linuxDidattica: la Rete

Umberto Zanatta umbertoz@inwind.it

2003.03.18

Umberto Zanatta

Via Pasubio, 22/3 - I-31022 Preganziol (TV) - umbertoz@inwind.it

Internet

- File PDF: <*http://linuxdidattica.org/docs/linuxdidattica/ld.pdf*>
- File PS: <*http://linuxdidattica.org/docs/linuxdidattica/ld.ps*>

It’s so simple, a stranger’s just a stranger that you’ve not met
Just talk it through and find out if you have yet
It’s too easy to find out if there’s something here to live for
I’m unavailable to see what’ll happen to me

"Let It Ride" - Shed Seven (1998)

Indice generale

Premessa

Ho approfondito la conoscenza di GNU/Linux in molti anni di lavoro, cercando di esplorare sempre le sue potenzialità e condividendo in modo completo la filosofia di distribuzione basata sulla licenza GPL. Da molti anni lo uso regolarmente a casa e al lavoro, applicando parte del mio tempo allo studio degli applicativi, dei servizi e dei comandi.

Ciò che viene esposto in questi appunti richiede un po' di conoscenze base su Unix, come l'utilizzo di un editor (tipo Vi: "il tool di amministrazione universale") e conoscenze di informatica generale.

Questa dispensa guida l'utente passo passo su come costruire una Intranet e come amministrare un server GNU/Linux. Per l'uso approfondito dei comandi e dei servizi (demoni) fare sempre riferimento alla documentazione dei pacchetti e naturalmente ad «Appunti di informatica libera» di Daniele Giacomini.

La parte preliminare della costruzione di una rete interna partendo dal file '`hosts`' viene tralasciata. Tutti gli esempi ivi riportati richiedeno un GNU/Linux installato nel vostro computer con kernel 2.2.x meglio un kernel 2.4.x.

Parte dei contenuti sono stati raccolti dalla documentazione standard, taluni invece fanno parte di informazioni raccolte da alcuni articoli altri dall'esperienza acquisita utilizzando server e workstation in GNU/Linux.

L'uso di questa dispensa è rivolto ad insegnanti e studenti con limitata esperienza in ambiente GNU/Linux; essa è quindi utile per la realizzazione di una rete scolastica.

Sono ben accettate critiche, commenti, suggerimenti e magari approfondimenti che eventualmente potranno essere inseriti in questi appunti. Se non violano i termini della licenza.

Buon lavoro.

"ELEPHANT MAN" - THE DANDYS (1997)

Umberto Zanatta

Capitolo 1

Costruzione di una LAN

> - Costruire una LAN?... Se dovessi seguire manuali ed esperti di reti sarei ancora a far conti!... Se vengono i carabinieri ci mettono dentro!

Così mi rispose un professore quando mi mostrò la sua rete scolastica, lontanta, ve lo posso garantire, da ogni standard qualunque esso sia. Con il tempo la rete si è trasformata, dal coassiale è passata al RJ-45; sono stati comperati degli HUB e centralizzati alcuni servizi cardine.

> - Architettura Server-Client?... 386 come server e 286 come client!

Generalmente «fare la rete» o «costruire la rete» può comportare molte spese in hardware dedicato ed in software proprietario, per questo un preventivo di costi è consigliabile prima di acquistare apparecchiature apparentemente utili, ma con il tempo sottoutilizzate od accantonate definitivamente.

Le esigenze di una rete con pochi client possono essere soddisfatte con un unico router; magari con una o più connessione ISDN o ADSL. L'utilizzo di una cache gestita da un proxy server diminuisce drasticamente il consumo di banda.

Se aumentano i client ed i servizi installati l'amministratore sarà vincolato nella scelta di distribuire su più server i servizi fondamentali: Web, Posta, Database.

Gestire molti server vuol dire investire più tempo uomo in manutenzione ed aggiornamento software (non sottovalutate per nulla questo aspetto).

Allora centralizzare servizi su uno stesso server è la soluzione migliore?

- Indubbiamente! - Ricordiamo che stiamo discutendo soluzioni didattiche e per la didattica, quindi non è richiesta elevata potenza di calcolo nè spazio su disco per contenere ingenti quantità di dati.

In una macchina Pentium con 64 o 128 Mb di RAM possono coesistere contemporaneamente server Web, server di posta Sendmail, server FTP, un server SQL e un Firewall.

Se la memoria RAM supera i 256 Mb si può installare anche un server Proxy per lo stoccaggio delle pagine Web.

Il nodo così configurato potrebbe inoltre fungere anche da instradatore per sottoreti della Intranet e/o verso Internet con la tecnica del mascheramento degli indirizzi dopo averlo opportunamente protetto la sottorete interna.

- Quale indirizzo per il mio network? Classe A, B o C? -

Generalmente consiglio gli indirizzi in classe B (172.16.0.0) dove è possibile costruire sottoreti con un'unica scheda di rete con uno o piu concentratori (HUB) e senza la necessità di configurare un gateway tra le sottoreti per farle comunicare.

Didatticamente è preferibile configurare la rete in classe C, dotando il server di due schede di rete; per lo studente risulta più immediato intuire il concetto fisico di Router o instradatore.

La complessità può comunque aumentare, soprattutto per chi usa portatili diventa probelmatico il punto rete (devo configurare sempre la scheda di rete). Dove mi «attacco»? Su quale rete?

Quindi l'installazione di un servizio che a priori distribuisca gli indirizzi IP ai client senza preoccuparsi di configurare gli stessi. Un servizio così particolare è il DHCP

Utilizzare sempre cavi RJ-45 in classe 5 con concentratori ed abbandonare il coassiale, fonte di innumerevoli problemi, spesso risolvibili «a naso» ed a tentativi per ogni nodo della rete.

Comperare concentratori in standard 10/100; il costo oramai si è notevolmente abbassato e nonostante non ci sia un vero guadagno in termini di banda (un'aula didattica generalmente non fa uso massiccio della rete) non necessita di aggiornamenti tecnologici nel futuro.

Per questo rifiutare le schede di rete NE2000 compatibili! xsComperare marche conosciute ed affidabili. Certo i Personal Computer che fungeranno da server, si definiscono tali per i servizi e non per l'hardware con cui sono costruiti; per questo non spendere in costose schede dedicate, bensì in normali schede di rete per Personal Computer da ufficio.

Capitolo 2

Protezione della rete interna: i Firewall

2.1 Filtraggio dei pacchetti TCP/IP

Un'applicazione che filtra i pacchetti TCP/IP generalmente controlla l'intestazione (o, in inglese, *header*) del pacchetto ed in base al suo contenuto decide che farsene. Può accettarlo (ACCEPT) o rifiutarlo (DROP).

Con il sistema operativo GNU/Linux il sistema di filtraggio dei pacchetti è costruito direttamente nel kernel.

