

Servizi di rete passo a passo

Appunti sulla installazione e configurazione dei servizi

Author: Andrea Manni

Copyright: GFDL

Version: 0.6

Questa guida e' dedicata agli studenti delle lezioni di informatica tenute da Andrea nel lab208. Nella parte iniziale sono presenti alcuni richiami alle impostazioni di rete e di installazione del laboratorio 208 (lab208) dove generalmente si tengono le lezioni. Questi parametri non sono interessanti per chiunque si trovasse al di fuori della rete piffa.net .

Indice degli argomenti

1 Configurazione sistema

1.1 Solo per uso interno

1.2 Rete

1.3 Bash completion

1.4 Vim

1.5 VNC

1.6 Lista dei pacchetti di base

1.7 Apt configurazione

1.7.1 sources.list

1.7.2 /etc/apt/apt.conf

2 Squid

2.1 Configurazione: squid.conf

2.1.1 Cache_dir

2.1.2 TAG: maximum_object_size

2.1.3 TAG: cache_mem

2.1.4 TAG: minimum_object_size

2.2 Negoziazione degli accessi al servizio

2.2.1 ACL e http access

2.3 Testare Squid

2.3.1 Client: ~/.wgetrc

2.3.2 Server: access.log

3 Apache

3.1 Pacchetti da installare::

3.2 Configurazione di Apache

3.3 apache.conf

3.4 Installazione di PHP

3.4.1 Test del modulo php

3.4.2 Installazione del supporto per Mysql

- 3.4.3 phpmyadmin
- 3.4.4 Installazione del supporto per Postgresql
- 3.4.5 phppgadmin
- 3.5 Virtual hosts
 - 3.5.1 Gestione DNS
 - 3.5.2 Virtual host
- 3.6 Negoziazione accessi
 - 3.6.1 Limiti su base ip
- 3.7 User Authentication
 - 3.7.1 Definire la cartella
 - 3.7.2 Creazione del database delle passwords
 - 3.7.3 Configurazione di Apache
- 3.8 Cavets
- 4 Domain Name System
 - 4.1 Nomi di dominio
 - 4.2 Tipologie di record
 - 4.3 Utilizzo
 - 4.4 Risoluzione dei nomi di dominio
 - 4.5 Dig
 - 4.6 resolv.conf
 - 4.7 /etc/hosts
 - 4.8 Hostname
- 5 DNSmasq
 - 5.1 Configurazione
 - 5.2 DHCP
 - 5.3 DNS cache
 - 5.4 DHCPd
- 6 Samba
 - 6.1 Pacchetti
 - 6.2 Passwords e autenticazione
 - 6.3 Creazione Utenti
 - 6.4 Creare la condivisione
 - 6.4.1 Sicurezza: permessi di esecuzione sul server
 - 6.5 Configurazione dell'applicativo Samba vero e proprio.
 - 6.6 Testare il Servizio
- 7 Firewall
 - 7.1 Links
 - 7.2 Ipfiler
 - 7.3 Progettazione di un firewall
 - 7.3.1 Collocazione

7.3.2 Policy di default

7.3.3 Hardware

7.4 Percorso dei pacchetti tra tabelle e catene

7.5 Concetti di base

7.5.1 Tabelle, catene, regole

7.5.2 Match

7.5.3 Targets

7.6 Tabella Filter

7.7 Flush automatico per macchine remote

7.8 Gestione regole (rules)

8 NOTE

Generato 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 .

Qui riportati per comodita' degli studenti (e del docente che non sara' **mai piu'** costretto a ripeterli continuamente!). Gli altri lettori potranno tenerli presenti per cercare di comprendere gli esempi nel testo. Ad esempio: quando leggerete `10.10.208.254:3128` saprete che si tratta del nostro *proxy http*, stara' quindi a voi sostituire i dati con gli *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	10.10.208.254
gateway	10.10.208.250 persistente
DNS	10.10.208.254
DNS	10.10.208.250 persistente
proxy http	10.10.208.254:3128

Sul portatile di Andrea, corrispondente all'ip 254, gira un DHCP, proxy http e mirror di Debian (<http://debian.piffa.net>). Se Andrea non e' in aula (o ancora peggio non c'e' il suo portatile Net) gli studenti dovranno darsi un indirizzo ip manualmente e disabilitare il proxy (che pero' e trasparente, quindi fate pure come se non ci fosse ;)). Questo in attesa che si sappia se sara' nuovamente utilizzabile il vecchio server Bender.

