzata per la cache di Squid. Questo dipendera' dalla RAM disponibile sul sistema, e da quanta di questa volete mettere a disposizione di Squid (altri servizi importanti girano sulla stessa macchina?). Questo parametro influisce sulle prestazioni e sul degrado dei supporti di storaggio (sopratutto se magnetici).

Se si stesse pensando di usare dell'hardware *embedded* a basse prestazioni / consumo per realizzare un server gateway / NAT / Squid si tenga presente che Squid e' relativamente esoso di risorse rispetto agli altri servizi: avra' bisogno di `~25MB` (MegaByte) di RAM e *~150MHZ di CPU ARM* per servire decorosamente una decina di client http su una rete ethernet 10/100. In questo caso non fate scendere `cache_mem` sotto i `2/4 MB` pena un accesso continuo al supporto di storaggio.

Se invece si disponesse di una macchina dedicata a Squid con gigabytes di RAM non si esiti a dedicarne buona parte a *cache_mem*.

TAG: cache_mem (1566):

```
#       'cache_mem' specifies the ideal amount of memory to be used
#       for:
#               * In-Transit objects
#               * Hot Objects
#               * Negative-Cached objects
#Default:
# cache_mem 8 M
cache_mem 100 M
```

2.1.4 TAG: minimum_object_size

Questo parametro imposta la dimensione minima degli oggetti salvati nella cache. Settato a `0` o a valori molto piccoli puo' influire negativamente sulla frammentazione del filesytem e consumare un numero elevato di *inode* (cosa non piu' importante con ext4 o altri filesytem).

In scenari con connessioni molto veloci (>~10Mb), pochi client (una decina) e server poco performanti nella velocita' di accesso ai filesystem (~20MB/s, per quanto il limte sia piuttosto il *seek-time*) tenere in cache i file piu' piccoli aumenta la latenza della navigazione.

TAG: minimum_object_size:

```
#  TAG: minimum_object_size     (bytes)
#       Objects smaller than this size will NOT be saved on disk.  The
#       value is specified in kilobytes, and the default is 0 KB, which
#       means there is no minimum.
#
#Default:
# minimum_object_size 0 KB
minimum_object_size 0 KB
```

2.2 Negoziazione degli accesi al servizio

Squid e' uno di quei servizi soggetto a problemi di tipo *open relay* , si deve quindi limitare la rete che puo' accedere al servizio.

Open Relay:
Un servizio a cui possono accedere tutti indiscriminatamente. La cosa puo' andare bene per servizi come i server web, che aspirano per loro natura al maggior numero possibile di utenti, ma non a servizi come i proxy http oppure ai server di posta elettronica (adibiti ai soli utenti della rete locale).

Generalmente non volete che il vostro proxy http venga usato da persone sconosciute ed esterne alla vostra rete, le quali sostanzialmente *navigherebbero sotto l'identita' del vostro proxy* (probabilmente per visionare materiali che non vorrebbero fossero ricondotti direttamente a loro) generando traffico e consumando banda della vostra connessione a internet. Tenere Squid in modalita' *Open relay* e' al giorno d'oggi un buon modo per essere inseriti in una *black list*.

Per poter limitare gli accessi a Squid dal punto di vista dell'applicazione (quarto livello TCP/IP) si identifichera' inizialmente l'entita' *rete locale* (es: `localnet`) con una ACL di tipo *src* (indirizzi IP sorgenti) indicando la *classe / range di IP* della nostra rete.

Dopodiche l'accesso (`http_access`) si concedera' (*allow*) a questa entita' (es: `localnet`) negando chiunque altro.

Per maggiori dettagli sulla sintassi utilizzabile per esprimere i range di IP: http://www.visolve.com/squid/squid24s1/access_controls.php

E' poi sempre possibile tenere il proxy in ascolto su un solo indirizzo IP, nel caso si disponga di piu' devicei di rete, oppure settare un firewall per limitare il traffico dai primi livelli del TCP/IP.

2.2.1 ACL e http access

Si proceda a creare una `ACL` di tipo `src` per identificare la nostra rete locale, poi si abiliti l'accesso a questa con la direttiva `http_access`. Tutto quanto non e' espressamente autorizzato viene poi negato da un `http_access deny all` finale.

```
#  TAG: acl
#       Defining an Access List
#
#       Every access list definition must begin with an aclname and acltype,
#       followed by either type-specific arguments or a quoted filename that
#       they are read from.
#       ...
#       ***** ACL TYPES AVAILABLE *****
#
#       acl aclname src ip-address/netmask ...          # clients IP address
# riga 588

# Example rule allowing access from your local networks.
# Adapt to list your (internal) IP networks from where browsing
# should be allowed
#acl localnet src 10.0.0.0/8    # RFC1918 possible internal network
#acl localnet src 172.16.0.0/12 # RFC1918 possible internal network
#acl localnet src 192.168.0.0/16        # RFC1918 possible internal network
#
acl localnet src 10.10.208.0/24

# Riga 606
#  TAG: http_access
#       Allowing or Denying access based on defined access lists
```

```
#
#       Access to the HTTP port:
#       http_access allow|deny [!]aclname ...
#
#       NOTE on default values:
#
#       If there are no "access" lines present, the default is to deny
#       the request.

# Riga 643
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS

# Example rule allowing access from your local networks.
# Adapt localnet in the ACL section to list your (internal) IP networks
# from where browsing should be allowed
#http_access allow localnet
http_access allow localnet
```

2.3 Testare Squid

Configurato squid e' fondamentale testarne il corretto funzionamento per assicurarsi di non aver creato un *open-relay*. Per fare dei test significativi serve utilizzare degli host remoti: ci si connetta via ssh a questi per poi utilizzare `wget` da riga di comando.

2.3.1 Client: ~/.wgetrc

Nel file `.wgetrc` (si noti il punto iniziale: e' un file nascosto) si puo' impostare il proxy per wget. Si utilizzi l'indirizzo IP del server che si vuole testare, e si seguano i log `/var/log/squid3/access.log` sul server.

Da notare che la prova va' fatta su una macchina della rete che si vuole testare, non da *localhost*. Per altro se si utilizzasse *direttamente* `localhost` non si testerebbe la *ACL* predisposta, dato che si si rientrerebbe nella ACL (pre-configurata di default) `localhost`.

.wgetrc

http_proxy=10.10.208.178:3128

Si proceda a scaricare dal client scelto con un wget:

```
wget http://www.google.it
```

2.3.2 Server: access.log

Si puo' controllare il corretto funzionamento del server seguendo i log di accesso a Squid:

```
# tail -f /var/log/squid3/access.log
```

In oltre e' possibile configurare diversi *analizzatori di log* come `Webalizer` per studiare i log di Squid.

3 Tiny proxy

Se avete l'esigenza di un proxy server per la condivisione della connessione ad internet ma non avete la necessita' o le risorse di un *caching* proxy come Squid potete considerare **tinyproxy**, questo e' molto piu' leggero (utilizza circa ~2MB di RAM e ovviamente non deve accedere continuamente ad un file system per lo storaggio della cache) e risulta piu' semplice nella configurazione.

TinyProxy puo' essere utilizzato come sostituto di emergenza in una rete in cui Squid e' momentaneamente non disponibile.

File di configurazione: `/etc/tinyproxy/tinyproxy.conf`

```
# Porta su cui ascoltare
Port 3128
# IP su cui ascoltare
Listen 10.10.208.160
# Negoziazione accessi
Allow 10.10.208.0/24
```

4 Apache

Apache HTTP Server, o piu' comunemente Apache (*a patchy NCSA web server*), e' il server web modulare piu' diffuso e strutturato disponibile con licenza libera, in grado di operare da sistemi operativi UNIX/Linux e Microsoft.

Un server web e' un processo, e per estensione il computer su cui e' in esecuzione, che si occupa di fornire, su richiesta del browser, una pagina web (spesso scritta in HTML). Le informazioni inviate dal server web viaggiano in rete trasportate dal protocollo HTTP. L'insieme di server web da' vita al World Wide Web, uno dei servizi piu' utilizzati di Internet.

4.1 Pacchetti da installare::

apache2 apache2-doc

Con la release 2.0 di Apache viene automaticamente resa disponibile anche la versione SSL (Secure Socket Layer, connessioni criptate) del web server senza che ci sia la necessita' di installare altri pacchetti.

4.2 Configurazione di Apache

I file di configurazione di apache si trovano nella cartella: `/etc/apache2` e sono strutturati come descritto nel file `/usr/share/doc/apache2/README.Debian.gz` . Sostanzialmente lo schema e' il seguente:

apache2.conf
File di configurazione principale del servizio.
httpd.conf e' il vecchio file di configurazione di Apache1, presente per motivi di retrocompatibilita' e' generalmente vuoto.

ports.conf
In questo file vengono specificate le porte sulle quali resta in ascolto il server web. Si noti che utilizzando dei virtual hosts generalmente viene specificata per questi la porta su cui ascoltare nel file di configurazione del virtual host, ad es: `<VirtualHost *:80>`

sites-available
In questa cartella vengono raccolti i file di configurazione dei virtual host disponibili.

sites-enabled
In questa cartella sono contenuti dei link simbolici ai files in ../sites-available : se il link e' presente in questa cartella il virtual host e' abilitato.

mods-available
Stesso metodo per i moduli: in questa cartella ci sono i moduli veri e propri che verranno poi abilitati grazie all'esistenza di link simbolici nella cartella mods-enabled .

mods-enabled
Moduli abilitati, effettivamente caricati.

4.3 apache.conf

File di configurazione del servizio Apache, contiene le impostazioni generiche (ad esempio utilizzo della RAM e risorse di sistema) dell'intero servizio. Nella configurazione di default per Debian non viene definito un vero e proprio sito di default ma solo dei virtual hosts.

Guardiamo alcune direttive interessanti:

Timeout
Numero di secondi da aspettare prima di chiudere la connessione con il client. Questo parametro serve a liberare le risorse di sistema nel caso che un client, magari a causa di una connessione particolarmente lenta o instabili, tenga attivo indefinitamente un processo di apache.

KeepAlive
L'estensione keep-alive (http 1.0) congiuntamente alle connessioni persistenti (http 1.1) permettono al server di rispondere a piu' richieste dei client mediante la stessa connessione. Il protocollo http per sua natura e' senza stato (*stateless*), quindi ogni risorsa richiesta (per pagine web si pensi ad esempio alle immagini) dal client necessita di una connessione autonoma. Keep-alive permette di ottimizzare la connessione anche fino al 50% a seconda delle situazioni e contenuti.

Server-Pool Size Regulation
Questi parametri (StartServers, MinSpareServers, ecc. Tutti spiegati nel manuale di apache) servono per attribuire le risorse di sistema disponibili al server Apache. Tenere questi parametri bassi serve a limitare il rischio di Denial of Service per il server, nel caso offra altri servizi. I settagli di default sono come sempre abbastanza conservativi, se si conta di usare il proprio Apache per servire un sito web con molti visitatori sara' necessario aumentare sensibilmente le impostazioni di base.

AccessFileName
Il nome del file che viene onorato per modificare le impostazioni per una singola directory, legato alla direttiva AllowOverride .

4.4 Installazione di PHP

Pacchetti da installare: `php5 php-pear`

4.4.1 Test del modulo php

Creare nella cartella `/var/www` (o altra cartella visibile) un file con estensione *.php (es `/var/www/info.php` contenete codice php eseguibile dall'interprete, ad es:

```
<?php phpinfo() ; ?>
```

Questa funzione di php generera' la tipica pagina con le impostazioni attuali per PHP. Richiamando la pagina (es: `http://localhost/info.php`) verra' generata dall'interprete PHP la pagina HTML e resa disponibile tramite Apache ai client HTTP, a prova del corretto funzionamento del modulo di PHP e della sua integrazione con il server web Apache. In caso contrario se il client http proporra' di scaricare la pagina invece che visualizzarla nel browser: non funziona l'interprete di php o sono mal configurati i MIME-type. prima di tutto assicurarsi di aver fatto ripartire Apache.

4.4.2 Installazione del supporto per Mysql a PHP

Installare i pacchetti:

```
php5-mysql phpmyadmin
```

Controllare tramite la pagina php.info che sia abilitato il supporto per Mysql (ripartito Apache, ricaricare la pagina e cercare con CTRL+f `mysql`).

4.4.3 phpmyadmin

L'interfaccia web Phpmyadmin non richiede necessariamente la presenza di un database Mysql locale, puo' infatti essere utilizzata per gestire database remoti (il suo file di configurazione: `/etc/phpmyadmin/config.inc.php`). Nel caso si voglia installare localmente Mysql si utilizzi il pacchetto `mysql-server` .

Phpmyadmin dovrebbe essere disponibile all'URL: `http://localhost/phpmyadmin/`, se cosi non fosse controllare che sia incluso il file `/etc/phpmyadmin/apache.conf` in `/etc/apache2/conf.d/` .

4.4.4 Installazione del supporto per Postgresql a PHP

Installare i pacchetti:

```
php5-pgsql phppgadmin
```

Controllare tramite la pagina php.info che sia abilitato il supporto per PostgreSQL (ripartito Apache, ricaricare la pagina e cercare con CTRL+f `pgsql`).

4.4.5 phppgadmin

L'interfaccia web Phppgadmin per il database server PostgreSQL non richiede necessariamente la presenza di un database locale, puo' infatti essere utilizzata per gestire database remoti (il suo file di configurazione: `/etc/phppgadmin/config.inc.php`). Nel caso si voglia installare localmente Mysql si utilizzi il pacchetto `postgresql` .

Phpmyadmin dovrebbe essere disponibile all'URL: `http://localhost/phppgadmin/`, se cosi non fosse controllare che sia incluso il file `/etc/phppgadmin/apache.conf` in `/etc/apache2/conf.d/` .