Ogni amministratore di Firewall è pienamente responsabile delle politiche di controllo del traffico. Per questo dovrà stabilire a priori quali pacchetti sono ammessi in entrata e quali pacchetti garantisce in uscita. Siano essi pacchetti destinati verso alcuni siti, sia verso alcune porte. In quest'ultima ipotesi potrei disabilitare l'uscita verso la porta 80, e costringere l'utilizzo di un Proxy Server.

2.2 Linux e NetFilter

Il filtraggio dei pacchetti in Linux fu introdotto dalla versione 2.0 del kernel, via via il sistema di filtraggio è stato modificato e spesso completamente riscritto. Il primo pacchetto di amministrazione fu IpFwadm, sostituito con il kernel 2.2 da IpChains per ritrovarsi, oggi, con IpTables.

Tutti questi pacchetti girano in spazio utente (*user space*) e comunicano con il kernel di GNU/Linux.

2.2.1 NetFilter e i pacchetti

I pacchetti, una volta entrati nel Firewall, fanno un giro determinato all'interno del kernel prima di raggiungere il destinatario. Questi passi sono regolati da delle *Chain* dove il pacchetto viene esaminato con più regole che determinano la sorte dello stesso. Queste regole agiscono per un insieme ben determinato di tabelle.

Esse vengono individuate con il nome di '**`mangle`**', '**`nat`**' e '**`filter`**'.

2.2.2 Tabella Mangle

La tabella agisce sul mangling dei pacchetti. Viene comunemente usata per modificare gli identificatori di TOS, TTL e MARK.

Il TOS (*Type Of Service*) decide la policy del pacchetto, tipo come esso viene reindirizzato. Generalmente non c'è bisogno di modificare questo comportamento a meno che uno non abbia le idee chiare in proposito.

Il TTL (*Time To Live*) del pacchetto è impostato con questo campo.

Il campo MARK viene usato per impostare il valore del campo *mark*. I mark hanno utilizzi per lo più rivolti al limitare la banda e per questo sono usati con il pacchetto '`iproute2`'.

2.2.3 Tabella NAT

La tabella NAT (*Network Address Translation*) viene adoperata per modificare i pacchetti sorgenti o quelli di destinazione. Sono tre i tipi di NAT gestiti: DNAT (*Destination Network Address Translation*), SNAT (*Source Network Address Translation*) e MASQUERADE.

Il primo metodo viene utilizzato per indirizzare l'indirizzo di destinazione verso un altro nodo, appartenente quasi sempre ad una zona nascosta e protetta dal Firewall.

Il secondo viene utilizzato per modificare l'indirizzo Ip dei pacchetti sorgenti. Questo sistema è impiegato per proteggere la rete interna locale dagli intrusi lasciando visibile solo il numero Ip statico del Firewall da dove escono tutti i pacchetti; è compito del Firewall reindirizzarli, una volta ricevuti, al nodo a cui sono diretti.

Il terzo ed ultimo metodo si comporta in modo equivalente al secondo, tuttavia è più oneroso di risorse. Esso controlla, ogni volta che manipola un pacchetto, l'indirzzo Ip della scheda di rete. Trova utilizzo quando si dispone di un Ip dinamico come una connessione con protocollo PPP.

2.2.4 Tabella Filter

La tabella viene adoperata per controllare i pacchetti con una serie di filtri preconfezionati. Con essa si manipolano tre catene dette *chain* per i pacchetti in entrata (chain di INPUT), in uscita (chain di OUTPUT) oppure destinati ad un'altra rete (chain di FORWARD). Una volta controllati i pacchetti e confrontati con i filtri, viene decisa la loro sorte.

Quando un qualsiasi pacchetto raggiunge una delle tre chain, quest'ultime decide che farsene dello stesso. Possono acettarlo (ACCEPT), oppure rifiutarlo o addirittura eliminarlo (DROP).

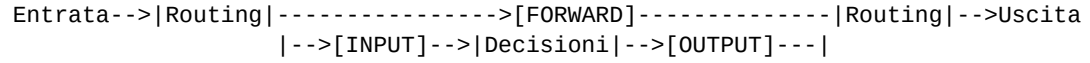
Nel dettaglio una chain, è formata da una lista di regole. Queste regole vengono confrontate con l'intestazione del pacchetto. Se il pacchetto passa la prima regola, viene confrontato con la seconda e così via. L'ultimo confronto viene fatto con la chain detta di *policy*, che per motivi di sicurezza dovrebbe sempre rifiutare il pacchetto (DROP).

Riassumendo:

1. Un pacchetto arriva attraverso un dispostivio di rete e il kernel individua la sua destinazione: routing.
2. Se il pacchetto è destinato alla rete interna, passa alla chain di '**`INPUT`**'.
3. Se il kernel non conosce la destinazione del pacchetto o, non ha il Forwarding abilitato, rifiuta il pacchetto.
4. Superata la '**`INPUT`**' chain, il pacchetto passa immediatamente alla '**`OUTPUT`**' chain, fino a destinazione se supera le regole.

In figura 2.1 lo scheletro dei passi necessari.

Figura 2.1. Regole di attraversamento.



In tabella 2.1 il riassunto sul funzionamento delle tre tabelle.

Tabella 2.1. Tabelle di NetFilter.

Tabella	Descrizione
nat	Tabella utilizzata per il NAT.
mangle	Tabella utilizzata per il mangling dei pacchetti.
filter	Tabella utilizzata per il filtraggio generale dei pacchetti.

2.3 NetFilter con IpTables

L'utility '`iptables`' aggiunge e rimuove regole dalla tabella di filtraggio dei pacchetti. Ad ogni riavvio del computer, questa viene interamente persa. Se alcuni componenti di NetFilter sono compilati come modulo, essi fanno riferimento ad una forma '`iptable_<nome_tabella>.o`' per quanto concerne le tabelle o '`ipt_<DESTINAZIONE>`' per quanto concerne i meccanismi di destinazione del pacchetto.

IpTables accetta la sintassi seguente:

`iptables` [*-t table*] *command* [*match*] [*target/jump*]

Non è necessario specificare la tabella in quanto in mancanza di questa IpTables utilizza quella di filter. Il mio personale consiglio è di iniziare ogni regola con l'opzione '**`-t`**'; pur se rindondante, non crea confusione inutile.

Il campo obbligatorio '**`command`**' agisce direttamente sulle regole: aggiunta, cancellazione o modifca. Il campo '**`match`**' associa il percorso di un pacchetto con un dispositivo come la scheda di rete. Il campo '**`target`**' è la decisione finale presa sul pacchetto.

2.3.1 Operare sulle regole e sulle chain

IpTables numera le regole in ordine crescente partendo dalla regola numero 1.

I comandi di IpTables che agiscono sulle chain sono i seguenti:

`-A | --append`
: Aggiunge una nuova regola in coda alle precedenti.