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, soprattutto 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 `~/.bashrc`

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

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

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

# Some more alias to avoid making mistakes:
# 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](#)

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 `vim` installando il pacchetto `vimi`:

```
# apt-get install vim

e modificare 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.
if has("autocmd")
  filetype plugin indent on
endif

" 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.
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

```

1.5 VNC

I Virtual Network Computing (o VNC) sono software di controllo remoto e servono per amministrare il proprio computer a distanza o visualizzare la sessione di lavoro di un altro computer sul proprio a scopo didattico. Installando un server VNC sulla propria macchina ed impostando una opportuna password si consente ai client VNC di ricevere una immagine dello schermo ed eventualmente di inviare input di tastiera e mouse al computer server (durante le lezioni questo non e' possibile per gli studenti, solo Andrea esegue i comandi). In pratica si può gestire il computer server da un'altra postazione, come se fosse il proprio computer fisico.

Scaricare il pacchetto `xtightvncviewer` e lo script `guarda.sh` in una posizione (collocazione nel *path* degli utenti, es `echo $PATH` per visualizzare l'attuale path) comoda per gli utenti (in genere `/bin`), rendere eseguibile lo script.

Procedura:

```
su root
cd /bin
wget http://debian.piffa.net/guarda.sh
chmod +x guarda.sh
exit
```

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-il8n-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
- `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$

# Archivio principale debian via http su piffa.net,
# non funziona al difuori dell'aula dei corsi
deb http://debian.piffa.net/debian/ Lenny main
# deb http://debian.piffa.net/debian/ Lenny non-free contrib

# Mirror da kernel.org 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 le cose soggette a 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 filesystem locale contenente i pacchetti:
# 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://10.10.208.254:3128";
```

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.

2 Squid

Squid e' un proxy cache http (ma anche FTP e https) robusto e strutturato, puo' essere usato sia in reti relativamente piccole grazie alla semplicita' di configurazione che in scenari piu' complessi grazie alla possibilita' di gestirne in modo granulare le risorse partendo dalle configurazioni piu' semplici per la semplice *condivisione della navigazione* internet, 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' hosts.

Inoltre svolge la funzione di *anonymizer*:

nasconde i client http alla rete internet: risulta solo il server proxy nei log dei server web frequentati dagli utenti di Squid.

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 un *NAT* od altro, basta il solo Squid. Per altro non servira' neanche un servizio DNS 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 storage. 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 stripe* (livello 0), 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 storage. Questo fino al momento in cui per voi non sia piu' importante *garantire la disponibilita' del servizio* (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 filesystem *normale*) con un RAID mirror o 5.

Altra considerazione: i dati del proxy vengono salvati sul filesystem 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/          ext3,noexec,nosuid,noatime 0 3
```

Ora possiamo impostare la cache direttamente nel file `/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
#      algoritmo
#      path della cache
#      dimensione in MB
#      directory primo livello
#      secondo livello di directory
```

Se si modifica la struttura del filesystem della cache di Squid, ad esempio variando il numero delle directory, puo' essere opportuno rigenerare la struttura della cache di squid (per lo meno se si *aumenta il numero delle directory di primo o secondo livello*). *Tipicamente e' opportuno 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 svuota la vecchia cache e se ne genera 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 storage, 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 you 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 dipenderà 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 storage (soprattutto 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 è relativamente esoso di risorse: avrà bisogno di una macchina con ~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 storage.

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 deframmentazione del filesystem e consumare un numero elevato di *inode* (cosa non piu' importante con ext4 o altri filesystem).

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 accessi 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 (che permetterebbero l'invio di SPAM).

Generalmente non volete che il vostro proxy http venga usato da persone sconosciute le quali sostanzialmente *navigherebbero sotto l'identita' del vostro proxy* (probabilmente per visionare materiali che non vorrebbero fossero ricondotti direttamente a loro, per motivi che sta a voi prendere in considerazione) consumando traffico e 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* (indirizi 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

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 Apache

Apache HTTP Server, o piu' comunemente Apache, e' il nome dato alla piattaforma server Web modulare piu' diffusa (ma anche al gruppo di lavoro open source che ha creato, sviluppato e aggiornato il software server), 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 dà vita al World Wide Web, uno dei servizi piu' utilizzati di Internet.

3.1 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.

3.2 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.

3.3 `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 indefinitivamente 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 settaggi 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 .

3.4 Installazione di PHP

Pacchetti da installare: `php5 php-pear`

3.4.1 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 utclient HTTP, a prova del corretto funzionamento del modulo di PHP e della sua integrazione con il serv 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.

3.4.2 Installazione del supporto per Mysql

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`).

3.4.3 phpmyadmin

L'interfaccia web Phpmyadmin non richiede necessariamente la presenza di un database Mysql locale, puo' infatti essere utilizzata per gestire databases 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/`

3.4.4 Installazione del supporto per Postgresql

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`).

3.4.5 *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 databases remoti (il suo file di configurazione: `/etc/phppgadmin/config.inc.php`). Nel caso si voglia installare localmente MySQL si utilizzi il pacchetto `postgresql` .

Phpmysql 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/`

3.5 Virtual hosts

- <http://www.apacheweek.com/features/vhost>
- <http://www.onlamp.com/pub/a/apache/2004/01/08/apachecheckbk.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.

3.5.1 *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:

Bind (DNS server)

Impostare i campi A nelle proprie zone gestite dal server dns Bind. Ad es: `papo A 212.22.136.248`

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.

Dnsmasq (DNS server)

Utilizzabile a livello locale per fare dei test, utilizzando direttive come: `address=/davide.piffa.net/10.10.208.178`

/etc/hosts

Per prove *strettamente a livello locale* potete impostare i nomi dei vostri virtual server nel file `/etc/hosts` .