4.5 Virtual hosts

- http://www.apacheweek.com/features/vhost
- http://www.onlamp.com/pub/a/apache/2004/01/08/apacheckbk.html

I virtual host permettono di avere piu' siti internet disponibili tramite lo stesso server web, eventualmente mappati su un solo indirizzo IP. Sono generalmente di due tipi:

- Basati su *indirizzi IP*. Se si ha la possibilita' di avere piu' indirizzi IP dedicati per i diversi siti che si vuole servire. ES: `<VirtualHost 192.168.0.2:80>` . Soluzione dispendiosa, si tende ad usarla solo se servono certificati di sicurezza (SSL) dedicati per ogni sito.
- Basati su *nomi di dominio* che puntano allo stesso IP. Soluzione piu' economica e diffusa che si basa sulle funzionalita' di http 1.1 .

Prenderemo in esame la gestione di virtual hosts basati su nomi di dominio.

4.5.1 Gestione DNS

Prima di tutto per poter impostare i virtual hosts dovete avere un server DNS che risolva i vostri nomi di dominio sull'indirizzo IP del server. Questo si puo' ottenere in vari modi, ad es:

`/etc/hosts`
Per prove sul proprio sistema potete impostare i nomi dei vostri virtual server nel file /etc/hosts .

***Dnsmasq* (DNS server)**
Utilizzabile al livello della rete locale per fare dei test, utilizzando direttive come: `address=/davide.piffa.net/10.10.208.178`

***Servizio DNS dinamico on line*.**
Utilizzare un servizio come ad es: https://www.dyndns.com/ per mappare nomi di dominio sul proprio indirizzo IP, comodo ad esempio se si dispone di un indirizzo IP pubblico (anche se dinamico) per la propria connessione ad internet.

***Bind* (DNS server)**
Impostare i campi A nelle proprie zone gestite dal server DNS Bind. Ad es: `papo A 212.22.136.248`

4.5.2 Eseguire una query DNS con `dig`::

Per testare la corretta risoluzione dei vostri nomi di dominio sui relaivi indirizzi IP si usi dig (o altre utlity, vedere la sezione relativa i DNS). Dig e' contenuto nel pacchetto `dnsutils`.

dig 177.piffa.net

; <<>> DiG 9.5.1-P1 <<>> 177.piffa.net ;; global options: printcmd ;; Got answer: ;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 38036 ;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION: ;177.piffa.net. IN A

;; ANSWER SECTION: 177.piffa.net. 0 IN A 10.10.208.177

;; SERVER: 10.10.208.248#53(10.10.28.248)

La parte interessante e' l'*ANSWER SECTION*: `177.piffa.net. 0 IN A 10.10.208.177` . Il nome di dominio 177.piffa.net viene risolto sull'ip 10.10.208.177 , nel nostro Apache (che risponde all'ip 10.10.208.177) dovra' essere disponibile un virtual host che corrisponde al nome `177.piffa.net` (`ServerName`) .

Il server DNS utilizzato dal sistema e' evidenziato dalla stringa: `;; SERVER: 10.10.28.248#53(10.10.28.248)` che dovrebbe corrispondere a quanto impostato nel vostro `/etc/resolv.conf`. Se il vostro browser web utilizza un proxy http sara questo a risolvere i nomi di dominio, tipicamente potete disabilitare l'uso del proxy per determinati domini nella sezione di configurazione del browser.

4.5.3 Virtual hosts

Esempio di Virtual host:

```
<VirtualHost *:80 >
    ServerName 177.piffa.net
    DocumentRoot /var/www/177.piffa.net/
    ServerAdmin webmaster@177.piffa.net
</VirtualHost>
```

1. `<VirtualHost \*:80 >` La prima riga indica l'inizio della stanza relativa al nostro virtual host, che ascoltera' su qualunque indirizzo IP (nel caso il server abbia piu' indirizzi dai quali e' raggiungibile) sulla porta `80`.
2. `Server/name` precisa quale sara' il nome di dominio a cui verra' associato questo sito rispetto ad altri eventualmente presenti sullo stesso server web.
3. `DocumentRoot` : il path della directory che contiene le pagine del sito.
4. `ServerAdmin`: l'indirizzo del webmaster, in modo da poterlo contattare in caso di problemi col sito.
5. `</VirtualHost>`: *tag* di chiusura della stanza di definizione del virtual host.

Quelle che abbiamo appena visto sono le direttive essenziali per definire un sito virtuale, potrebbe essere utile aggiungere altre:

- **`ErrorLog /var/log/apache2/177.piffa.net/error.log`**

Log degli errori separato dai restanti siti web ospitati dal server.

- **`LogLevel warn`**
  Livello di importanza degli eventi loggati: warning *attenzione* .
- **`CustomLog /var/log/apache2/177.piffa.net/access.log combined`**
  Log di accesso separati dagli altri siti, utile anche qua per statistiche di accesso per il solo sito virtuale.

Potrebbe essere utile modificare le impostazioni di una intera directory, ad esempio per abilitare l'`AuthConfig`:

```
<Directory "/var/www/miosito.net/privata">
        AllowOverride AuthConfig
        Options ExecCGI Indexes MultiViews FollowSymLinks
        Order allow,deny
        Allow from all
</Directory>
```

`AllowOverride AuthConfig` ora vale per l'intera directory, come le altre opzioni.

4.6 Negoziazione accessi

Tipicamente quando si installa un server web il proprio desiderio e' di dare accesso ai materiali disponibili al maggior numero di visitatori possibile. Talvolta pero' puo' essere utile poter limitare questi accessi, ad esempio per escludere un *bot* indesiderato che scansiona ininterrottamente le nostre pagine o per creare una *Area Riservata* i cui materiali non devono essere disponibile a tutti.

4.6.1 Limiti su base IP

La forma piu' semplice di restrizione degli accessi e' su base degli indirizzi IP dei client: tipicamente i siti web sono settati per dare accesso a chiunque:

```
<VirtualHost *:80 >
       # ...
       <Directory "/var/www/177.piffa.net">
         Order allow,deny
         Allow from all
       </Directory>
</VirtualHost>
```

Potremmo negare l'accesso a uno o piu' indirizzi IP in questo modo:

```
<VirtualHost *:80 >
       # ...
       <Directory "/var/www/177.piffa.net">
         Order allow,deny
         Allow from all
         Deny from 192.168.0.1
       </Directory>
</VirtualHost>
```

Ora l'IP 192.168.0.1 non potra' piu' accedere ai materiali dell'intero sito virtuale, oppure potremmo lavorare su una sola directory:

```
<Directory "/var/www/miosito.net/limitata">
       Order allow,deny
```

```
        Allow from 192.168.0.0./24
        Deny from all
</Directory>
```

In questo modo solo la classe IP `192.168.0.0/24` potra' accedere alla directory `/limitata` Si tenga pero' conto che e' relativamente facile per un malintenzionato cambiare il proprio indirizzo IP, oppure collegarsi da un altra zona. Meno facile e' accedere ad una classe privata trovandosi all'esterno di questa, ma ci sono comunque soluzioni piu' eleganti.

- Mod_access: http://httpd.apache.org/docs/2.0/mod/mod_access.html
- mod_authz_hosti(Available in Apache 2.1 and later): http://httpd.apache.org/docs/2.2/mod/mod_authz_host.html

4.7 User Authentication

Si puo' negoziare gli accessi ad un area del sito tramite autenticazione basata su *nome utente* / *password*. Questo puo' venire utile per creare una area download *intranet*, alla quale possano accedere solo gli utenti previsti a prescindere dagli indirizzi IP dei loro client.

Tramite il modulo di Apache *mod-auth* e' possibile implementare questo paradigma, per quanto esistano soluzioni piu' granulari e sofisticate, che richiedono pero' l'implementazione di interpreti di linguaggi di programmazione, criptazione delle passwords, gestione degli utenti ed eventualmente delle sessioni. Mod auth non richiede l'installazione di niente di tutto questo.

link: http://www.apacheweek.com/features/userauth

4.7.1 Definire la cartella

Decidere quale sara' il *path* della cartella da sottoporre ad autenticazione:

```
mkdir /var/www/177.piffa.net/privata
```

4.7.2 Creazione del database delle passwords

Un modo semplice per gestire una database di *user-id* / *passwords* e' utilizzare l'utility `htpasswd` di Apache. Questa crea un file in cui un *crypt* delle password viene associato agli utenti.

Si dovra' decidere dove tenere questo file, la cosa importante e' che non sia visibile nel sito web: non deve essere scaricabile dai visitatori. Deve essere cioe' all'esterno della *DocumentRoot*: un buon posto potrebbe essere la /home dell'utente.

Creiamo (con il *flag* `-c`) il file `/home/utente/passwords` con l'utente `luca`:

```
htpasswd -c /home/utente/passwords luca
```

`htpasswd` ci chiedera' la password da associare all'utente `luca`. Per successive modifiche della password o aggiunta di nuovi utenti non sara' necessario usare il flag `-c`.

4.7.3 Configurazione di Apache

Ora possiamo passare alla configurazione vera e propria di Apache, ma con una novita': andremo a inserire la voce in un `.htaccess` invece che modificare (tramite una direttiva `<Directory>`) il file di impostazione del virtual-host.

Questo per motivi pratici: solo l'utente *root* puo' modificare l'impostazione del virtual host nel file `/etc/apache2/sites-enabled/177.piffa.net`, ma spesso il motivo per cui creiamo i virtual hosts e' ospitare i siti di altri utenti, che possono solo pubblicare (generalmente tramite *FTP*) i loro documenti nella loro *DocumentRoot*, senza poter quindi modificare in alcun modo la configurazione del virtual host.

Dando agli utenti la possibilita' di modificare (*AllowOverride*) autonomamente alcuni parametri (in questo caso solo l'*AuthConfig*) relativi al funzionamento del loro spazio web ci togliera' l'incombenza di dover intervenire continuamente sui vari virtual host.

Abilitiamo l'AllowOverride nel file di configurazione del virtual host per la sola directory `privata`:

```
<VirtualHost *:80 >
    ServerName 177.piffa.net
    DocumentRoot /var/www/177.piffa.net/
    ServerAdmin webmaster@177.piffa.net
    <Directory "/var/www/177.piffa.net/privata">
        AllowOverride AuthConfig
    </Directory>
</VirtualHost>
```

Per rendere il cambiamento effettivo sara' necessario fare un restart / reload di Apache.

Ora sara' possibile, anche per l'utente di sistema, creare un file `.htaccess` che sara' onorato da Apache.

/var/www/177.piffa.net/privata/.htaccess

```
# Messaggio visualizzato al prompt per l'autenticazione
AuthName "Area privata soggetta ad autentizazione"
# tipo di autenticazione da usarsi
AuthType Basic
# File precedentemente generato con htpasswd
AuthUserFile  /home/utente/passwords

# Negoziazione degli accessi
# valid users permette l'accesso agli utenti specificati
# nel file generato da htpasswd
require valid-user
```

Si noti che non e' necessario fare ripartire Apache per onorare i cambiamenti (un utente non avrebbe la possibilita' di farlo!).

Oltre a `valid-users` si potrebbe scegliere di usare la formula `users` che permette di elencare esplicitamente gli utenti::

require user pippo pluto

L'utente *paperino* che fosse comunque presente nel file generato da htpasswd non potrebbe accedere alla risorsa.

Nel caso ci fossero molti utenti conviene gestirli tramite *gruppi*::

require group staff studenti

I gruppi vengono definiti in un file in modo simile a `/etc/groups` per gli utenti di sistema:

```
staff:andrea sara
studenti: lucap federico luca
```

da richiamare tramite la direttiva `AuthGroupFile`.

4.8 Cavets

Problemi di cache:

- Proxy: nei settaggi del browser specificare di non utilizzare un server proxy http per il sito web locale (o per gli altri che si stanno monitorando). Se si ha il controllo del proxy server: stopparlo, ricaricare la pagina (operazione che fallira'), far ripartire il proxy, ricaricare la pagina.
- Provare con un altro browser, o cercare di svuotare la cache chiudere/riaprire l'applicativo. Provare a fermare Apache, ricaricare la pagina (operazione che fallira'), far ripartire Apache, ricaricare la pagina.

5 Domain Name System

Domain Name System (spesso indicato con DNS) e' un servizio utilizzato per la risoluzione di nomi di host in indirizzi IP e viceversa. Il servizio e' realizzato tramite un sistema **gerarchico** (quindi una struttura ad albero, simile ai *file system*) **distribuito** (ogni server DNS facente parte del sistema puo' mantenere solo una parte delle informazioni, ad esempio per la sua sola *zona*), costituito dai server DNS.

I DNS sono un servizio *core* (fondamentale) per la rete internet come per qualunque rete locale. Ad esempio durante la navigazione web un client vorrebbe vedere l'*URL* `http://ww.piffa.net/`, quindi per potersi connettere via *http* al server web deve prima ottenere l'indirizzo IP del *server http* corrispondente a *www.piffa.net*. Se il DNS gli fornisce un IP sbagliato l'utente non potra' raggiungere il servizio: di fatto e' come se il serve http fosse spento.

Stessa cosa vale per gli altri servizi, come la posta elettronica, ssh, ecc. : *prima si deve effettuare una query DNS*.

Potrebbe verificarsi uno scenario simile a questo: i vostri server per i siti web funzionano correttamente come i siti ospitati, stessa cosa per i vostri server di posta, IMAP e POP3, e tutto il resto. Ma se poi un errore nella configurazione del DNS non rende raggiungibile l'intero *sito*: per l'utente finale e' come se nulla funzionasse.