`-I | --insert`
: Inserisce una nuova regola in qualsiasi posizione.

`-R | --replace`
: Sostituisce una regola precedente con una nuova.

`-D | --delete`
: Elimina una regola.

`-L | --list`
: Visualizza le informazioni sulle chain.

`-N | --new-chain`
: Crea una nuova chain.

`-X | --delete-chain`
: Elimina una chain solo se vuota.

`-P | --policy`
: Cambia la policy per una chain.

`-F | --flush`
: Elimina tutte le regole sulle chain.

-Z | --zero

Azzera i byte e i contatori in tutte le chain.

-E | --rename-chain

Cambia nome ad una chain.

Alcuni di questi comandi sono seguiti da opzioni:

-v | -verbose da utilizzare con **-L, -A, -I, -D, -R**

Visualizza informazioni dettagliate sul comando eseguito.

-x, -n da utilizzare con **-L**

La prima opzione visualizza le informazioni in byte, la seconda informazioni dettagliate in modo numerico.

--line-numbers da utilizzare con **-L**

Visualizza le regole con il rispettivo numero di riga.

-c | --set-counters PKTS BYTES da utilizzare con **-I, -A, -R**

Imposta il valore iniziale del contatore dei pacchetti e dei byte quando si crea una regola.

--modprobe = command

Il comando da utilizzare per caricare i moduli quando si manipola una chain.

All'avvio del server, la tabella dei filtri gestita da NetFilter è vuota e tutte le policy delle chain sono impostate su ACCEPT.

In molte distribuzioni GNU/Linux le policy di NetFilter sono spesso gestite da uno script di avvio. Fare riferimento, quindi, ai manuali di installazione o ai file di avvio per verificare lo stato delle chain.

2.3.2 Estensioni a IpTables: Le Match

Le regole possono agire su cinque categorie differenti. Le prime tre sono rappresentate dai protocolli TCP, UDP e ICMP. La quarta è generica e, la regola, agisce in modo globale. La quinta, più specifica agisce sul contenuto stesso del pacchetto per identificarne per esempio lo stato.

In lingua inglese questa azione viene identificata come *match*. Per semplificare la lettura adopereremo questo termine.

La match generica agisce su ogni tipo di protocollo che il kernel è capace di controllare. Non è richiesto nessun comando, o opzione per controllare la match.

-s | --src | --source

Questa match viene usata per individuare i pacchetti in base all' indirizzo IP sorgente. L'indirzzo IP può essere assegnato in più modi, quelli comuni sono o il numero IP o il nome per esteso del nodo.

Gli inidirzzi IP vengono raggrupptati con una maschera tipo: 172.17.0.0/16 oppure 172.17.0.0/255.255.0.0.

-d | --dst | --destination

Come per il precedente, tuttavia la regola agisce sull'indirizzo IP di destinazione.

-p | --protocol

Questa match individua il protocollo o una lista di protocolli separata dal carattere "," da controllare nella regola.

Il protocollo viene confrontato con il numero, se conosciuto; oppure con i casi comuni tcp, udp o icmp. Non ha importanza lo stile maiuscolo.

`-i | --in-interface`

Questa macth impone alla regola di controllare solo i pacchetti in entrata per l'interfaccia indicata.

Il carattere jolly "+" identifica le interfacce con la stringa che lo precede ('**`eth+`**' equivale a tutte le schede di rete).

I pacchetti che attraversano la chain di INPUT non rispondono all'opzione '**`-o`**', mentre quelli che attraversano la chain di OUTPUT non rispondono all'opzione '**`-i`**'. I pacchetti verranno ignorati. La chain di FORWARD accetta entrambi i casi.

Questa match è quindi valida solo per le chain di INPUT, FORWARD e PREROUTING.

Non è indispensabile che l'interfaccia sia al momento attiva, tuttavia le regole nelle chain si attiveranno nel momento in cui lo sarà.

`-o | --out-interface`

Questa match impone alla regola di controllare solo i pacchetti in uscita per l'interfaccia indicata.

Questa match è quindi valida solo per le chain di OUTPUT, FORWARD e POSTROUTING.

Non è indispensabile che l'interfaccia sia al momento attiva, tuttavia le regole nelle chain si attiveranno nel momento in cui lo sarà.

`-f | --fragment`

I pacchetti troppo lunghi non sono spediti in blocco, il protocollo TCP/IP si incarica di suddividerli in più frammenti, i quali, giunti a destinazione, vengono ricostruiti.

Solo il frammento iniziale porta con esso l'intestazione (*header*) del pacchetto, gli altri solo un subset di quest'ultima. Per questo motivo i pacchetti vengono ricomposti prima di raggiungere lo strato gestito da NetFilter. NetFilter, tuttavia, non controlla tutti il blocco dei frammenti ma, solamente, il primo.

Una regola per evitare che qualsiasi frammento raggiunga un nodo può venire così scritta:

`iptables -A OUTPUT -f -d 172.17.1.254 -j DROP` [*Invio*]

`!`

Le regole nelle match possono venire anticipate dal carattere "!".

La sua funzione è di negare la regola. Per esempio: '**`-s ! 172.17.0.0/16`**' individua solo i pacchetti che non provengono dalla sottorete specificata.

2.3.3 Le Match implicite

Le capcità elaborative di IpTables possono essere potenziate aggiungendo nuove caratteristiche in forma di estensioni.

Alcune di queste sono standard, altre vengono create e distribuite dai programmatori. Le estensioni vengono fornite come librerie condivise, che il programma carica a runtime. Allo scopo è consigliabile consultare la documentazione della propria distribuzione per verificare la directory dove risiedono.

Le Match per il protocollo TCP

Le estensioni del protocollo TCP sono caricate a runtime se viene specificata l'opzione '**`-p tcp`**'.

`--tcp-flags`

Filtra un flag specifico del TCP. La prima stringa passsata è la maschera di bit, la seconda verifica quali di questi bit sono impostati a 1. La lista dei bit viene separata dal carattere "," e quelli riconosciuti sono **SYN**, **ACK**, **FIN**, **RST**, **URG**, **PSH** e naturalmente **ALL** e **NONE**.

Può essere invertita la regola con il carattere '**`!`**'.

`--syn`

Equivale a '**`--tcp-flags SYN,ACK,FIN SYN`**' ossia controlla i pacchetti che hanno impostato il bit di SYN ma non quelli di ACK e FIN.

`--sport | --source-port`

Accetta come argomento una porta singola o un insieme, range, di porte sia in codice numerico (preferibile per la velocità) che mnemonico. Questa match viene usato per controllare i pacchetti in base alla porta di origine.

Un insieme di porta viene separato dal carattere ":" (tipo: '**`--sport 50:80`**'), mentre sono accettate le forme "fino alla porta" (tipo: '**`--sport :50`**') e "dalla porta" (tipo: '**`--sport 50:`**'). Puograve; essere invertita la regola con il carattere !.