```
# 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

;; Query time: 12 msec
;; SERVER: 10.10.208.254#53(10.10.208.254)
;; WHEN: Wed May 6 12:27:08 2009
;; MSG SIZE rcvd: 47
```

La parte interessante e' 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 .

3.5.2 Virtual host

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 eventuali 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.

3.6 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 o necessario limitare gli 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.

3.6.1 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 e' comunque possibile mandare delle richieste `GET` per cercare di mandare in Denial Of Service il webserver.

3.7 User Authentication

A volte conviene negoziare gli accessi ad un'area di un sito tramite autenticazione basata sull'accoppiata *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. Per quanto esistano soluzioni piu' granulari e sofisticate per ottenere questo, *mod-auth* puo' essere sufficiente. E *mod-auth* non richiede l'installazione di software aggiuntivi.

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

3.7.1 Definire la cartella

Decidere quale sara' il *path* della cartella da sottoporre ad autenticazione:(e creiamo la cartella):

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

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

3.7.3 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 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 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`

```
# Questo file viene incluso
# nella configurazione del sito web
# Messaggio visualizzato al prompt per l'autenticazione
AuthName "Area privata soggetta ad autenticazione"
# tipo di autenticazione da usarsi
AuthType Basic
# File generato precedentemente 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!).

3.8 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.

4 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 database distribuito, costituito dai server DNS.

Il nome DNS denota anche il protocollo che regola il funzionamento del servizio, i programmi che lo implementano, i server su cui questi girano, l'insieme di questi server che cooperano per fornire il servizio.

I nomi DNS, o "nomi di dominio", sono una delle caratteristiche piu' visibili di Internet.

C'e' confusione in merito alla definizione dell'acronimo: la S spesso viene interpretata come service, ma la definizione corretta e' system.

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

4.1 Nomi di dominio

Un nome a dominio e' costituito da una serie di stringhe separate da punti, ad esempio it.wikipedia.org. A differenza degli indirizzi IP, dove la parte piu' importante del numero e' la prima partendo da sinistra, in un nome DNS la parte piu' importante e' la prima partendo da destra. Questa e' detta dominio di primo livello (o TLD, Top Level Domain), per esempio .org o .it.

Un dominio di secondo livello consiste in due parti, per esempio wikipedia.org, e cosi' via. Ogni ulteriore elemento specifica un'ulteriore suddivisione. Quando un dominio di secondo livello viene registrato all'assegnatario, questo e' autorizzato a usare i nomi di dominio relativi ai successivi livelli come it.wikipedia.org (dominio di terzo livello) e altri come some.other.stuff.wikipedia.org (dominio di quinto livello) e cosi' via.

4.2 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.

4.3 Utilizzo

I computer vengono identificati in rete grazie agli indirizzi *IP*, questi però non sono comodi per gli utenti come riferimento per i vari server. Ad esempio sarebbe scomodo riferirsi al motore di ricerca Google 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.
```

4.4 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` .

4.5 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 servers* 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.

4.6 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".
```

Si veda anche la pagina man di `resolv.conf`.

Attenzione: se si usa un client DHCP o simile questo file potra' essere riscritto automaticamente in base a quanto ottenuto dal DHCP. Si veda la documentazione del pacchetto `resolvconf`.

4.7 /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 e' un associazione tra un IP e stringhe di testo (anche piu' di una es: `mirror` e `mirror.piffa.net`), un record per riga.

Il problema e' la gestione di questo file: quando gli host cambiano IP si devono aggiornare i records, e c'e' poi il problema di distribuire questo file tra i vari hosts della propria LAN. Un metodo semplice per distribuire questo file e' utilizzare `Dnsmasq`: questo infatti legge e onora il file `hosts` locale e lo *distribuisce* ai clients.