Infatti quando si parla di un intervento della Polizia Postale per l'*oscuramento* di un sito dal punto di vista pratico questo si traduce generalmente nella rimozione o mistificazione del record DNS relativo a quel dominio (la *PP* ha facolta' di chiedere un simile intervento ai principali provider internet che forniscono connettivita' agli utenti italiani, oltre che poter agire direttamente sul NIC italiano per i domini della TLD *.it*)

L'operazione di convertire un nome in un indirizzo e' detta risoluzione DNS, convertire un indirizzo IP in nome e' detto risoluzione inversa.

> Un *Registar* e' un operatore che ha la facolta' (accreditamento da parte dell ICANN) di registrare i domini di secondo livello per gli utenti finali, dietro compenso di una modica cifra (una decina di euro) che vale come contributo su base annuale per il mantenimento dell'infrastruttura.

5.1 Risoluzione Inversa

Per la risoluzione inversa sono invece i provider di connettivita' a gestire i DNS: se volete impostare il *PTR* associato al vostro indirizzo IP dovete contattare il vostro provider (tipo *telecom* per una connessione ADSL) e *non il Registar del vostro dominio*.

Ad esempio all'IP `212.22.136.248` era associato un PTR `bender.piffa.net`, corrispondente al record `212` facente parte della zona `136.22.212.in-addr.arpa` gestito dal provider Tiscali/Nextra proprietario della classe C `212.22.136.0`. Se avete un solo IP conviene lasciare al fornitore la gestire del PTR, ma se avete a disposizione un'itera classe potete chiedere sempre al vostro provider che vi *deleghi* la gestione della zona tramite i vostri DNS.

Per alcuni servizi, ad esempio la spedizione della posta elettronica, e' richiedeiesto che venga impostata correttamente l'associazione tra il PTR dell'indirizzo IP usato dal server di postai e il record A RR al quale questo punta(RFC1912 sezione 2.1, paragrafo 2).

Si veda:

- http://www.faqs.org/rfcs/rfc1912.html 2.1 Inconsistent, Missing, or Bad Data

- http://www.ietf.org/rfc/rfc2505.txt

5.2 Nomi di dominio

Un nome a dominio e' costituito da una serie di stringhe separate da punti, ad esempio bender.piffa.net. I nomi di dominio si leggono da destra verso sinistra: *TLD* o dominio di primo livello `net`, secondo livello `piffa`, terzo livello `bender`. Il dominio di primo livello (o TLD, Top Level Domain, pronunciato *tilde* in Italia), per esempio .net o .it sono limitati e decisi direttamente dall'ente assegnatario ICANN (Internet Corporation for Assigned Names and Numbers).

L'utente finale potra' chiedere l'assegnazione (pagando un contributo al Register preferito per il mantenimento delle spese dell'infrastruttura) di un dominio di *secondo* livello (es `piffa`) di una delle varie TLD disponibili (noi italiani diciamo *tildi*), sempre che non sia gia' stato assegnato a qualcun altro.

Ottenuto il secondo livello sara' l'utente a gestirlo: potra' in stanziare domini di terzo livello (es `bender`) e anche oltre (es www.andrea.bender.piffa.net). Tali records saranno mantenuti dall'utente, sotto la sua responsbilita': se il proprio server DNS non fosse raggiungibile o risultasse mal configurato gli utenti non potrebbero risolvere / raggiungere i siti di loro interesse.

Tipicamente si ha almeno un server DNS secondario per garantire la sussistenza del servizio in caso di guasto del DNS principale. I secondari *replicano* i dati presenti nei DNS principali.

5.3 Tipologie di record

Ad un nome DNS possono corrispondere diversi tipi di informazioni. Per questo motivo, esistono diversi tipi di record DNS. Ogni voce del database DNS deve essere caratterizzata da un tipo. I principali tipi sono:

- Record A - Indica la corrispondenza tra un nome ed uno (o piu') indirizzi IP (per la precisione indirizzi IPv4, ovvero la versione attualmente in uso).
- Record MX - (Mail eXchange) indica a quali server debba essere inviata la posta elettronica per un certo dominio.
- Record CNAME - Sono usati per creare un alias, ovvero per fare in modo che lo stesso calcolatore sia noto con piu' nomi. Uno degli utilizzi di questo tipo di record consiste nell'attribuire ad un host che offre piu' servizi un nome per ciascun servizio. In questo modo, i servizi possono poi essere spostati su altri host senza dover riconfigurare i client, ma modificando solo il DNS.
- Record PTR - Il DNS viene utilizzato anche per realizzare la risoluzione inversa, ovvero per far corrispondere ad un indirizzo IP il corrispondente nome a dominio. Per questo si usano i record di tipo "PTR" (e una apposita zona dello spazio dei nomi in-addr.arpa).
- Record AAAA - Restituisce un indirizzo IPv6.
- Record SRV - Identificano il server per un determinato servizio all'interno di un dominio. Possono essere considerati una generalizzazione dei record MX.
- Record TXT - Associano campi di testo arbitrari ad un dominio. Questi campi possono contenere una descrizione informativa oppure essere utilizzati per realizzare servizi.

Vi sono anche tipi di record "di servizio", necessari al funzionamento del database distribuito: * Record NS - Utilizzato per indicare quali siano i server DNS autoritativi per un certo dominio, ovvero per delegarne la gestione. * Record SOA - (Start of Authority) usato per la gestione delle zone DNS.

5.4 Utilizzo

I computer vengono identificati in rete grazie agli indirizzi *IP*, questi pero' non sono comodi per gli utenti come riferimento per i vari server. Ad esempio sarebbe scomodo riferirsi al motore di ricerca Goggle con uno dei suoi IP: `74.125.43.104`, e' preferibile usare il nome di dominio *www.google.com*:

```
ping -c 1 www.google.com
PING www.l.google.com (74.125.43.104) 56(84) bytes of data.
```

5.5 Risoluzione dei nomi di dominio

Ci sono vari strumenti per interrogare i server DNS e ottenere l'indirizzo IP associato al nome di dominio che ci interessa:

```
$ host www.piffa.net
www.piffa.net is an alias for piffa.net.
piffa.net has address 65.98.21.97
piffa.net mail is handled by 10 65.98.21.97


$ nslookup www.piffa.net
Server:         192.168.0.10
Address:        192.168.0.10#53

Non-authoritative answer:
www.piffa.net   canonical name = piffa.net.
Name:   piffa.net
Address: 65.98.21.97


$ dig www.piffa.net

; <<>> DiG 9.6.0-P1 <<>> www.piffa.net
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 47751
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 4, ADDITIONAL: 4

;; QUESTION SECTION:
;www.piffa.net.                 IN      A

;; ANSWER SECTION:
www.piffa.net.          3489    IN      CNAME   piffa.net.
piffa.net.              3489    IN      A       65.98.21.97

;; AUTHORITY SECTION:
piffa.net.              86289   IN      NS      ns2.mydomain.com.
piffa.net.              86289   IN      NS      ns1.mydomain.com.
piffa.net.              86289   IN      NS      ns4.mydomain.com.
piffa.net.              86289   IN      NS      ns3.mydomain.com.

;; ADDITIONAL SECTION:
ns1.mydomain.com.       96208   IN      A       64.94.117.193
ns2.mydomain.com.       96208   IN      A       64.94.31.67
ns3.mydomain.com.       96208   IN      A       66.150.161.137
ns4.mydomain.com.       96208   IN      A       63.251.83.74

;; Query time: 1 msec
;; SERVER: 192.168.0.10#53(192.168.0.10)
;; WHEN: Sun May 10 21:23:11 2009
;; MSG SIZE  rcvd: 209
```

Lo strumento piu' esaustivo e' `dig`, installabile con il pacchetto `dnsutils` .

5.6 Dig

Vediamo alcune opzioni utili nell'utilizzo di `dig` per l'interrogazione dei DNS Server:

```
$ dig www.google.it

; <<>> DiG 9.6.0-P1 <<>> www.google.it
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 18816
;; flags: qr rd ra; QUERY: 1, ANSWER: 6, AUTHORITY: 7, ADDITIONAL: 0

;; QUESTION SECTION:
;www.google.it.                 IN      A

;; ANSWER SECTION:
www.google.it.          250683  IN      CNAME   www.google.com.
www.google.com.         334819  IN      CNAME   www.l.google.com.
www.l.google.com.       186     IN      A       74.125.43.103
www.l.google.com.       186     IN      A       74.125.43.104
www.l.google.com.       186     IN      A       74.125.43.147
www.l.google.com.       186     IN      A       74.125.43.99

;; AUTHORITY SECTION:
l.google.com.           80856   IN      NS      f.l.google.com.
l.google.com.           80856   IN      NS      d.l.google.com.
l.google.com.           80856   IN      NS      b.l.google.com.
l.google.com.           80856   IN      NS      c.l.google.com.
l.google.com.           80856   IN      NS      a.l.google.com.
l.google.com.           80856   IN      NS      e.l.google.com.
l.google.com.           80856   IN      NS      g.l.google.com.

;; Query time: 1 msec
;; SERVER: 192.168.0.10#53(192.168.0.10)
;; WHEN: Sun May 10 21:34:47 2009
;; MSG SIZE  rcvd: 255
```

$ dig

(senza opzioni o oggetti) Fornisce l'elenco dei *root server* utilizzati. I root server sono i server che mantengono le informazioni sui domini di primo livello (TLD) e sono quindi il punto di partenza per scorrere nella directory dei DNS per recuperare le informazioni (tipicamente un campo `A` per un indirizzo IP) che ci servono per raggiungere un certo servizio.

$ dig

...

;; ANSWER SECTION: . 192032 IN NS C.ROOT-SERVERS.NET. . 192032 IN NS E.ROOT-SERVERS.NET. . 192032 IN NS B.ROOT-SERVERS.NET. . 192032 IN NS L.ROOT-SERVERS.NET. . 192032 IN NS A.ROOT-SERVERS.NET. . 192032 IN NS F.ROOT-SERVERS.NET. . 192032 IN NS H.ROOT-SERVERS.NET. . 192032 IN NS G.ROOT-SERVERS.NET. . 192032 IN NS K.ROOT-SERVERS.NET. . 192032 IN NS M.ROOT-SERVERS.NET. . 192032 IN NS I.ROOT-SERVERS.NET. . 192032 IN NS J.ROOT-SERVERS.NET. . 192032 IN NS D.ROOT-SERVERS.NET.

...

dig @nome_dns

Permette di fare una query ad un server dns particolare. Es: `dig @151.99.25.1 www.google.it`

dig MX www.google.it
Chiede un campo in particolare, in questo caso il campo MX

dig ANY www.google.it
Chiede tutti i campi, non solo i campi *a*

dig -x 74.125.43.104
Effettua una richiesta inversa: dall'IP al PTR associato.

5.7 resolv.conf

Il file `/etc/resolv.conf` contiene le impostazioni sul dns usato dal sistema, in genere anche altre applicazioni che devono effettuare query DNS leggono resolv.conf per conoscere l'ubicazione del DNS.

/etc/resolv.conf:

- `nameserver`: indica il nameserver da utilizzare, indicato con l'indirizzo IP.
- `domain`: indica il nome di dominio della rete attuale, vedi voce successiva.
- `search`: nome di dominio usato dalla rete sul quale cercare gli hosts. Ad esempio se impostato su `piffa.net` pingando l' host `bender` viene automaticamente fatto un tentativo di ricerca per `bender.piffa.net`.

Predisponendo l'infrastruttura di rete della vostra LAN e' consigliabile impostare sempre almeno un DNS cache sul vostro server locale per i vari client. In questo modo in caso di malfunzionamento del DNS o necessita' di intervenire / sostituire i DNS non sara' piu' necessario dover reimpostare ogni singolo client della LAN: bastera' intervenire sul server DNS cache, ad esempio per utilizzare un nuovo forwarder, o modificare al volo un record DNS. La modifica, anche detta *mascheramento*, di un record come il *server smtp* o un *MX* potrebbe tirarvi rapidamente fuori dai guai nel caso di un problema improvviso con la posta elettronica o qualunque altro servizio che possiate reindirizzare col DNS.

Utilizzare un server DHCP e una DNS cache come `Dnsmasq` possono permettervi di risolvere al volo molte delle problematiche relative alla configurazione della vostra LAN: ad esempio dover intervenire manualmente su decine di client per modificare le impostazioni di SMTP | gateway | DNS | proxy.

Si veda anche la pagina man di resolv.conf.

> ***Avvertenza***
>
> Attenzione: se si usa un client DHCP, ppp (ADSL compresa) o simile questo file potrebbe' essere riscritto automaticamente in base a quanto ottenuto dal DHCP. Si veda la documentazione del pacchetto `resolvconf`.

5.8 /etc/hosts

Tabella statica per l'associazione tra IP e nomi di dominio:

```
# cat /etc/hosts
```

127.0.0.1 localhost.localdomain localhost 10.10.208.162 daniela daniela.piffa.net 10.10.208.254 mirror mirror.piffa.net 91.191.138.15 thepiratebay.org 192.168.0.11 chrome chrome.mydomain.com

Il contenuto del file e' un associazione tra un *IP* e stringhe di testo (anche piu' di una per IP) es: `mirror` o veri e propi nomi di dominio `mirror.piffa.net`. Si puo inserire un nome semplice come *casa* per riferirsi ad un host che si ha necessita' di contattare spesso, oppure mappare un nome di dominio completo su un indirizzo IP.

Il problema e' la gestione di questo file su molti hosts: quando gli host cambiano IP si devono aggiornare manualmente i records, operazione in se' non particolarmente gravosa ma che andra' fatta per ogni client della vostra LAN. Un metodo semplice per distribuire questo file e' utilizzare `Dnsmasq`: questo infatti legge e onora il file `hosts` che avete prodotto e lo rende disponibile ai clients tramite le query DNS.