`--dport | --destination-port`

Come '**`--sport`**', ma la porta specificata è quella di destinazione.

`--tcp-option`

Confronta un pacchetto TCP con una opzione equivalente al numero passato come argomento. Un pacchetto che non presenta una intestazione corretta viene scartato immediatamente. Può essere invertita la regola con il carattere !.

Le Match per il protocollo UDP

Le estensioni del protocollo UDP sono caricate a runtime se viene specificata l'opzione '**`-p udp`**'. Le opzioni che seguono vengono accettate.

`--sport | --source-port`

è equivalente al funzionamento della match per il protocollo TCP.

`--dport | --destination-port`

è equivalente al funzionamento della match per il protocollo TCP.

Le Match per il protocollo ICMP

Le estensioni del protocollo ICMP sono caricate a runtime se viene specificata l'opzione '**`-p icmp`**'. Le opzioni che seguono vengono accettate.

`--icmp-type`

Accetta come argomento un tipo numerico, l'equivalente mnemonico o numerico e codice separati dal carattere "/". Può essere invertita la regola con il carattere !.

2.3.4 Le Match esplicite

Le match esplicite vengono utilizzate solamente quando ne viene fatta richiesta con l'opzione '**`-m | --match`**'. A differenza delle match implicite non vengono caricate in modo automatico.

MAC match

Sono caricate con l'opzione '**`-m mac`**'.

`--mac-source`

Confronta i pacchetti con il numero MAC dell'indirizzo sorgente. La forma dell'indirizzo è composta da sei numeri esadecimali così composti XX:XX:XX:XX:XX:XX. La regola può essere inveritita con il carattere "!".

Limit match

Sono caricate con l'opzione '**`-m limit`**'.

`--limit`

Limita il numero di attività svolte su una data catena; svolge quindi una attività di controllo nel caso venga superato un limite medio imposto. La match viene seguita da un numero e da un identificatore orario opzionale. Questi identificatori sono: '**`/second /minute /hour /day`**'. Il valore predefinito è '**`3/hour`**'. La regola può essere invertita con il carattere "!" quindi i pacchetti verranno confrontati solo quando supereranno il limte.

`--limit-burst`

Limita il numero massimo di pacchetti iniziali che IpTables deve gestire. Se il numero specificato con l'opzione '**`-limit`**' non viene raggiunto allora viene aggiunta una quantità pari ad uno fino al limite prefissato. Il valore predefinito è cinque. La regola può essere invertita con il carattere "!".

2.3.5 Le Match multiporta

Le match multiporta sono utilizzate per assegnare un gruppo di porte di destinazione invece di una sola.

Sono caricate con l'opzione -m multiport.

`--source-port`

Accetta un numero massimo di 15 porte in ingresso separate da una virgola e solo per i protocolli '**`tcp`**' e '**`udp`**'.

`--destination-port`

Come la precedente per le porte di destinazione.

`--port`

Come le precedenti tuttavia controlla sia le porte in ingresso che in uscita.

2.3.6 Le Match mark

Sono utilizzate per confrontare i pacchetti in base al marcatore che li accompagna. Questi marcatori sono considerati campi speciali assegnati e mantenuti solamente dal kernel durante l'attraversamento del Firewall.

Sono caricate con l'opzione '**`-m marck`**' e lavorano sulla tabella di '**`mangle`**'.

`--mark`

Il valore da confrontare è un numero intero senza segno che può avere un massimo di 65535 valori. In questo modo si identifica il pacchetto con quello particolare marcatore.

2.3.7 Le Match owner

Sono utilizzate per confrontare i pacchetti in base all'utente che li ha generati. Funzionano solamente con la catena di '**`OUTPUT`**'.

Sono caricate con l'opzione '**`-m owner`**'.

`--uid-owner`

Confronta il pacchetto con la UID dell'utente.

`--gid-owner`

Confronta il pacchetto con la GID del gruppo.

--pid-owner

Confronta il pacchetto con il PID del processo. La configurazione di tale regola può essere ardua e necessita di uno script costruito appositamente.

--sid-owner

Confronta il pacchetto con il SID della sessione. Questo è unico per tutti i processi figli creati da un processo padre.

2.3.8 Le Match state

Sono utilizzate unitamente con la traccia della connessione individuata dal kernel e garantiscono l'accesso a questa informazione. Ogni connessione viene organizzata in un database e superato un certo tempo viene eliminata da quest'ultimo.

Sono caricate con l'opzione '**-m state**'.

--state

Indica lo stato che deve essere confrontato della connessione: '**INVALID**', '**ESTABLISHED**', '**NEW**' e '**RELATED**'. Lo stato '**INVALID**' riguarda quei pacchetti che non sono associati a nessuna connessione e contengono dati non corretti. Lo stato '**ESTABLISHED**' riguarda quei pacchetti associati ad una connessione già effettuata in entrambe le direzioni di comunicazione. Lo stato '**NEW**' riguarda quei pacchetti associati ad un'apertura di una nuova connessione non ancora avvenuta in entrambe le direzioni. Lo stato '**RELATED**' riguarda quei pacchetti associati ad un'apertura di una nuova connessione associata ad una connessione già stabilita.

2.3.9 Impostare l'obiettivo

Una volta che NetFilter ha esaminato un pacchetto Internet deve fare alcune considerazioni sulla destinazione di quest'ultimo. Questa regola rappresenta il target.

IpTables è fornito di due target interni: '**DROP**' e '**ACCEPT**'. Altri tipi di target sono le estensioni e target definiti dagli utenti come, per esempio, '**REJECT**'.

Essi sono l'unione di un modulo kernel per NetFilter e un nuovo comando, visto come estensione, per IpTables. Ogni aggiornamento del kernel porta con sè nuove estensioni. Alcune di queste sono: '**REJECT**', '**MIRROR**', '**LOG**', '**ULOG**', '**REDIRECT**'.

Quando un pacchetto incontra una regola con target definito dall'utente, il pacchetto inizia il confronto con le regole nella catena utente. Se quest'ultima ignora il pacchetto esso riattraversa la catena originaria.

Gli obiettivi sono impostati con l'opzione '**-j**'.

ACCEPT

Il pacchetto viene accettato e non attraversa più le regole definite nella catena di attraversamento; tuttavia potrebbe essere interrotto da regole in altre catene o tabelle.

DROP

Il pacchetto viene rilasciato e non viene più processato dal firewall. Generalmente può lasciare dei socket inutilizzati nel cliente o nel servente per cui si preferisce utilizzare al suo posto '**REJECT**'.

REJECT

Il funzionamento è simile a '**DROP**', tuttavia invia un pacchetto Icmp come errore all'indirizzo sorgente. Viene impiegato solamente per le catente di '**INPUT**', '**FORWARD**' e '**OUTPUT**'.