Modificare (riconducendola a un ip interno, cosi' annullandola) la risoluzione di un nome di dominio e' un modo drastico e funzionale per *annullarlo* rendendolo indisponibile alla propria rete locale, ad esempio aggiungendo al file `/etc/hosts`:

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

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

4.8 Hostname

Ogni computer ha un *proprio nome* visualizzabile (e modificabile) con il comando `hostname`.

Per modificare in modo permanente il nome del computer si modifichi il contenuto del file `/etc/hostname`.

Tipicamente si vuole mantenere una correlazione tra il nome dell'host, o meglio la stringa con cui il server si qualifica all'esterno, e il PTR dell'ip. Nel caso di *servizi virtuali* ci sara' un *nome server* principale associato al PTR condiviso. Non e' automatico che un servizio, ad esempio un server di posta, si qualifichi leggendo il contenuto di questo file e magari aggiungendo come suffisso il dominio di cui fa parte l'host: a volte questo parametro puo' essere specificato nel file di configurazione del servizio:

```
* Squid: ``visible_hostname``  
* Postfix: ``myhostname``
```

5 DNSmasq

Dnsmasq puo' svolgere le funzioni di un DNS cache / forwarder e un server DHCP caratterizzato dalla facilita' di configurazione, dalla leggerezza e dalla possibilita' di modificare rapidamente i record DNS serviti alla rete. 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' 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 Pubbico).
- Ben integrato con connssioni PPP (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, soprattutto per testare server SMTP impostando al volo i campi MX per nomi di dominio fittizi.

5.1 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.

5.2 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
```

5.3 DNS cache

Aggiungere al file `/etc/resolv.conf` il nameserver localhost in cima alla lista dei *nameserver* disponibili. Dnsmasq usera' la propria cache e in caso non abbia disponibile il *record DNS* richiesto fara' partire una query al primo DNS:

```
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 dovrete aver bisogno di cambiare i DNS una volta impostati (a meno che non usiate un portatile!).

5.4 DHCPd

Dnsmasq puo' lavorare anche come DHCP server per la vostra LAN.

6 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.

6.1 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
```

6.2 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.

6.3 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 vorrà modificare la password di un utente già esistente si usi:

```
smbpasswd sambo
```

La password sotto Windows verrà modificata sul sistema Windows.

6.4 Creare la condivisione

La condivisione altro non è che una cartella sul server che viene resa disponibile ai client negoziando l'accesso in base a una autenticazione basata su *user-name / password*. È per altro possibile permettere l'accesso a una risorsa a chiunque indiscriminatamente (a tutti i *guest*) ma la cosa è 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 più 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/
```

6.4.1 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'è 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 possibilità di eseguire programmi con credenziali diverse.

6.5 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 può 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` .

6.6 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 condivisione 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
```

7 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 sottoreti: 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 sottorete 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 è 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 realtà 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 funzionalità principale in sostanza è 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 capacità 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.

7.1 Links

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

7.2 Ipfiler

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 *inestazioni*) facendo abbondante 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.

7.3 Progettazione di un firewall

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

7.3.1 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>

7.3.2 Policy di default

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

7.3.3 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, flessibilità, strumenti di gestione, sicurezza, OpenBSD.

7.4 Percorso dei pacchetti tra tabelle e catene

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

7.5 Concetti di base

7.5.1 Tabelle, catene, regole

Iptables lavora su 3 tabelle (tables) di default:

- filter - Regola il firewalling: quali pacchetti accettare, quali bloccare
- nat - Regola le attività 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 è 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 ...)

7.5.2 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 è 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 sottorete)

-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

7.5.3 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 è previsto per fare un transparent proxy (con proxy server in esecuzione sulla macchina con iptables)
-j RETURN	Interrompe l'attraversamento della catena. Se questa è una secondaria, il pacchetto torna ad attraversare la catena madre da punto in cui aveva fatto il salto nella secondaria. Se il RETURN è 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 è come se facesse da specchio per tutti i pacchetti ricevuti. Da usare con cautela, per evitare attacchi DOS indiretti.

7.6 Tabella Filter

E' quella implicita e predefinita (-t filter) Riguarda le attività 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
Analogamente: 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

7.7 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]
```

7.8 Gestione regole (rules)

Il comando `iptables` viene usato per ogni attività 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

8 NOTE

- controllare apache

sintassi: in `monospace` :

- nomi di files
- comandi
- pacchetti

-
- 1 `kde-core` e' piu' leggero del pacchetto `kde`, esiste anche un equivalente `gnome-core` `gnome` e il log-in manager `gdm` per il l'ambiente grafico `Gnome`.
 - 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`.