Dnsmasq lavora come un server DNS, i vostri client lo interrogheranno per tradurre nomi di host e domini in indirizzi IP, risolvendo il problema della *distribuzione* del file `hosts` tra molteplici clients. Infatti il servizio DNS indica appunto una *directory distribuita* per la risoluzione dei nomi di dominio, risolvendo i problemi dell'aggiornamento e diffusione dei continui cambiamenti di questa.

Modificare la risoluzione di un nome di dominio esistente (ad esempio riconducendola a un IP interno) e' un modo drastico e funzionale per *annullarlo* rendendolo non disponibile alla propria rete locale. Ad esempio aggiungere al file `/etc/hosts`:

```
127.0.0.1       www.facebook.com
```

Impedira' agli utenti della LAN di raggiungere *facebook*, ora reindirizzato a `localhost`.

Oppure si potrebbe ricondurre l'indirizzo IP di un server HTTP pubblico usato per i downloads (ad esempio un mirror della propia distribuzione come `ftp.it.debian.org`) a un equivalente mirror creato all'interno della rete locale, riducendo il traffico verso l'esterno e aumentando notevolmente la velocita' di scaricamento.

5.9 Hostname

Ogni computer ha un *proprio nome* visualizzabile (e modificabile) con il comando `hostname`. Quando utilizzate a una shell su un host in genere l'hostname compare nel prompt della shell.

Per visualizzare il nome dell'host su cui si sta operando si digiti semplicemnte `hostname`, lo stesso comando con un oggetto modifica temporaneamente il nome dell'host. Per modificare in modo permanente il nome del computer si modifichi il contenuto del file `/etc/hostname`.

Si faccia attenzione a non aver un hostname puramente numerico: ad es. `161`. E' opportuno che il nome sia comunque un alfanumerico: `host-161` o simile.

5.9.1 FQDN

Per semplicita' gli host sono generalemente raggiungibili dall'esterno mappando il loro IP su un nome di dominio FQDN: fully qualified domain name, composto generalmente da *hostname*.``domain-name``, ad es. *bender*.``piffa.net``.

Alcuni servizi internet fanno affidamento sul PTR dell'IP del server per cercare una conferma che il *servizio* sia veramente chi afferma di essere (ad esempio STMP).

Non e' automatico che un servizio, ad esempio un server di posta, si qualifichi leggendo il contenuto del file `hostname` aggiungendo come suffisso il dominio della rete di cui fa parte l' host: a volte questo parametro puo' essere specificato nel file di configurazione del servizio:

```
* Squid (HTTP proxy): ``visible_hostname``

* Postfix (SMTP server): ``myhostname``
```

I motivi sono diversi, senza entrare nel dettaglio dei vari protocolli si pensi comunque che un host ha sempre un solo nome, ma puo' avere un numero variabile di *device di rete* sia fisici che virtuali con relativi *indirizzi IP*, e piu' servizi in ascolto sui vari IP.

6 DNSmasq

Dnsmasq puo' svolgere le funzioni di un DNS cache / forwarder, server DHCP, e' caratterizzato dalla facilita' di configurazione, limitato uso di risorse, adattabilita' a connessioni *dinamiche* come ADSL o altre punto a punto (anche via cellulari) per condividere rapidamente la rete (cosa molto utile se ci dovesse trovare a ridare connettetivita' a una rete momentaneamente sprovvista), dalla possibilita' di modificare rapidamente i record DNS serviti alla rete anche grazie alla distribuzione del file `/etc/hosts` locale. Puo' essere anche utilizzato come *server per il boot da rete <http://www.debian-administration.org/articles/478>_* .

Dnsmasq e' un interessante alternativa all'uso del server DNS Bind in modalita' *forwarding e cache-only* (non autoritativo) accompagnato dal server DHCPd. I vantaggi sono:

- Leggerezza: puo' essere fatto girare su una macchina relativamente debole in caso di bisogno.
- Rapidita' di configurazione (in particolare per servire dei record A / MX alla rete, modificando al volo i valori originali ospitati sul server DNS pubblico).
- Ben integrato con connessioni PPP : e' ingrado di rilevare i cambiamenti dei dns suggeriti e impostarli come forwarders (utile se dovete rendere disponibile rapidamente una connessione a internet a una rete in difficolta').

Tutto cio' rende Dnsmasq una soluzione valida in particolare quando si deve intervenire in una rete pre-esistente in cui il server principale e' in crisi: si potra' utilizzare Dnsmasq anche su una macchina piu' debole e *mascherare* i servizi al momento non disponibili. Molto utile per scopi didattici, sopratutto per testare server SMTP impostando al volo i campi MX per nomi di dominio fittizi.

6.1 Configurazione

Vediamo alcune direttive di basi del file di configurazione `/etc/dnsmasq.conf` utili per la configurazione sia del DNS cache che per il DHCP server:

domain-needed
Non inoltrare query ai server DNS esterni per nomi semplici (es andrea, portatile, pippo) che verranno risolti solo in locale o causeranno direttamente una risposta *not found* .

bogus-priv
Simile alla voce precedente ma per i reverse look-up.

domain
Nome di dominio della rete da passare ai client.

expand_hosts
Aggiunge il `nome host` (`/etc/hostname`) dei client al nome di dominio per qualificarli in rete, senza bisogno di dover comporre a un elenco statico di record nel file `/etc/hosts` o nello stesso file di configurazione di dnsmasq. Es: se un vostro client si chiama `chrome` e il vostro dominio `piffa.net` dnsmasq rendera' disponibile il campo *A* per il dominio `chrome.piffa.net` diretto all'ip che verra' assegnato al client.

6.2 DHCP

Per attivare il demone DHCP di Dnsmasq basta aggiungere al file di configurazione il *range* degli IP che si vuole assegnare ai client con il *lease time* (tempo di rilascio: quanto a lungo saranno validi gli IP assegnati) espresso in ore.

Si faccia *attenzione*: in una rete puo' essere presente **un solo server DHCP**, o per meglio dire qualunque server DHCP ascolta sul broadcast `255.255.255.255` e potrebbe rispondere a un pacchetto di richiesta DHCP. Quindi non fate partire inavvertitamente un server DHCP in una rete gia' servita e **non vi azzardate ad andare in giro con un portatile con un server DHCP attivo** nelle reti altrui. Questo vale anche per i laboratori di informatica dei corsi di reti: non fate partire il vostro server DHCP se siete collegati alla rete interna!

/etc/dnsmasq.conf (riga 118):

```
dhcp-range=192.168.0.20,192.168.0.50,24h
```

6.3 DNS cache

Dnsmasq lavora di default come cache dns: inserire al file `/etc/resolv.conf` il nameserver localhost in cima alla lista dei *nameserver* disponibili.

nameserver 127.0.0.1

Questo pero' potrebbe essere problematico se un altro servizio, ad esempio il DHCP client, riscrive il contenuto del file `/etc/resolv.conf`. Per superare il problema si aggiunga (riga 20) al file di configurazione `/etc/dhcp3/dhclient.conf`

```
prepend domain-name-servers 127.0.0.1;
```

Oppure potrebbe essere il nostro *PPP client* (per la connessione ADSL) a intervenire sul file `//etc/resolv.conf`, si modifichi quindi `/etc/ppp/peers/dsl-provider` commentando `usepeerdns`. Se la vostra connessione ad internet e' ADSL raramente dovreste aver bisogno di cambiare i DNS una volta impostati (a meno che non usiate un portatile!).

7 Bind : DNS Autoritativo

Le soluzioni viste possono bastare per la rete locale o per fare delle prove, ma prima o poi verra' il momento in cui si e' chiamati a gestire dei domini su internet: lo standard e' da sempre *Bind* (demone *named*), ora alla versione 9.

Installare i pacchetti:

```
bind9
```

7.1 DNS cache

Bind appena installato funzionera' come DNS cache: si faccia un test con un `dig @localhost` . Bind a differenza di Dnsmasq e' autonomo: non ha bisogno di forwardare (inoltrare) le query a un DNS esterno: queste verranno risolte direttamente da Bind partendo dai *DNS root servers*.

E' comunque possibile impostare dei DNS forwarders, tipicamente i DNS server forniti dal proprio provider, per velocizzare le query:

/etc/bind/named.conf.options (riga 13):

```
forwarders {
                10.10.208.254;
          };
```

Nel caso si voglia usare Bind solo come server DNS cache per la propria LAN senza ospitare delle zone DNS pubbliche sara' il caso di limitare gli accessi al server alla sola LAN:

/etc/bind/named.conf.options (riga 19):

```
// Se il proprio server ha IP 10.10.208.254
// sulla rete LAN privata:
listen-on { 10.10.208.254; }
```

E non si lasci il server in ascolto su uno degli eventuali indirizzi IP pubblici.

Se questo non fosse possibile si puo' sempre lavorare su una *acl*:

/etc/bind/named.conf

```
acl "localnet" {
        10.10.208.0/24 ; 127.0.0.0/8 ;
        } ;
```

Per poi aggiungere all'interno della stanza options la direttiva che abilita' l'entita' `localnet`:

/etc/bind/named.conf.options

```
allow-query {"localnet" ;} ;
```

7.2 Ospitare una zona

Se avete acquistato un nome di dominio e vi serve un software DNS per gestirlo Bind e' la scelta piu' diffusa. Ora vedremo come configurare una *zona* (come piffa.net) in modo che Bind sia autoritativoper questa, rispondendo alle query DNS di tutta la rete internet.

7.2.1 named.conf.local

Prima di tutti impostiamo il server bind per gestire la zona, per non fare confusione e' opportuno inserire le propie zone DNS nel file `named.conf.local` e non in `named.conf`.

named.conf.local:

```
/
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "piffa.net" {
        type master;
        file "/etc/bind/pz/piffa.net";
        }
```

type master
Il nostro server DNS sara' il principale, al quale poi potremo affiancare dei DNS secondari nel caso questo non sia disponibile.

file "/etc/bind/pz/piffa.net"
Dove verranno inserite le informazioni vere e propie di questa zona.

7.2.2 Configurazione della zona

Ora dovremo preparare il file contenente i record DNS della zona *piffa.net*, come abbiamo indicato prima questi saranno contenuti nel file `/etc/bind/pz/piffa.net` . Tenere le zone dentro una sottocartella e' buona abitudine, usare `pz` per queste e' una vecchia abitudine.

piffa.net:

```
; Zona per il dominio di secondo livello piffa.net

$TTL 3D     ; 3 days
```

```
@                   IN SOA  ns1.piffa.net. hostmaster.piffa.net. (
                                200905245  ; serial
                                8H         ; refresh (8 hours)
                                2H         ; retry (2 hours)
                                4W         ; expire (4 weeks)
                                1D         ; minimum (1 day)
                                )
;
                       NS      ns1
                       NS      ns2
                       A       94.23.63.105
                       MX      10 smtp
                       TXT     "Piffanet main site"
;
ns1             A       94.23.63.105
ns2             A       65.98.21.97
zoo             A       94.23.63.105
smtp            A       94.23.63.105
test.piffa.net. A       94.23.63.105
*.piffa.net.    A       94.23.63.105 ; *catch all domain
www             CNAME   zoo
ftp             CNAME   zoo
```

All'interno di questo file si possono inserire dei commenti con il carattere `;` (punto-e-virgola), si faccia attenzione alla rigida sintassi: apertura e chiusura delle parentesi tonde nella parte `IN SOA`, uso del `punto` finale per precisare un nome di dominio specifico (*FQDN*: Fully-qualified Domain Name) come `test.piffa.net.` a differenza degli altri domini di terzo livello come `pop,imap,smtp` .

La zona inizia con una direttiva `$TTL 3D` (RFC 2308) che indica la durata (in questo caso tre giorni) che ogni record dovrebbe avrebbe nella cache degli altri serber DNS. Questo valore dovrebbe essere superiore a un giorno, se non modificate spesso i valori dei vostri record DNS e' consigliabile settarlo a 2/3 settimane in modo da limitare la frequenza delle query al propio server. Questo parametro puo' essere modificato per singoli record:

```
$TTL 3D     ; 3 giorni: default se non specificato altrimenti
rapido  5h      IN      A       94.23.63.105 ; usa un TTL di 5 ore
lento   3w      IN      A       94.23.63.105 ; usa un TTL di 3 settimane
normale         IN      A       94.23.63.105 ; usa il TTL di default: 3 giorni
```

Segue poi il nome della zona, indicato con la `@` per richiamare la zona originale precisata nel file `named.conf.options` . Segue il campo SOA.

7.2.2.1 SOA: Start of Authority Record

Il record SOA puo' comparire solo una volta in una zona, contiene informazioni relative all'autorita' del server DNS.

ns1.piffa.net. name-server
primary master DNS di questo dominio.

hostmaster.piffa.net. email-addr
email-addr: indirizzo email della persona responsabile di questa zona, il primo punto viene tradotto in una *chiocciola* `@` dato che questo carattere ha un'altro utilizzo all'interno di questo file. Il referente della zona **deve** essere un email valido e controllato, come consuetudine si usa `hostmaster@dominio.tilde` .

200905245 serial number
Questo valore serve per indicare quando e' stato modificato questo file di configurazione, secondo il formato `yyyymmddss`: `yyyy` = anno, "mm" = mese, "dd" = giorno, "ss" = seriale. Il seriale che deve

essere sempre specificato anche per una cifra, va incrementato di una unita' nel caso vengano fatte piu' modifiche *nello stesso giorno*.

8H refresh
Indica ai DNS secondari quanto tempo attendere per cercare di aggiornare i loro dati con il DNS master.