`--reject-with`

Inidica come il firewall invia il segnale di rifiuto al computer remoto. Attualmente vi sono i seguenti tipi di segnale: '**`icmp-net-unreachable`**', '**`icmp-host-unreachable`**', '**`icmp-port-unreachable`**', '**`icmp-proto-unreachable`**', '**`icmp-net-prohibited`**' e '**`icmp-host-prohibited`**'. Il segnale di errore predefinito è '**`icmp-port-unreachable`**'. Il segnale '**`tcp-reset`**' viene usato in congiunzione al solo protocollo Tcp ed invia un pacchetto di tipo '**`TCP RST`**' che chiude la connessione in modo corretto. Il segnale '**`echo-reply`**' viene usato in congiunzione con il protocollo Icmp quando si intercettano pacchetti di ping.

SNAT

Lo SNAT (*Source network address translation*) viene utilizzato per riscrivere l'indirizzo Ip sorgente nell'intestazione del pacchetto Tcp/Ip. L'utilizzo principale si ha quando più computer condividono la stessa connessione Internet con un unico Ip statico assegnato al Firewall/Router. Senza questa traduzione il nodo remoto non potrebbe spedire i pacchetti Tcp/Ip verso la LAN che utilizza, generalmente, indirizzi Ip privati e non pubblici.

Lo SNAT è impiegato solo con la tabella di '**`nat`**' e con la catena di '**`POSTROUTING`**'.

`--to-source`

Indica il numero Ip che si vuole utilizzare come indirizzo sorgente. Più indirizzi Ip, separati dal carattere "-" possono essere impiegati per bilanciare il carico.

MASQUERADE

Il meccanismo di funzionamento è identico a quello di SNAT tuttavia non richiede di specificare un numero Ip sorgente in quanto questo viene calcolato ogni volta. L'utilizzo principale si ha quando più condividono la stessa connessione Internet con un unico Ip dinamico assegnato al Firewall/Router.

LOG

Le informazioni relative al pacchetto vengono registrate in un file di registro. Questo comportamento è particolarmente indicato per i pacchetti accompagnati da una «matrice» illegale destinati ad attaccare l'integrità del sistema. Attraverso '**`dmesg`**' o '**`syslogd`**' è possibile leggere le informazioni registrate.

`--log-level`

Indica qual è il livello di Syslog da utilizzare; esso corrisponde ai canonici: debug, info, notice, warning, err, crit, alert e emerg.

`--log-prefix`

Aggiunge un prefisso ai messaggi inseriti nel registro. La lunghezza del testo è limitata a 29 caratteri qualsiasi.

`--log-tcp-sequence`

Registra la sequenza del protocollo tcp con il messaggio vero e proprio. La sequenza del protocollo tcp è un numero speciale che identifica ogni pacchetto e il modo con cui esso verrà riassemblato. Per ragioni di sicurezza è preferibile che il log non sia leggibile agli utenti.

`--log-tcp-options`

Registra le differenti opzioni dall'intestazione del pacchetto tcp. Può essere necessario per valutare cosa, nella configurazione, è andato «storto».

`--log-ip-options`

Come il precedente, tuttavia registra le informazioni dei pacchetti ip.

ULOG

Consente di registrare le informazioni in modo più complesso e articolato di quanto consentito con '**`LOG`**'. La gestione avviene nello spazio utente via il dispositivo netlink grazie ad un socket multicast. In questo modo qualsiasi processo utente può intercettare le informazioni sui pacchetti e registrarle a piacimento. Ulogd è un demone che può immagazzinare le attività di NetFilter anche su database PostgreSql o Mysql.

`--ulog-nlgroup`

Indica qual'è il gruppo netlink a cui spedire i pacchetti. Vi sono in totale 32 gruppi, quello preimpostato è il gruppo 1.

`--ulog-prefix`

Aggiunge un prefisso ai messaggi inseriti nel registro. La lunghezza del testo è limitata a 32 caratteri qualsiasi.

`--ulog-cprange`

Indica quanti byte del pacchetto da spedire in multicast ai demoni gestiti nello spazio utente. Il valore preimpostato è 0 che indica l'intera informazione sul pacchetto.

`--ulog-qthreshold`

Gestisce una cache di pacchetti prima che essi siano inviati in multicast ai demoni gestiti nello spazio utente. Il valore preimpostato è 1.

2.3.10 Ulogd

Sintassi

`ulogd` [*opzioni*]

Ulogd è un demone che legge la coda dei messaggi generati da IpTables se quest'ultimo ha generato messaggi diretti verso '**`ULOG`**'.

Ulogd genera un file di registro testuale o avanzati con l'utilizzo di server SQL tipo MySQL o PostgreSQL.

Alcune opzioni

Opzione	Descrizione
-d	Avvia ulogd come servizio
-c *nome_file*	Utilizza *nome_file* come file di configurazione
-h	Visualizza l'aiuto contestuale
-v	Visualizza il numero di versione

2.3.11 '`/etc/ulogd.conf`'

Tutti i parametri di Ulogd possono essere modificati sia da IpTables che dal file di configurazione '`/etc/ulogd.conf`'.

Parametro	Descrizione
nlgroup	Il gruppo multicast da cui il demone '**`ulogd`**' prende le informazioni
logfile	Il file di registro principale, dove '**`ulogd`**' riporta le attività inerenti al demone
loglevel	Modifica il grado di accuratezza dei messaggi inviati al file di registro. I livelli predefiniti sono: 1=debug information, 3=informational messages, 5=noticable exceptional conditions, 7=error conditions, 8=fatal errors.
plugin	Carica un modulo aggiuntivo all'avvio, come il gestore di MySQL

Un esempio

```
# file /etc/ulogd.conf

# necessario
plugin /usr/lib/ulogd/ulogd_BASE.so
nlgroup 1
logfile /var/log/ulog/ulogd.log
loglevel 5

# emulazione del demone syslogd
plugin /usr/lib/ulogd/ulogd_LOGEMU.so
syslogsync 1
syslogfile /var/log/ulog/syslogemu.log

# registra le attivita' di ULOG
plugin /usr/lib/ulogd/ulogd_OPRINT.so
dumpfile /var/log/ulog/pktlog
```

2.3.12 Creare una nuova catena

Passo passo arriviamo alla costruzione di una catena che chiameremo per comodità test.

iptables -N test [*Invio*]

L'opzione '**-N**' crea la nuova catena, è equivalente a '**--new-chain**'. Ogni nuova catena creata conterrà delle regole e degli obiettivi.

Per visualizzare le regole presenti nella catena si usa l'opzione '**-L**'.

iptables -L test [*Invio*]

La catena, appena creata è vuota.

```
Chain test (0 references)
target     prot opt source               destination
```

Per avere informazioni estese aggiungere in coda l'opzione '**-v**'. Per evitare che IpTables interroghi un server Dns per risolvere gli indirizzi Ip aggiungere l'opzione '**-n**'.

La catena viene elimnata con l'opzione '**-X**', mentre le regole vengono svuotate con l'opzione '**-F**'.

iptables -X test [*Invio*]

2.3.13 IpTables esempi pratici

Le opzioni comuni che si effettuano sulle chain sono di aggiunta ('**-A**') o di eliminazione delle regole ('**-D**').

Ogni regola impone di specificare la condizione e l'obiettivo da svolger se i pacchetti la passano. Per esempio se si desidera disabilitare le risposte al comando '`ping`', la condizione va configurata per il protocollo Icmp, mentre l'obiettivo si imposta in rifiuto del pacchetto ('**DROP**').

iptables -F [*Invio*]

Elimina tutte le regole impostate all'avvio del computer da eventuali file '`rc`' di script nella catena di '**filter**'.

iptables -t nat -F [*Invio*]

Elimina tutte le regole impostate all'avvio del computer da eventuali file '`rc`' di script nella catena di '**nat**'.

iptables -X [*Invio*]

Elimina tutte le catene definite all'avvio del computer da eventuali file '`rc`' di script nella catena di '**filter**'.

iptables -t filter -P INPUT DROP [*Invio*]

iptables -t filter -P FORWARD DROP [*Invio*]

iptables -t filter -P OUTPUT ACCEPT [*Invio*]

Sono le politiche predefinite per i pacchetti in ingresso, in inoltro ed in uscita nella catena di '**filter**'.

Per un firewall «personale» si preferisce non controllare i pacchetti in uscita liberi.

iptables -A INPUT -s 127.0.0.1 -p icmp -j DROP [*Invio*]

Digitiamo il comando '`ping`' e come indirizzo di destinazione scegliamo l'interfaccia di loopback:

ping 127.0.0.1 [*Invio*]
```
  PING 127.0.0.1 (127.0.0.1) from 127.0.0.1 : 56(84) bytes of data.
```

Il kernel rifiuta (DROP) il protocollo ICMP verso l'interfaccia di loopback.

ssh localhost [*Invio*]

Il sistema non risponde; assicurarsi di avere comunque installato il servente Ssh.

iptables -A INPUT -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT [*Invio*]

iptables -A INPUT -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT [*Invio*]

Il firewall accetta in ingresso una connessione proveniente dalla porta 22 generalmente un servente Ssh e una connessione destinata alla porta 22 verso il servente Ssh locale.

Lo stato della connessione nella prima regola deve essere '**ESTABLISHED**' in quanto l'utente locale ha già creato una connessione stabile richiedendo il servizio Ssh al servente. Lo stato della connessione nella seconda regola deve essere '**NEW,ESTABLISHED**' in quanto l'utente remoto deve aprire una nuova connessione con il servente Ssh locale.

iptables -A INPUT -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT [*Invio*]

iptables -A INPUT -p tcp --dport 80 -m state --state NEW,ESTABLISHED -j ACCEPT [*Invio*]

Il firewall accetta in ingresso una connessione proveniente dall porta 80 generalmente un servente Http e una connessione destinata alla porta 80 verso il servente Http locale.

iptables -A FORWARD -p tcp -j LOG --log-prefix "Pacchetti inoltrati: " [*Invio*]

Registra le informazioni di tutti i pacchetti inoltrati sfruttando il demone Syslogd.

iptables -A FORWARD -p tcp -j ULOG --ulog-prefix "Pacchetti inoltrati: " [*Invio*]

Registra le informazioni di tutti i pacchetti inoltrati sfruttando il dispositivo netlink in spazio utente. I pacchetti in multicast vengono intercettati da un demone tipo '**ulogd**'.

Per eliminare la regola dalle chain vi sono due modi: il primo conoscendo l'esatta posizione della regola nella chain, la seconda usando lo stesso comando però sostituendo '**-A**' con '**-D**'.

iptables -D INPUT 1 [*Invio*]

Elimina la regola in posizione 1 nella catena di INPUT.

Oppure:

 # iptables -D INPUT -s 127.0.0.1 -p icmp -j DROP [*Invio*]

 # iptables -t nat -D INPUT 2 [*Invio*]

Elimina la regola in posizione 2 nella catena di '**INPUT**' della tabella di '**nat**'.

 # iptables -L [*Invio*]

Visualizza le regole nella tabella di '**filter**'.

 # iptables -t nat -L [*Invio*]

Visualizza le regole nella tabella di '**nat**'.

2.3.14 FTP passivo e attivo

La connessione in modalità ftp richiede due regole separate per la gestione della connessione e del flusso dei dati.

 # iptables -A INPUT -p tcp --sport 21 -m state --state ESTABLISHED -j ACCEPT [*Invio*]

 # iptables -A OUTPUT -p tcp --dport 21 -m state --state NEW,ESTABLISHED -j ACCEPT [*Invio*]

Queste due regole consentono l'accesso al firewall sia in entrata quindi, verso la rete locale, sia in uscita dal firewall.

Il flusso dei dati invece è gestito in due modi diversi: o attraverso l'ftp attivo o attraverso l'ftp passivo.

Ftp attivo

Il cliente ftp invia attraveso il comando '**PORT**' il numero della porta al server di ftp che a sua volta lo contatta utilizzando la porta 20 come origine. In questo caso non si conosce la porta di origine che può essere una qualsiasi sopra la 1023. In questo caso è necessario aggiungere una regola che autorizzi una qualsiasi connessione dalla porta 20 del server ad una qualsiasi del nodo cliente. In generale questa assunzione è insicura.

 # iptables -A INPUT -p tcp --sport 20 -m state --state ESTABLISHED,RELATED -j ACCEPT [*Invio*]

 # iptables -A OUTPUT -p tcp --dport 20 -m state --state ESTABLISHED -j ACCEPT [*Invio*]

La connessione è classificata come '**RELATED**' in quanto ereditata dal traffico generato sulla porta 21; per questo non è indispensabile aggiungere lo stato '**NEW**'.

> Nei clienti di ftp sia grafici che testuali è necessario disabilitare la modalità passiva. Per Ncftp questo avviene digitando '**set passive off**'.

Ftp passivo

 # iptables -A INPUT -p tcp --sport 1024: --dport 1024: -m state --state ESTABLISHED -j ACCEPT [*Invio*]

 # iptables -A OUTPUT -p tcp --sport 1024: --dport 1024: -m state --state ESTABLISHED,RELATED -j ACCEPT [*Invio*]

Nella modalità passiva è il cliente ftp che invia al servente il numero della porta; per questò non è possibile conoscere il numero della porta che può essere scelta in modo arbitrario.

2.4 NAT (*Network Address Translation*)

Nel kernel Linux versione 2.4 è stato introdotta una nuova infrastruttura per controllare i pacchetti TCP/IP denominata "Netfilter". La struttura, rispetto alle versioni precedenti del kernel, è stata interamente riscritta.