2H retry
Intervallo di tempo per il DNS slave (secondario) da aspettare prima di cercare di ricontattare il *master* in caso di problemi col *refresh*.

4W expire
Indica quando i dati dei dns secondarinon sono piu' autoritativi in caso di impossibilita' degli *slaves* di ri-aggiornarsi con il *master*. Consigliato un valore di 2/4 settimane.

1D minimum
Questo valore indicava il TTL fino alla versione 8 di Bind, da Bind 9 e secondo la RFC2308 indica la durata del *negative caching*, quanto i resolvers (ad esempio un server dns cache) puo' mantenere un record *negativo* (che non indica la corrispondenza tra un nome di dominio e un ip, ma la non esistenza del record). Nell'uso per il negative caching viene fissato un valore massimo di 3 ore dalla RFC 2308.

7.2.2.2 Altri campi:

All'interno della zona possono essere utilizati vari tipi di records (RR):

TXT
Informazioni testuali associate ad un record

NS
Name Server della zona. Non deve essere un cname.

A
Indirizzo ipv4 da associare al record

AAA
Indirizzo ipv6 da associare al record

CNAME
Canonical Name: un alias per un host: ad esempio per il dominio piffa.net possiamo settare degli alias come `www.piffa.net`, `http.piffa.net`, `virtual.piffa.net`, `ftp.piffa.net`, `imap.piffa.net`. Comodo quando diversi alias sono sempre riferiti allo stesso ip.

MX
Mail Exchanger: server di posta che si occupera' della posta elettronica per questo dominio.E' opportuno avere almeno un server di posta di back-up, per indicare la priorita' di un MX rispettoad un altro si usa un valore di 2 cifre: il valore piu' basso indica priorita' piu' bassa. Es: `MX 10 smtp.piffa.net.` per il server SMTP principale e `MX 40 smtp2.piffa.net` per il secondario. Non deve essere un cname.

PTR
Reverse look-up, usato per la mappatura inversa di un indirizzo ip a una stringa identificativa dell'host. Si noti che per poter modificare questi record si deve avere *in gestione* la *zona IP*, se cosi' non fosse si dovra' chiedere al propio provider la modifica di questo record per il propio ip. Links: http://www.zytrax.com/books/dns/ch3/

7.3 DNS slave

Data l'importanza del servizio DNS e' necessario avere ridondanza per i server DNS che ospitano i vostri dati: in caso di indisponibilita' del server *master* (nel caso fosse il solo a tenere i dati questo comporterebbe la *scomparsa* di tutti i servizi / host da esso seviti!) il client potrebbe contattare uno degli *slave*.

Gli slave recuperano i dati dei recordos RR direttamente dal master e non sara' quindi necessario dover mantenere manualmente il file di configurazione della zona sugli slaves, ogni volta che aggiorneremo il master questi dati si propaghera' agli slaves automaticamente.

Per attivare uno *slave* per la nostra zona di esempio `piffa.net` si inserisca nel file `named.conf.local` dello slave server:

```
zone "piffa.net" {
        type slave;
        file "/etc/bind/pz/piffa.net";
        masters { 192.168.0.1; };
        };
```

Facendo ripartire Bind il file `/etc/bind/pz/piffa.net` viene creato automaticamente.

Segue un estratto di `/var/log/syslog` al `restart` di `bind9` sullo slave:

```
... slave named[2256]: zone piffa.net/IN: loaded serial 200905245
... slave named[2256]: running
... slave named[2256]: zone piffa.net/IN: sending notifies (serial 200905245)
... slave named[2256]: client 192.168.0.1#1464: received notify for zone 'piffa.net'
... slave named[2256]: zone piffa.net/IN: notify from 192.168.0.1#1464: zone is up to date
```

Avvertenza

Bind9 (versione 9.3 presente in Debian Lenny) richiede una esplicita autorizzazione alla notifica per lo stesso server slave, che in fase di avvio interroghera' (inviando un notify) se' stesso per valutare se i dati relativi alla zona di cui e' slave sono aggiornati. Si aggiunga quindi al file `/etc/bind/named.conf.options` dello slave: `allow-notify { 192.168.0.1; };` all'interno della stanza `options`, in cui l'inidirizzo IP inserito e' quello dello stesso slave server.

7.4 Aggiornamento dinamico: nsupdate

Dalla versione 8 di Bind e' dsponibile l'utility `nsupdate` (disponibile nel pacchetto `dnsutils`) per aggiornare automaticamente i record di una zona secondo il paradigma client / server (RFC2136) . Posto che abbiate a disposizione un server DNS Bind on-line su un indirizzo IP fisso e un zona da gestire (che potrebbe essere anche solo la delega di un dominio di terzo livello come *casa.miodominio.net*) sara' possibile aggiornare automaticamente i record che tirano a degli indirizzi IP *pubblici ma dnamici*, come quelli spesso messi a disposizione dei provider per le connessioni ad internet residenziali, in modo da poter rendere sempre raggiungibile la vostra workstation a casa anche dopo un aggiornamento dell'ip dinamico associato alla connessione.

L'auenticazione del client nsupdate che avra' la possibilita' di aggiornare il server DNS master avviene tramite *Transaction signatures* (TSIG, RFC2845) usando un algoritmo di criptazione dati asimmetrico *HMAC-MD5* : generata una coppia di chiavi sul client / nsupdate con l'utility si dovra' trasferire la chiave pubblica sul server *master*, che verra' configurato per onorare gli aggiornamenti (eliminazione e inserimento di record RR) autenticati dalla chiave privata.

7.4.1 Configurazione client (nsupdate)

Sul client, sul quale non deve essere necessariamente installato un server DNS Bind ma la sola utility `nsupdate`, generiamo la coppia di chiavi con l'utility `dnssec-keygen` installabile tramite il pacchetto `bind9utils`:

```
dnssec-keygen -a HMAC-MD5 -b 512 -n USER home.piffa.net.
```

Otterremo le due chiavi `Khome.piffa.net.+157+04331.key` `Khome.piffa.net.+157+04331.private`, la chiave pubblica dovra' essere resa noto al server master che ricevera' l'update dei records.

7.4.2 Configurazione server: riconoscimento chiave

Per rendere nota al server la chiave pubblica generata sul client si aggiunga quindi al file `/etc/bind/named.conf` sul server::

key home.piffa.net. {
algorithm HMAC-MD5; secret "txfAkNTScANEu2V73mCeiDpXNc3pmf+7ONOoKnTKQKIZMzierSmeHjK5 Z8ntnByt/PJwv26jCIsVh8n+xzVsRw==";};

> ***Nota***
>
> La parte `secret`, che potete leggere direttamente nel file *.key della chiave genearta, e' scritto *tutto sulla stessa riga* senza ritorni a capo.

7.4.3 Server: gestione dell'intera zona

Sul server modifichiamo il file di configurazione `named.conf.local` della zona della quale vogliamo concedere l'aggiornamento al client:

```
zone "piffa.net" {
        type master;
        file "/etc/bind/pz/piffa.net" ;
        allow-update {
                        key  home.piffa.net;
                        };
};
```

Sara' necessario assicurarsi che il demone di Bind sia in grado di modificare il file `/etc/bind/pz/piffa.net`: dato che questo file ora sara' gestito da lui si proceda a cedergli la propieta' del file::
chown bind /etc/bind/pz/piffa.net

Altro problema che si potrebbe porre: gli orologi di sistema dei due host devono essere sincronizzati per poter valutare l'opportunita' di un aggiornamento: si consigla di installare su entrambi l'utility `ntpdate` e di eseguirla facendo riferimento ai time server di Debian:

```
apt-get install ntpdate
ntpdate-debian
```

Ora possiamo provare dal client a effettuare l'iserimento di un record per testarne il funzionamento:

```
# nsupdate -k Khome.piffa.net.+157+04331.private -v
> server ns1.piffa.net
> update add home.piffa.net. 86400 A 192.168.0.2
> show
Outgoing update query:
;; ->>HEADER<<- opcode: UPDATE, status: NOERROR, id:      0
;; flags: ; ZONE: 0, PREREQ: 0, UPDATE: 0, ADDITIONAL: 0
```

```
;; UPDATE SECTION:
home.piffa.net.         86400   IN      A       192.168.0.1


> send
```

Per comprendere meglio l'uso dell'utility `nsupdate` si consiglia la lettura della relativa pagina man. Nella prima riga viene invocato il comando `nsupdate` impostando col flag `-k` la chiave privata generata precedentemente, con `server` si imposta quale server NS autoritario della zona (che abbiamo precedentemente configurato per ricevere gli aggiornamenti) vogliamo contattare. Alla riga sucessiva `update` viene aggiunto un record `A` per la il dominio `home.piffa.net` indirizzato all'IP `192.168.0.2`, poi `show` mostra quanto ci si prepara a comunicare al server con il finale `send` .

Si noti che in questo modo *l'intera zona* piffa.net e suscettibile di essere modificata dal client, che potra' eliminare e inserire qualunque record. E' possibile gestire in modo piu' granulare la zona, ad esempio concedendo al client i privilegi per gestire solo una parte della zona o i tipo di record da gestire.

7.4.4 Automatizzare l'aggiornamento dinamico

Nsupdate risulta comodo per tenere aggiornati i record DNS degli host connessi ad internet con indirizzi IP dinamici (pubblici) assegnati dal provider. Il client deve essere in grado di contattare autonomamente il server DNS per comunicare un cambiamento del suo ip. Vediamo innanzi tutto un primo script per nsupdate:

```
#!/bin/bash
# Diamo al demone ppp un po' di tempo per negoziare la connessione
# prima di leggere l'IP ottenuto
sleep 15
IPADDR=$(/sbin/ifconfig ppp0 | awk '/inet/ { print $2 } ' | sed -e s/addr://)

nsupdate  -k /root/dns/Khome.piffa.net.+157+04331.private <<-EOF
        server 192.168.0.254
        zone home.piffa.net.
        update delete home.piffa.net. A
        update delete home.piffa.net. MX
        update add home.piffa.net. 432000 A $IPADDR
        update add home.piffa.net. 432000 MX 10  home.piffa.net.
        show
        send
        EOF
```

Questo script legge il valore del device di rete `ppp0` creato dal *pppoe* di una connessione ADSL per ottenere l'indirizzo IP ottenuto dal provider (prima di farlo aspetta 15 secondi per dare il tempo al `pppoe` di negoziare la connessione).Vengono poi eliminati gli esistenti valori `A` e `MX` per `home.piffa.net` (si noti il punto finale dopo *net*) e inseriti quelli attuali.

Resta da decidere quando richiamare questo script: l'evento che causa l'assegnazione del nuovo IP in questo caso e una nuova connessione `pppoe`, quindi sarebbe consigliabile inserire lo script nelle routine comprese in `/etc/ppp/ip-up.d` (si veda la documentazione di ppp), nel caso questo non desse i risultati sperati (per problemi di connessione) come via estrema si consideri di mettere lo script nella routine del demone `cron` in modo che venga eseguito periodicamente (ad esempio ogni giorno).

7.5 Link suggeriti:

- DNS for Rocket Scientists http://www.zytrax.com/books/dns/
- DNS HOWTO http://www.langfeldt.net/DNS-HOWTO/BIND-9/

8 Samba

Samba e' un progetto libero che fornisce servizi di condivisione di file e stampanti a client SMB/CIFS.

Samba e' liberamente disponibile, al contrario di altre implementazioni SMB/CIFS, e permette di ottenere interoperabilita' tra Linux, Unix, Mac OS X e Windows.

Samba e' un software che puo' girare su piattaforme che non siano Microsoft Windows, per esempio, UNIX, Linux, IBM System 390, OpenVMS e altri sistemi operativi. Samba utilizza il protocollo TCP/IP utilizzando i servizi offerti sul server ospite. Quando correttamente configurato, permette di interagire con client o server Microsoft Windows come se fosse un file e print server Microsoft agendo da Primary Domain Controller (PDC) o come Backup Domain Controller, puo' inoltre prendere parte ad un dominio Active Directory.

8.1 Pacchetti

Pacchetti da installare per utilizzare Samba in modalita' client [2]

```
samba-client
```

Pacchetti da installare per utilizzare Samba in modalita' server:

```
samba smbfs smbclient
```

Durante la prima installazione viene chiesto il nome del gruppo di appartenenza, il default per Windows e' `WORKGROUP`. In aula usiamo invece `208` .

Per riconfigurare Samba si usi il comando:

```
dpkg-reconfigure samba-common
```

8.2 Passwords e autenticazione

Per poter configurare Samba in modo che usi un sistema di negoziazione degli accessi alle cartelle condivise basato su accoppiate *nome utente* / *password* bisogna distinguere tra 3 livelli di password (e generalmente volete usare *sempre la stessa password* per ognuno di questi) e delle differenze tra le modalita' di *autenticazione* (e quindi anche di criptaggio delle passwords) usate da sistemi GNU/Linux e Windows:

1 Sistema *Unix (GNU/Linux)
E' la password dell'*utente di sistema* che viene usata sul sistema operativo su cui gira il software Samba. E' importante tenere conto anche delle *user-id* e *group-id* degli utenti che dovranno fisicamente scrivere sui file system. Se un utente non puo' scrivere in una certa posizione del file system (ad esempio nella cartella `/mnt/condivisione` che sara' stata necessariamente creata inizialmente dall'utente `root`) per mancanza dei privilegi di scrittura allora neanche Samba potra' farlo nel momento in mette a disposizione la risorsa all'utente. Se si montano file-system dedicati per le condivisioni controllare i permessi e proprieta' dei *punti di mount**. Queste passwords sono salvate nel solito file /etc/shadow (richiamato da /etc/passwd).