Il NAT fa parte della tabella filter.

I pacchetti spediti da un computer sorgente arrivano ad un computer destinatario attraversando numerosi strati intermedi. Il protocollo TCP/IP, non modifica il contenuto di questi pacchetti, quindi arrivano a destinazione inalterati. Il NAT, invece, altera il contenuto di questi pacchetti o dal computer sorgente, o al computer di destinazione.

Questa destinazione è, per il NAT, reversibile così il cambiamento è trasparente all'utente.

2.4.1 Tipi di NAT

La famiglia dei NAT si divide in due: i Source NAT (SNAT) e gli Destination NAT (DNAT).

Nei Source NAT viene modificato l'indirizzo sorgente del primo pacchetto. Questo metodo è sempre condotto in postrouting, prima che il pacchetto sia spedito nella rete.

Nei Destination NAT viene modificato l'indirizzo di destinatione del primo pacchetto. Questo metodo è sempre condotto in prerouting, quando il pacchetto è arrivato in rete locale.

Le forme di DNAT più comuni sono forwarding, load sharing e transparent proxying.

2.4.2 Mascheramento con IpTables

Per mascherare gli indirizzi Ip di una rete locale quando si dispone di pochi indirizzi Ip statici o peggio di un unico indirizzo dinamico (connessione PSTN o ADSL), si procede così:

```
# iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE [ Invio ]
```

Con l'opzione '**-t**' decidiamo la tabella in cui operare. La predefinita è di filter; le altre sono quella di nat e quella di mangle.

Disponendo di un numero Ip statico è preferibile utilizzare il metodo SNAT, in quanto evita al processo di ricalcolare ogni volta l'indirizzo Ip.

```
# iptables -t nat -A POSTROUTING -o ppp0 -j SNAT --to-source x.x.x.x
[ Invio ]
```

La tabella di nat, in particolare, viene consultata quando un pacchetto che sta creando una nuova connessione è intercettato. Le catene predefinite in questa tabella sono PREROUTING, OUTPUT e POSTROUTING.

Ricordiamo che la filter contiene le catene di INPUT, FORWARD e OUTPUT.

Il forwarding di un pacchetto inviato all'indirizzo di un Ip statico verso un indirzzo Ip locale viene così generato:

```
# iptables -t nat -A PREROUTING -p tcp -d x.x.x.x --dport 8080 -j DNAT
--to 172.17.1.10:80 [ Invio ]
```

2.5 Un esempio di firewall

Viene proposto un firewall sia per uso «domestico» che per uso aziendale. La blackbox dispone di due schede di rete, una con indirizzo Ip pubbico dinamico o statico assegnato dal provider, una con indirizzo Ip privato.

La variabile ***$EXTERN_INF*** diversamente può assumere un valore tipo ***eth1***, ***ppp0*** ...

```
# file firewall.sh

#!/bin/sh

EXTERN_INF="ppp0"
LOCAL_INF="eth0"
DNS1="212.216.112.112"
DNS2="81.74.224.227"

# Elimina le regole
iptables -t filter -F
iptables -t filter -X
iptables -t filter -Z

iptables -t nat -F
iptables -t nat -X
iptables -t nat -Z

iptables -t mangle -F
iptables -t mangle -X
iptables -t mangle -Z

# Policy predefinite
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT

# LOG

# Interfaccia di loopback
iptables -t filter -A INPUT -s 127.0.0.0/8 -j ACCEPT
iptables -t filter -A FORWARD -d 127.0.0.0/8 -j ACCEPT

# Spoofing
iptables -t filter -A INPUT -i $EXTERN_INF -s 10.0.0.0/8 -j DROP
iptables -t filter -A INPUT -i $EXTERN_INF -s 172.16.0.0/12 -j DROP
iptables -t filter -A INPUT -i $EXTERN_INF -s 192.168.0.0/16 -j DROP
iptables -t filter -A INPUT -i $EXTERN_INF -s 224.0.0.0/4 -j DROP
iptables -t filter -A INPUT -i $EXTERN_INF -s 240.0.0.0/5 -j DROP
iptables -t filter -A INPUT -i $EXTERN_INF -d 127.0.0.0/8 -j DROP
iptables -t filter -A INPUT -i $EXTERN_INF -d 172.17.255.255 -j DROP

# Interfaccia eth0
iptables -t filter -A INPUT -s 172.16.0.0/12 -j ACCEPT
iptables -t filter -A INPUT -s 192.168.0.0/16 -j ACCEPT
iptables -t filter -A FORWARD -d 172.16.0.0/12 -j ACCEPT
iptables -t filter -A FORWARD -d 192.168.0.0/16 -j ACCEPT

# SYN-FOODING
iptables -t filter -A INPUT -i $EXTERN_INF -p tcp ! --syn -m state --state NEW -j ULOG --ulog-prefi
iptables -t filter -A INPUT -i $EXTERN_INF -p tcp ! --syn -m state --state NEW -j DROP

# FRAGMENTS
iptables -t filter -A INPUT -i $EXTERN_INF -f -j ULOG --ulog-prefix "IpTables fragments:"
iptables -t filter -A INPUT -i $EXTERN_INF -f -j DROP

# DNS1
iptables -t filter -A INPUT -p udp -s $DNS1 --sport 53 -m state --state ESTABLISHED -j ACCEPT
iptables -t filter -A FORWARD -p udp -d $DNS1 --dport 53 -m state --state NEW,ESTABLISHED -j ACCEPT
# DNS2
iptables -t filter -A INPUT -p udp -s $DNS2 --sport 53 -m state --state ESTABLISHED -j ACCEPT
iptables -t filter -A FORWARD -p udp -d $DNS2 --dport 53 -m state --state NEW,ESTABLISHED -j ACCEPT
# DNS (locale)
iptables -t filter -A INPUT -p udp --sport 53 -m state --state ESTABLISHED -j ACCEPT
iptables -t filter -A FORWARD -p udp --dport 53 -m state --state NEW,ESTABLISHED -j ACCEPT

# SSH (uscita)
iptables -t filter -A INPUT -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
iptables -t filter -A FORWARD -p tcp --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT
# SSH (entrata)
iptables -t filter -A INPUT -p tcp -s 0/0 --dport 22 -m state --state NEW,ESTABLISHED -j ACCEPT

# HTTPD (uscita)
iptables -t filter -A INPUT -p tcp --sport 80 -m state --state ESTABLISHED -j ACCEPT
```

Perché l'ftp attivo funzioni come dovrebbe è necessario precaricare il modulo '`ip_conntrack_ftp.o`' ed il modulo '`ip_nat_ftp.o`'.

```
# file /etc/sysctl.conf
...
net/ipv4/ip_forward=1
...
```

Il forwarding dei pacchetti Ip deve essere impostato a 1.