2 Password per l'applicativo Samba
Samba deve essere compatibile con Windows e quindi utilizzare un sistema di criptazione delle password diverso da /etc/shadow . Le password per Samba possono essere gestite ad esempio col comando `smbpasswd` e vengono generalmente salvate all'interno di `/var/lib/samba/passdb.tdb` .

3 Password per Windows.
Gli utenti Windows effettuano il log-in alla partenza della sessione di Windows. Se si avra' l'accortezza di usare sempre la *stessa password* data precedentemente anche a Windows (o viceversa impostare la

password per GNU/Linux / Samba uguale a quella di Windows) l'utente potra' accedere automaticamente alle condivisioni a lui disponibili.

8.3 Creazione Utenti

Creiamo per primo l'utente sotto GNU/Linux, facendo attenzione a *non dargli una shell di sistema*. Gli utenti Windows che accedono al server solo per le condivisioni non hanno bisogno di poter eseguire comandi sul server!

Creazione di un utente denominato sambo:

```
adduser --shell /bin/false sambo
```

Nel file `/etc/passwd` avremo qualcosa come:

```
sambo:x:1001:1001:Sambo utente Samba,,,:/home/sambo:/bin/false
```

Aggiunta dell'utente al database delle password per Samba e generazione della sua password:

```
smbpasswd -a sambo
```

Se successivamente si vorra' modificare la password di un utente gia' esistente si usi:

```
smbpasswd  sambo
```

La password sotto Windows verra' modificata sul sistema Windows.

8.4 Creare la condivisione

La condivisione altro non e' che una cartella sul server che viene resa disponibile ai client negoziando l'accesso in base a una autenticazione basata su *user-name / password*. E' per altro possibile permettere l'accesso a una risorsa a chiunque indiscriminatamente (a tutti i `guest`) ma la cosa e' sconsigliabile dal punto di vista della sicurezza. Si decida se la cartella condivisa debba risiedere nella *home* di un utente (nel caso quest'ultimo ne sia l'unico fruitore) o in una cartella in /mnt/ (nel caso piu' utenti accedano a questa). Nel secondo caso si potranno gestire gli accessi sotto GNU/Linux tramite i gruppi.

Creazione della risorsa sambo_share nella home dell'utente sambo:

```
# mkdir /home/sambo/sambo_share
# chown sambo:sambo /home/sambo/sambo_share/
```

8.4.1 Sicurezza: permessi di esecuzione sul server

Bisognerebbe notare sul server i permessi di esecuzione del file-system che ospita la cartella da condividere. Se i file che saranno contenuti nella condivisione saranno da usarsi sotto Windows non c'e' motivo che questi siano eseguibili sotto GNU/Linux. Si potrebbe avere quindi, ipotizzando una condivisione in `/mnt/share` che risieda su di un file system dedicato:

`/etc/fstab`

> /dev/hda10 /mnt/share ext3 rw, **nosuid,noexec** 0 3

Si noti anche l'uso di *nosuid* per evitare la possibilita' di eseguire programmi con credenziali diverse.

8.5 Configurazione dell'applicativo Samba vero e proprio.

Avendo preparato gli utenti (ancora una volta: non si dia una shell completa a un utente che serve solo per Samba o la posta elettronica) e la cartella sul file system si puo' procedere a configurare la condivisione su Samba.

/etc/samba/smb.conf riga ~235 , Share Definitions (in vim si usi 235gg):

```
[sambo_share]
        # Percorso della cartella condivisa
        path = /home/sambo/sambo_share
        # Se gli utenti possono scrivere / modificare file
        writable = yes
        # Negoziazione degli accessi su base utenti / passwords
        valid users = sambo

        # ########################################
        # Altri parametri opzionali di interesse
        # Se posso vedere la condivisione da esplora risorse
        # anche se non ho i privilegi per accedervi.
        browseable = yes
        # Commento indicativo della risorsa
        comment = Condivisione per Sambo
```

Dopo aver salvato il file si puo' fare un primo controllo tramite l'utility `testparm` , che controlla la sintassi del file di configurazione di Samba. Se questo non rileva problemi si puo' procedere a un `# /etc/init.d/samba restart` .

8.5.1 Creazione di un gruppo

Se si deve condividere una risorsa con un numero consistente di utenti e' consigliabile lavorare in termini termini di gruppi piuttosto che elencare la lista degli utenti in `valid users`.

Dopo aver creato il gruppo del quale volete facciano parte i vostri utenti (`addgroup nome_gruppo`), inserite i vostri utenti nel gruppo (`adduser nome_utente nome_gruppo`) e modificate la direttiva `valid users` in `smb.conf` per riferirsi ad un gruppo piuttosto che a degli utenti. Per riferirsi a un gruppo si usi il carattere `@ chicciola` col `nome_del_gruppo`:

```
# Negoziazione degli accessi su base gruppo
valid users = @nome_gruppo
```

8.6 Testare il Servizio

Come testare il servizio

es:

```
smbclient -U sambo -L localhost
```

Questo comando permette di esplorare la risorsa qualificandosi come utente, in questo modo potete testare il corretto funzionamento dell'autenticazione. Si provi inizialmente a sbagliare la password deliberatamente, poi a inserirla correttamente: dovrebbero essere visibili le risorse disponibili al solo utente sambo: la suo /home e la cartella samba_share:

```
Sharename       Type      Comment
---------       ----      -------
```

```
sambo_share     Disk      Condivisione per Sambo
print$          Disk      Printer Drivers
IPC$            IPC       IPC Service (base server)
sambo           Disk      Home Directories
```

In particolare l'ultima voce relativa alla home directory dell'utente dovrebbe essere visibile solo agli utenti autenticati.

In alternativa e' possibile montare realmente la condivisone anche su GNU/Linux tramite un client per samba e testarne il corretto funzionamento:

```
mount -t smbfs //localhost/sambo_share /mnt/sambo_mount/ --verbose -o user=sambo
```

9 Server di posta: Postfix

Il server di posta che prenderemo in considerazione e' Postfix, a seguire un estratto di un file di configurazione *semplice* con l'abilitazione delle *Maildir* nelle `/home` degli utenti per la consegna della posta:

`/etc/postfix/main.cf`:

```
# ...segue dalla riga ~30
myhostname = 162.piffa.net
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = 162.piffa.net
mydestination = 162.piffa.net, localhost
# Se non avete un ip pubblico e statico, con un adeguato record PTR
# dovrete usare un realy host per l'invio della posta
relayhost = smtp.piffa.net

mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
# Se dovete inviare la posta per i client della vostra LAN privata:
# mynetworks = 127.0.0.0/8 192.168.0.0/24 [::ffff:127.0.0.0]/104 [::1]/128
# E si faccia BEN ATTENZIONE a non diventare un open realay smtp


# Per effettuare lo storaggio della posta nelle home directory degli utenti
# in una Maildir invece che nella Mailbox in /var/mail/nome_utente
# si disabiliti procmail
#mailbox_command = procmail -a "$EXTENSION"

# cartella_i abiliti lo storaggio della posta nella Maildir/ (si noti lo slash)
# all'interno della  home dell'utente:
home_mailbox = Maildir/
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
```

E' disponibile un file di configurazione di esempio ben piu' articolato e commentato::
/usr/share/postfix/main.cf.dist .

9.1 Test del server smtp

Per testare il corretto funzionamento del server di posta si puo' procedere in vari modi.

- Spedire una mail a una casella locale / remota e controllare i log (syslog)

- Collegarsi via *telnet* al server di posta: http://www.netadmintools.com/art276.html
- usare una utility come SWAKS

9.1.1 Swaks

Per gli utenti meno esperti e' consigliabile utilizzare *SWAKS*: si installi l'omonimo pacchetto e si esegua un test con::

swaks --to utente@destinatario.tilde --from utente@propio.mail.tilde

Ecco un esempio di una sessione corretta:

```
swaks --to andrea@piffa.net from andrea@mydomain.com
=== Trying smtp.piffa.net:25...
=== Connected to smtp.piffa.net.
<-  220 zoo.piffa.net ESMTP Postfix (Debian/GNU)
 -> EHLO alice.mydomain.com
<-  250-zoo.piffa.net
<-  250-PIPELINING
<-  250-SIZE 10240000
<-  250-VRFY
<-  250-ETRN
<-  250-STARTTLS
<-  250-ENHANCEDSTATUSCODES
<-  250-8BITMIME
<-  250 DSN
 -> MAIL FROM:<root@alice.mydomain.com>
<-  250 2.1.0 Ok
 -> RCPT TO:<andrea@piffa.net>
<-  250 2.1.5 Ok
 -> DATA
<-  354 End data with <CR><LF>.<CR><LF>
 -> Date: Thu, 28 May 2009 13:11:19 +0200
 -> To: andrea@piffa.net
 -> From: root@alice.mydomain.com
 -> Subject: test Thu, 28 May 2009 13:11:19 +0200
 -> X-Mailer: swaks v20061116.0 jetmore.org/john/code/#swaks
 ->
 -> This is a test mailing
 ->
 -> .
<-  250 2.0.0 Ok: queued as 41FB261AFC
 -> QUIT
<-  221 2.0.0 Bye
=== Connection closed with remote host.
```

9.2 Imap e pop

Postfix e' un server SMTP, di conseguenza se volete che i vostri utenti possano *scaricare* in locale la posta generalmente volete mettere a loro disposizione un server *POP3* o *IMAP*. Oppure entrambi.

Pacchetti da installare

courier-imap courier-pop

Si noti che IMAP necessita delle *Maildir*, non funziona con le Mailbox in `/var/mail/` .

9.3 Client a riga di comando

Per testare il corretto funzionamento del server di posta e' utile avere a disposizione delle utility per inviare e leggere la posta: ovviamente da riga di comando.

9.3.1 mailx

Uno dei client piu' semplici, sopratutto per inviare un messaggioi. e' sufficiente usare una formula come::

mail utente@dominio.com

Se il comando `mail` non fosse disponibile si installi il pacchetto `mailx`.

Al primo prompt si digitera' l'oggetto, il testo del messaggio (per terminare l'inserimento lasciare una riga vuota, digitare un `punto + Invio` su una riga vuota), la Carbon Copy (se necessaria).

es:

```
mail andrea@localhost
Subject: Oggetto della mail
Testo del messagio,
per terminare il messaggio
lasciare una riga vuota
e un punto (poi Invio).

.
Cc:
```

Per altrre opzioni si veda la pagina man.

9.3.2 Mutt

Mutt e' uno dei gestori di posta preferiti da chi preferisce utilizzare l'interfaccia testuale per la gestione della posta.

Mutt ha un file di configurazione `.muttrc` nella *home* dell'utente, alcuni settaggi possono essere utili:

set folder="~/Maildir"
Per utilizzare `/home/nome_utente/Maildir` come `mailbox`, invece del default `/var/mail/nome_utente`.

set editor="vim"
Utilizzare `vim` come editor per comporre i messaggi.

Spesso e' utile poter *levvere al volo* la Mailbox / Maildir di un utente sul server di posta, per controllare se i messaggi vengono recapitati correttamente:

```
mutt -f /var/mail/utente
mutt -f /home/utente/Maildir
```

In modo analogo si puo' consultare al volo la propia mailbox su un server remoto tramite IMAP/POP:

```
mutt -f imap://nome_utente@piffa.net
```

9.3.3 Web client

Per mettere a disposizione degli utenti un client web per gestire la propria posta si installi il pacchetto: `squirrelmail` . Ci sono tanti altri client web disponibili: questo e' particolarmente semplice. Naturalmente dovrete aver installato: `php5 apache2` .

L'interfaccia dovrebbe essere disponibile all'url: `http://localhost/squirrelmail` . Se cosi' non fosse assicuratevi che Apache abbia incluso il file di configurazione di squirrelmail:

```
cd /etc/apache2/conf.d/
ln -s /etc/squirrelmail/apache.conf ./squirrelmail.conf
```

9.4 Graylisting

Il *graylisting* e' un sistema relativamente poco invasivo, con un limitato consumo di risorse per limitare lo *SPAM* in arrivo sul propio server di posta. Come suggerisce il nome e' una via di mezzo tra una *white list* (una lista di mittenti privilegiata, sempre benvenuti) e una *black list* (mittenti *bannati*, banditi dal poter inviare nuovi messaggi).

Il funzionamento e' relativamente semplice: ogni mittente sconosciuto viene immediatamente rifiutato con un errore *non grave* come un *server non disponibile, provare piu' tardi*. Questo inconveniente non dovrebbe mettere in difficolta' un server di posta / mittente legittimo, che dopo un periodo di attesa tentera' nuovamente di inviare il messaggio ottenendo finalmente il risultato atteso. Diversamente un *bot* per l'invio di SPAM o un applicazione improvvisata (tipicamente di derivazione virale) che stesse inviando il messaggio *probabilmente* non insisterebbe, rinunciano ad inviare il messaggio preferendo destinazioni meno problematiche.

9.4.1 Abilitazione in Postfix

Installare il pacchetto: `postgrey` e aggiungere il file di configurazione di Postfix `/etc/postfix/main.cf`:

```
smtpd_recipient_restrictions =
        permit_mynetworks,
        reject_unauth_destination,
        check_policy_service inet:127.0.0.1:60000
```

9.4.2 Test

Inviando un messaggio il client dovrebbe ricevere un iniziale messaggio di rifiuto del messaggio:

```
swaks --to andrea@piffa.net from andrea@mydonain.com
=== Trying smtp.piffa.net:25...
=== Connected to smtp.piffa.net
...
<-  250 2.1.0 Ok
 -> RCPT TO:<andrea@piffa.net>
<** 450 4.2.0 <andrea@piffa.net>: Recipient address rejected:
Greylisted, see http://postgrey.schweikert.ch/help/piffa.net.html
 -> QUIT
<-  221 2.0.0 Bye
=== Connection closed with remote host.
```

A lato server si dovrebbe rilevare su `/var/log/syslog` qualcosa di simile:

```
connect from alice.mydomain.com[65.98.21.97]
May 28 14:53:34 r24266 postgrey: action=greylist, reason=new,
   client_name=alice.mydomain.com,
   client_address=10.0.0.1, sender=root@alice.mydomain.com, recipient=andrea@piffa.net
May 28 14:53:34 r24266 postfix/smtpd[22538]:
   NOQUEUE: reject: RCPT from alice.mydomain.com[10.0.0.1]:
```

```
   450 4.2.0 <andrea@piffa.net>: Recipient address rejected: Greylisted,
   see http://postgrey.schweikert.ch/help/piffa.net.html;
   from=<root@alice.mydomain.com> to=<andrea@piffa.net>
   proto=ESMTP helo=<alice.mydomain.com>
May 28 14:53:34 r24266 postfix/smtpd[22538]: disconnect from alice.mydomain.com[10.0.0.1]
```

9.4.3 Statistiche

E' sempre utile poter tracciare qualche statistica sulle percentuali di messaggi ricevuti, da chi, messaggi rifiutati (e per quale motivo). Statistiche che attingono dai soliti log del server di posta `/var/log/syslog` di default oltre che i dedicati `/var/log/mail` .

Una utility semplice per analizzare l'attivita' del propio server smtp potrebbe essere `pflogsumm` , installato il pacchetto la si puo' invocare con:

```
pflogsumm.pl  /var/log/mail.log
```

oppure utilizzare i log piu' vecchi ad es. `/var/log/mail.log.0`

10 Firewall

In Informatica, nell'ambito delle reti di computer, un firewall (termine inglese dal significato originario di parete refrattaria, muro tagliafuoco, muro ignifugo; in italiano anche parafuoco o parafiamma) e' un componente passivo di difesa perimetrale che puo anche svolgere funzioni di collegamento tra due o piu' tronconi di rete. Usualmente la rete viene divisa in due sotto reti: una, detta esterna, comprende l'intera Internet mentre l'altra interna, detta LAN (Local Area Network), comprende una sezione piu' o meno grande di un insieme di computer locali. In alcuni casi e' possibile che si crei l'esigenza di creare una terza sotto rete detta DMZ (o zona demilitarizzata) atta a contenere quei sistemi che devono essere isolati dalla rete interna ma devono comunque essere protetti dal firewall.

Una prima definizione chiusa di firewall e' la seguente:

Apparato di rete hardware o software che filtra tutti i pacchetti entranti ed uscenti, da e verso una rete o un computer, applicando regole che contribuiscono alla sicurezza della stessa.

In realta' un firewall puo' essere realizzato con un normale computer (con almeno due schede di rete e software apposito), puo' essere una funzione inclusa in un router o puo' essere un apparato specializzato. Esistono inoltre i cosiddetti "firewall personali", che sono programmi installati sui normali calcolatori, che filtrano solamente i pacchetti che entrano ed escono da quel calcolatore; in tal caso viene utilizzata una sola scheda di rete.

La funzionalita' principale in sostanza e' quella di creare un filtro sulle connessioni entranti ed uscenti, in questo modo il dispositivo innalza il livello di sicurezza della rete e permette sia agli utenti interni che a quelli esterni di operare nel massimo della sicurezza. Il firewall agisce sui pacchetti in transito da e per la zona interna potendo eseguire su di essi operazioni di: controllo modifica monitoraggio

Questo grazie alla sua capacita' di "aprire" il pacchetto IP per leggere le informazioni presenti sul suo header, e in alcuni casi anche di effettuare verifiche sul contenuto del pacchetto.

10.1 Links

- http://openskill.info/topic.php?ID=124
- http://iptables-tutorial.frozentux.net/iptables-tutorial.html

10.2 Ipfilter

Link: http://iptables-tutorial.frozentux.net/iptables-tutorial.html#IPFILTERING

Natura di un firewall ip: su cosa lavora (livello 2 e un po' del 3) e su cosa *non* lavora (livello 4). Netfilter lavora anche su parti del livello 3 (TCP, UDP, etc) e del livello 1 (MAC source address). Iptables comunque permette di fare il *connection-tracking*, mediante il quale possiamo implementare il Network Address Translation.

Netfilter non ricostruisce il flusso di dati tra pacchetti, non puo' quindi rilevare la presenza di virus o simili che si trasmettono su pacchetti separati: ricomporre, analizzare e tornare a scomporre i frammenti richiederebbe troppa RAM e risorse di sistema, con il conseguente rischio di saturare il firewall fino all'abbandono dei nuovi pacchetti in transito. Ci sono altri software piu' adatti a questi compiti, ad esempio un proxy HTTP come Squid che e' appunto una applicazione di quarto livello, progettata e strutturata per analizzare e modificare i flussi di dati (il *contenuto* dei pacchetti, non le sole *intestazioni*) facendo abbondate uso delle risorse RAM e di calcolo del sistema. Non a caso su macchine embedded dalle prestazioni molto ridotte (CPU ARM ~250MHZ con ~30MB di RAM) Squid sfrutta al massimo le risorse di sistema per gestire il traffico di una rete 10/100, mentre il lavoro tipico svolto da netfilter e' quasi irrilevante.

10.3 Progettazione di un firewall

Per implementare un firewall bisogna decidere un aio di cose: la collocazione e l'approccio (inclusivo o esclusivo) al filtraggio, il tipo di hardware.

10.3.1 Collocazione

DMZ e MZ, internet, intranet, extranet. Frammentazione della rete, decidere se diversi reparti di una azienda si possano vedere tra loro e in che misura.

Collocazione:

1. sul router
2. tra router e servers / LAN
3. Unico server / router / firewall e connessi rischi. considerare l'acquisto di un router hardware dedicato.

Layeed security:
Implementare piu' device / software sui diversi livelli: http://iptables-tutorial.frozentux.net/iptables-tutorial.html#HOWTOPLANANIPFILTER

10.3.2 Policy di default

Drop o Accept: conseguenze per sicurezza, facilita' di gestione.

10.3.3 Hardware

Sostanzialmente potremmo distinguere due tipologie di hardware:

Network appliance dedicata::
Un dispositivo hardware dedicato alla funzione di Firewall, ad es un Cisco / Fortigate. Si noti che molti firewall economici altro non sono che Linux box molto striminzite.

Server / Personal computer:
Un server sul quale viene fatto girare Netfilter ad uso del server stesso e della rete connessa.

Vantaggi e svantaggi: consumo elettrico, efficienza, flessibilita', strumenti di gestione, sicurezza, OpenBSD.

10.4 Percorso dei pacchetti tra tabelle e catene

link: http://iptables-tutorial.frozentux.net/iptables-tutorial.html#TRAVERSINGOFTABLES

10.5 Concetti di base

10.5.1 Tabelle, catene, regole

Iptables lavora su 3 tabelle (tables) di default:

- filter - Regola il firewalling: quali pacchetti accettare, quali bloccare
- nat - Regola le attivita' di natting
- mangle - Interviene sulla alterazione dei pacchetti.

Ogni tabella ha delle catene (chains) predefinite (INPUT, OUTPUT, FORWARD ...) a cui possono essere aggiunte catene custom. Ogni catena e' composta da un elenco di regole (rules) che identificano pacchetti di rete secondo criteri diversi (es: -p tcp --dport 80 -d 10.0.0.45) Ogni regola termina con una indicazione (target) su cosa fare dei pacchetti identificati dalla regola stessa (es: -j ACCEPT, -j DROP ...)

10.5.2 Match

I Match di una regola (rule) servono a testare un pacchetto per valutare se corrisponda a certe caratteristiche. I match di possono servire a controllare se un pacchetto e' destinato a una porta particolare o utilizza un protocollo particolare.

Alcuni esempi:

-p [!] proto
Protocollo IP. Secondo IP number o nome (es: tcp, udp, gre, ah...)

-s [!] address[/mask]
Indirizzo IP sorgente (o network con maschera di sotto rete)

-d [!] address[/mask]
Indirizzo IP destinazione (o network)

-i [!] interface[+]
Interfaccia di rete di entrata ([+] wildcard)

-o [!] interface[+]
Interfaccia di rete di uscita ([+] wildcard)

`-f`	Frammento di pacchetto

10.5.3 Targets

Se un pacchetto soddisfa le condizioni del Match *salta* (jump) su uno dei target possibili, in caso contrario continua il suo percorso tra regole catene e tabelle.

Target principali:

-j ACCEPT
Il pacchetto matchato viene accettato e procede verso la sua destinazione. Si usa per definire il traffico permesso.

-j DROP
Il pacchetto viene rifiutato e scartato, senza alcuna notifica al mittente. Si usa, in alternativa a REJECT, per bloccare traffico.

-j REJECT
Il pacchetto viene rifiutato. Al mittente viene mandato un pacchetto (configurabile) di notifica tipo ICMP port-unreachable (--reject-with icmp-port-unreachable)

`-t LOG`	Il pacchetto viene loggato via syslog e procede l'attraversamento della catena. Opzioni: (--log-level, --log-prefix, --log-tcp-sequence, --log-tcp-options, --log-ip-options)

`-j DNAT`	Viene modificato l'IP di destinazione del pacchetto. Target disponibile solo in nat / PREROUTING e nat / OUTPUT. L'opzione --to-destination IP:porta definisce il nuovo IP di destinazione. Si usa tipicamente su network firewall che nattano server di una DMZ
`-j SNAT`	Viene modificato l'IP sorgente. Solo in nat / POSTROUTING. Prevede l'opzione --to-source IP:porta. Si usa per permettere l'accesso a Internet da una rete locale con IP privati.
`-j MASQUERADE`	Simile a SNAT, si applica quando i pacchetti escono da interfacce con IP dinamico (dialup, adsl, dhcp...). Si usa solo in nat / POSTROUTING e prevede l'opzione --to-ports porte.
`-j REDIRECT`	Redirige il pacchetto ad una porta locale. Usabile solo in nat / PREROUTING e nat / OUTPUT e' previsto per fare un transparent proxy (con proxy server in esecuzione sulla macchina con iptables)
`-j RETURN`	Interrompe l'attraversamento della catena. Se questa e' una secondaria, il pacchetto torna ad attraversare la catena madre da punto in cui aveva fatto il salto nella secondaria. Se il RETURN e' in una delle catene di default, il pacchetto interrompe l'attraversamento e segue la policy di default.
`-j TOS`	Usabile solo nella tabella mangle, permette di cambiare il TOS (Type Of Service) di un pacchetto con l'opzione --set-tos. Per un elenco dei parametri disponibili: iptables -j TOS -h
`-j MIRROR`	Curioso e sperimentale, questo target invia un pacchetto speculare al mittente. In pratica e' come se facesse da specchio per tutti i pacchetti ricevuti. Da usare con cautela, per evitare attacchi DOS indiretti.

10.6 Tabella Filter

E' quella implicita e predefinita (-t filter) Riguarda le attivita' di filtraggio del traffico. Ha 3 catene di default: INPUT - Riguarda tutti i pacchetti destinati al sistema. In entrata da ogni interfaccia. OUTPUT - Riguarda i pacchetti che sono originati dal sistema e destinati ad uscire. FORWARD - Riguarda i pacchetti che attraversano il sistema, con IP sorgente e destinazione esterni.

Esempio per permettere accesso alla porta 80 locale: iptables -t filter -I INPUT -p tcp --dport 80 -j ACCEPT Analoga a: iptables -I INPUT -p tcp --dport 80 -j ACCEPT

Esempio per permettere ad un pacchetto con IP sorgente 10.0.0.4 di raggiungere il server 192.168.0.1 attraversando il firewall: iptables -I FORWARD -s 10.0.0.4 -d 192.168.0.1 -j ACCEPT

10.7 Flush automatico per macchine remote

Se state provando una configurazione del firewall per una macchina remota e' buona norma per evitare brutte figure attivare uno script che faccia il *flush* delle regole dopo qualche minuto. Potreste infatti inavvertitamente impostare una regola che vi impedisca di raggiungere la macchina remota, cosi' da non poter neanche eliminare quella regola e ripristinare la situazione precedente.

Veramente, prima di lavorare sul firewall di una macchina remota impostate almeno un `at now +5 min` o con un'oretta di margine per fare il *flush* delle regole (su tutte le tabelle):

```
    at now +5 min
at> /sbin/iptables -F
at> [CTR+d]
```

10.8 Gestione regole (rules)

Il comando iptables viene usato per ogni attivita' di gestione e configurazione.

Inserimento regole:

iptables -A CATENA ...
Aggiunge una regola alla fine della catena indicata

iptables -I CATENA [#] ...
Inserisce alla riga # (default 1) una regola nella catena indicata

iptables -N CATENA
Crea una nuova catena custom

iptables -P CATENA TARGET
Imposta il target di default per la catena indicata

Rimozione regole e azzeramenti:

iptables -F [catena]
Ripulisce tutte le catene (o quella indicata)

iptables -X [catena]
Ripulisce tutte le catene custom (o quella indicata)

iptables -Z [catena]
Azzera i contatori sulle catene

iptables -D catena #
Cancella la regola numero # dalla catena indicata

Interrogazione:

iptables -L
Elenca le regole esistenti

iptables -L -n -v
Elenca, senza risolvere gli host, in modo verboso le regole esistenti

10.9 Salvataggio regole

Il comando `iptables` serve per interagire con il framework `Netfilter` che gestisce il firewall di Linux al livello del kernel. Questo comporta, in modo analogo a quando avviene col comando `ifconfig`, che i cambiamenti impostati siano in *tempo reale, RAM*, non persistenti nel sistema: al boot successivo del sistema tutto tornera' alle impostazioni di base (in questo caso *nulle*, con policy di default settate su `ACCEPT` per tutto).

Le varie invocazioni di iptables potrebbero essere richiamate da degli scripts dedicati, ma fortunatamente e' stata predisposta una apposita utility per gestire questi scripts in modo da avere a disposizione un *formato standard* per il salvataggio e il ripristino delle regole del firewall.

Altro problema: decidere quando attivare / disattivare queste regole. Utilizzare i *runlevels* non e' una soluzione adeguata: le regole del firewall sono legate all'attivita' delle schede di rete (e un host con diverse schede di rete puo' attivarle a secondo delle esigenze di routing, partenza di servizi es file_sharing per un back-up...): il sistema operativo Debian permette di legare l'esecuzione di comandi alla attivazione di una device di rete (`up`), dopo la sua attivazione (`post-up`, utile per devices che richiedono un certo tempo per inizializzarsi: come un tunnel o una connessione punto a punto), prima della sua attivazione (`pre-up`). Allo stesso modo sono disponibili eventi analoghi per accompagnare la disattivazione dei device di rete: si veda la pagina man di `interfaces`.

Nel nostro caso avremo per una possibile scheda `eth0`:

`/etc/network/interfaces`

```
iface eth1 inet static
        up /sbin/iptables-restore /root/firewall/basic_fw
        # Seguono i soliti parametri della scheda di rete
        address 10.10.208.21
```

10.9.1 Iptables-save

Per salvare le regole di iptables attualmente presenti nel kernel si usi il comando:

```
# iptables-save >> /root/firewall/basic_fw
```

Il contenuto del file dovrebbe essere *comprensibile*: sostanzialmente sono regole di iptables, senza il comando iptables ripetuto, suddivisi per le varie tabelle. Potete comunque correggere eventuali parametri con un edito di testo.

Se non avete un'idea migliore potreste voler tenere gli script dei firewall in una cartella `~/firewall` nella home directory dell'utente `root`.

10.9.2 Iptables-restore

Per ripristinare un set di regole precedentemente salvate con `iptables-save` si utilizzi `iptables-restore`. Se questo deve essere fatto in modalita' *non interattiva*, ad esempio deve essere eseguito dal demone che si occupa di inizializzare le schede di rete, oppure un *cron* o altro, e' buona norma richiamare i percorsi completi sia dei comandi che dei file:

```
/sbin/iptables-restore /root/firewall/basic_fw
```

10.10 Esempi

Seguono alcuni esempi sull'uso di iptables, lo scenario e' un computer con un paio di schede di rete fisiche una delle quali collegata alla rete internet l'altra a una rete privata per la LAN interna.

1. `eth0` scheda di rete principale sulla rete privata interna 192.168.0.0/24
2. `eth1` scheda di rete secondaria per la connessione ad internet
3. `ppp0` punto-a-punto per una connessione ad internet

10.10.1 Bloccare i ping dall'esterno

Spesso gli script che attaccano *automaticamente* le varie reti provano a fare un ping per verificare quali IP sono on-line: bloccare il traffico `ICMP` in ingresso puo' aiutare ad evitare parte di questi attacchi:

```
iptables -A INPUT -i ppp0 -p ICMP -j DROP
```

10.10.2 Masquerading (sNAT)

Per attivare la network address translation (in questo caso un SNAT) per la rete locale privata sull'indirizzo ip del *modem*::

iptables -A POSTROUTING -s 192.168.0.0/255.255.255.0 -o ppp0 -j MASQUERADE

Il *Masquerading* a differenza dello *SNAT* puro (`-j SNAT --to-source proprio_ip_pubblico`) legge `l'indirizzo ip del device ``ppp0`. In questo modo se l'IP cambia automaticamente si aggiorna anche il source natting. Se avete un indirizzo IP statico assegnato al vostro gateway potete

invece usare lo SNAT semplice.

Altri esempi::

Change source addresses to 1.2.3.4. # iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 1.2.3.4

Change source addresses to 1.2.3.4, 1.2.3.5 or 1.2.3.6 # iptables -t nat -A POSTROUTING -o eth0 -j SNAT --to 1.2.3.4-1.2.3.6

Change source addresses to 1.2.3.4, ports 1-1023 # iptables -t nat -A POSTROUTING -p tcp -o eth0 -j SNAT --to 1.2.3.4:1-1023

10.10.3 Brute force

Per limitare attacchi di tipo brute force su SSH:

```
iptables -A INPUT -i ppp0 -p tcp -m tcp --dport 22 -m state --state NEW -m recent --update --seconds 3000 --hitcount 4 --name DEFAULT --rsource -j DROP

iptables -A INPUT -i ppp0 -p tcp -m tcp --dport 22 -m state --state NEW -m recent --set --name DEFAULT --rsource
```

11 FTP Server

Il File Transfer Protocol (FTP) (protocollo di trasferimento file), è un Protocollo per la trasmissione di dati tra host basato su TCP, in genere usato dagli autori di pagine web per *pubblicare* queste nei propio spazi web. Storicamente veniva anche usato, mediate l'utilizzo di utenze anonime, come punto di scambio per materiali di vari utenti tra loro sconosciuti (una directory dei materiali scaricabili e una dedicata agli *uploads* degli utenti, poi riordinati dall'*ftpmaster*). Tuttora si mantiene la cosuetudine di renedere disponibile i materiali dei *mirrors* anche tramite FTP, probabilmente per garantire l'accesso ai client piu' datati che non possono utilizzare tecnologie piu' recenti.

Il protocollo FTP e' in chiaro (cioe' non criptato), sia per quanto riguarda il traffico ad esso associato che per il passaggio delle passwords degli utenti, facilmente sniffabili da chiunque abbia accesso alla rete. Naturalmente vsftp per quanto votato alla sicurezza non modifica queste caratteristiche del protocollo FTP (ma consente di usare OpenSSL per la autenticazione degli utenti).

Se propio si deve mettere a disposizione un server FTP ai propi utenti si considerino le seguenti alternative:

- Spingere gli utenti ad usare SFTP invece che FTP
- Spingere gli utenti ad usare SSL per autenticarsi al server FTP
- Nel caso di webdesigners si consideri la possibilita' di offrire alternative come GIT, Subversion, Rsync o Webdav

Nel caso non si possa evitare il server FTP:

- Non dare agli utenti FTP una shell di sistema (Concedere come shell `ftp` al posto di `bash` in `/etc/passwd`)
- Rendere il filesytem su cui scrive il demone FTP `noexec` e `nosuid` (vedi dopo)
- Utilizzare un demone FTP come Vsftp: un server FTP con una forte inclinazione alla sicurezza: *Very Secure FTP Daemon*.

Per maggiori informazioni sulle scelte di design legate alla sicurezza del demone si veda: http://vsftpd.beasts.org/DESIGN

Vsftp mette a disposizione le seguenti funzionalita':

- Virtual IP configurations
- Virtual users
- Standalone or inetd operation
- Powerful per-user configurability
- Bandwidth throttling
- Per-source-IP configurability

- Per-source-IP limits
- IPv6
- Encryption support through SSL integration

11.1 Pacchetti

Per installare il demone vero e propio si usi il pacchetto `vsftpd` , mentre per aver un client da cui fare qualche test sono dipsonibili:

- `ftp` (pacchetto da installare) e' il solito client a riga di comando
- `gftp` e' un client grafico simile al classico *WSftp*
- Normalmente i file mananager com Konqueror possono lavarorare come client FTP

11.2 Sessioni ftp

Vediamo alcuni dei comandi di base per gestire una sessione ftp a riga di comando:

ftp nome_host
stabilire la connessione all'host, poi verra' chiesta la password dell'utente. Se avete sbagliato utente: user .

help
Lista dei comandi disponibili.

help [nome_comando]
Cosa fa quel comando.

put
Per caricare un file.

get
Per scaricare un file.

ls
Lista dei file disponibili.

cd
Spostarsi in un altra directory.

lcd
Cambio directory in LOCALE.

mput/mget
Per lavorare su file multipli.

prompt

Per uscire dalla modalita' interattiva

(non vi chiede conferma di ogni singola operazione
su ogni singolo file...).

binary
Entra in modalita' trasferimento binario.

asii
Entra in modalita' trasferimento ascii.

bye
Per chiudere la sessione.

11.3 Configurazione iniziale

Il demone di vsftpd e' immediatamente disponibile ma solo in modalita' anonima (si pensi a uno scenario in cui si vuole rendere disponibili dei files tramite FTP) e in *sola lettura*. Per accedere al servizio si usi quini come utente `anonymous` (la passwords in genere e' come consuetudine il propio indirizzo email), la cui *home* directory sara' `/home/ftp/` (`/srv/ftp` in Squeeze):

```
zoo:~# ftp localhost
Connected to localhost.localdomain.
220 (vsFTPd 2.0.7)
Name (localhost:root): anonymous
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
200 PORT command successful. Consider using PASV.
150 Here comes the directory listing.
-rw-r--r--    1 0        0               0 Feb 03 17:17 anoni
226 Directory send OK.
```

11.4 Abilitare gli utenti locali

Per poter modificare le impostazioni iniziali, ad esempio per permettere l'accesso agli utenti del server, si modifichera' il file `/etc/vsftpd.conf`, a seguire le impostazioni fondamentali ed altre interessanti per rendere il server accessibile da utenti di sistema (autenticati tramite la loro password, quindi con PAM) per il tipico utilizzo di web designers che debbano pubblicare le loro pagine web (e non si siano fatti convincere a usare SFTP!):

```
# Allow anonymous FTP? (Beware - allowed by default if you comment this out).
anonymous_enable=NO
# Disabilitiamo l'utente anonimo

# Uncomment this to allow local users to log in.
local_enable=YES
# Accesso garantito agli utenti di sistema

# Uncomment this to enable any form of FTP write command.
write_enable=YES
# Permettiamo agli utenti di caricare documenti nella loro home

# You may fully customise the login banner string:
ftpd_banner=Benvenuti al servizio ftp del sito example.com
```

Per abilitare i cambiamenti si proceda a riavviare il server: `/etc/init.d/vsftpd restart` e si monitorizzi il file di log `tail -f /var/log/vsftpd.log` per controllarne il funzionamento (e anche `/var/log/syslog` nel caso non si riuscisse a far partire correttamente il servizio.

NOTE: Se non riuscite ad ottenere un *directory listing* (`ls`) ottenendo un errore `500 Illegal PORT command? FTP error` abilitare la modalita' passiva col comando ftp `passive`.

11.5 Jail chroot

Si puo' impedire all'utente di spostarsi arbitrariamente per il file system del servere visualizzare il contenuto delle directory, ad esempio la cartella `/etc`, confinandolo in una jail chroot limitata alla sua home directory:

```
# You may restrict local users to their home directories.  See the FAQ for
# the possible risks in this before using chroot_local_user or
# chroot_list_enable below.
chroot_local_user=YES
```

Generalmente un utente di sistema con il solo accesso FTP non dovrebbe avere la possibilita' di poter navigare liberamente per il file system del server, esponendo file di configurazione e quant'altro l'utente potrebbe trarre utili informazioni sul quali software siano installati e di che tipo:

```
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> pwd
257 "/"
ftp> cd /etc/
550 Failed to change directory.
```

11.6 Permessi sul filesystem

Come accennato precedentemente e' opportuno che i filesystems sui quali un utente puo' scrivere o modificare il contenuto non abbiano i privilegi di eseguibilita' e suid, nel nonstro caso vsftpd lavora sull'intera `/home/` directory quindi avremo in `/etc/fstab`:

```
/dev/mapper/store-homes /home   ext3    rw,nosuid,noexec     0      2
```

11.7 Shell dell'utente

Come gia' detto piu' volte le passwords degli utenti viaaggiano in rete in chiaro, ponendo un grave problema di sicurezza. Sara' quindi opportuno disbilitare la shell di questi utenti, tramite il flag `--shell /bin/false` in fase di creazione degli utenti:

```
# adduser --shell /bin/false nome_utente
```

Oppure correggiendo manualmente il file `/etc/passwd` per modificare l'inpostazione della shell dell'utente:

```
nome_utente:x:1001:1001::/var/spool/postfix:/bin/bash
# la riga sopra deve essere trasformata in
nome_utente:x:1001:1001::/var/spool/postfix:/bin/false
```

Sui sistemi DEbian REcenti sara' necessario aggiungere `/bin/false` all'elenco delle shell valide.

`/etc/shells`

```
...
/bin/false
```

11.8 Altre opzioni

xferlog_enable=YES
Verra' tenuto un file di log `/var/log/vsftpd.log` degli upload e download sul server.

hide_ids=YES
Nasconde le userid e groupid mascherandole con `ftp` .

anon_root=/home/ftp
Home directory dell'utente anonimo.

write_enable=YES
Permette agli utenti di eseguire i comandi che possono modificare il filesystem: STOR, DELE, RNFR, RNTO, MKD, RMD, APPE e SITE .

idle_session_timeout=600
Permette agli utenti di restare connessi piu' a lungo, utile per i webdesigners che passano intere giornate connessi al server.

---

1 `kde-core` e' piu' leggero del pacchetto `kde`. Esiste un equivalente `gnome-core` per chi preferisce Gnome, nel caso si potrebbe installare il log-in manager *gdm* al posto di *kdm*.

2 Anche se nato per i sistemi Windows, Samba puo' essere usato anche per montare cartelle sotto GNU/Linux come alternativa a NFS. Per la condivisione di stampanti sarebbe invece opportuno intervenire direttamente su CUPS.