`sh ./firewall.sh` [*Invio*]

Carica le impostazioni del firewall.

`/etc/init.d/iptables save active` [*Invio*]

Salva nel file '`/var/lib/iptables/active`' le impostazioni «attive» per il prossimo avvio della blackbox.

> In genere salvare con questo metedo le impostazioni del firewall è valido per una distribuzione Debian GNU/Linux; tuttavia in altre i meccanismi sono simili.

Capitolo 3

Firewall e Masquerade (obsoleto)

Il Masquerading è una tecnica che consente ad un server LiNUX di fungere da gateway verso Internet per un numero di workstation che non dispongono di un indirizzo IP valido. Un computer di una intranet potrebbe spedire pacchetti all'esterno, esempio una richiesta HTTP, il gateway intercetta la richiesta e "maschera" i pacchetti fingendo che il destinatario sia esso stesso. Alla replica della richiesta il gateway inoltra i pacchetti al computer della rete locale che ne ha fatto richiesta.

Questa funzionalità prende il nome di NAT (*Network Address Translation*) altre tecniche simili al masquerading sono il Transparent Proxying ed il portforwarding. L'utility '`ipchains`' serve per modificare le regole di mascheramento degli indirizzi.

L'installazione di una release del kernel 2.4.x impone l'utilizzo di '`iptables`'.

3.1 IP Firewalling

Prima di entrare nel merito della tecnica di "maschermento" esaminiamo le capacità di firewall dell'utility '`ipchains`'. Essa permette di maneggiare regole in entrata, uscita e di forwarding per i protocolli TCP, UDP ed ICMP .

Per ogni regola è in grado di accettare il pacchetto ('**`ACCEPT`**'), rifiutarlo ('**`DENY`**') e di rifiutarlo con spedizione di messaggio di blocco ('**`REJECT`**').

Il logging può essere aggiunto con l'opzione -l.

```
# ipchains -A input -s 192.168.1.0/24 -d 192.168.1.1 -j DENY [ Invio ]
```

Tutti i pacchetti che provengono dalla rete 192.168.1.0 ('**`-s`**' indica la sorgente) e destinati all'indirizzo IP 192.168.1.1 ('**`-d`**' indica la destinazione) vengono ignorati; se si utilizzasse, al posto di '**`DENY, REJECT`**' l'host sorgente riceverebbe una segnalazione di errore (messaggio '**`ICMP`**'). L'opzione '**`-A`**' (append) aggiunge una regola a quelle eventualmente già presenti. Analogamente si può sperimentare '**`-A output`**' per i pacchetti in uscita dal firewall.

L'interfaccia cui applicare una policy (regola) si specifica con l'opzione '**`-i`**'. In una firewall è necessario applicare le regole per entrambe le interfacce, per chiarezza e comodità si può procedere prima configurando il network locale, poi l'interfaccia (scheda di rete o connessione PPP) verso Internet e/o il mondo esterno.

Un banale script che accetti tutti i pacchetti e mascheri quelli con indirizzo locale verso l'esterno attraverso un'interfaccia PPP.

```
# file /etc/rc.d/rc.firewall

#!/bin/sh

ipchains -F

ipchains -P input ACCEPT
ipchains -P output ACCEPT
ipchains -P forward ACCEPT

ipchains -A forward -i ppp0 -s 192.168.1.0/24 -j MASQ
ipchains -A forward -i ppp0 -s 192.168.2.0/24 -j MASQ
```

Rimuove (-F *flush*) tutte le regole esistenti dei filtri di *Forward*, *Input* ed *Output*. Successivamente imposta le politiche di default in ACCEPT, ossia accetta tutti i pacchetti.

è immediato che le regole vengono applicate in ordine di immissione: "prima si accetta tutto poi si nega quello che serve".

Come si potrebbe rifiutare di ricevere traffico verso gli indirizzi di *broadcast* e del *network*:

 # ipchains -A input -i eth0 -d 192.168.2.0 -j DENY [*Invio*]

 # ipchains -A input -i eth0 -d 192.168.2.255 -j DENY [*Invio*]

Ho aggiunto -i perché con questa opzione si indica la scheda di rete, quindi il network, dove verranno applicate le politiche; è normale aspettarsi che una *firewall* contenga almeno due schede di rete.

Allo stesso modo il traffico con una rete locale può tranquillamente essere accettato in blocco:

 # ipchains -A input -i eth1 -s 192.168.1.0/24 -j ACCEPT [*Invio*]

 # ipchains -A ouput -i eth1 -d 192.168.1.0/24 -j ACCEPT [*Invio*]

Un esempio più impegnativo può essere ben rappresentato con la comunicazione tra due DNS in configurazione di *caching only*, per la prima volta si utilizzerà il protocollo UDP.

Con la variabile ${DNS} e ${IPADDR} si indicano rispettivamente l' IP del DNS esterno e l'IP statico della scheda di rete del firewall che configuriamo (eth0 o ppp0).

 # ipchains -A input -i eth0 -p udp -s ${DNS} 53 -d ${IPADDR} 53 -j ACCEPT [*Invio*]

 # ipchains -A output -i eth0 -p udp -s ${IPADDR} 53 -d ${DNS} 53 -j ACCEPT [*Invio*]

I due numeri 53 indicano proprio la porta utilizzata dai due DNS server, altrimenti tutto il traffico UDP sarebbe accettato. Vediamo ora come andrebbe configurata la parte client: ossia quando un DNS viene interrogato da un client.

Tutti i servizi client per comunicare utilizzano le porte comprese tra la 1024 e la 65535, il servizio viene quindi "replicato". Da queste porte parte la richiesta di comunicazione verso la porta dell'host remoto, che è quella del servizio server; nel caso del DNS è la 53.

 # ipchains -A input -i eth0 -p udp -s ${DNS} 53 -d ${IPADDR} 1024:65535 -j ACCEPT [*Invio*]

 # ipchains -A output -i eth0 -p udp -s ${IPADDR} 1024:65535 -d ${DNS} 53 -j ACCEPT [*Invio*]

 # ipchains -A input -i eth0 -p tcp ! -y -s ${DNS} 53 -d ${IPADDR} 1024:65535 -j ACCEPT [*Invio*]

 # ipchains -A output -i eth0 -p tcp -s ${IPADDR} 1024:65535 -d ${DNS} 53 -j ACCEPT [*Invio*]

In ordine: dal DNS porta 53 al clients che comunica da una porta compresa tra 1024 e 65535.

L'opzione -y è valida per il TCP, permette la connessione solo in una direzione (dai clients al DNS in questo caso). Usa la tecnica del TCP SYN packets only.

Si potrebbe accettare la connessione verso un web server esterno, ma non accettare (in entrata: -A input) la connessione verso la nostra rete (un probabile attacco?). è sufficiente quindi bloccare solo i pacchetti TCP che consentono la connessione e non quelli che "portano" i dati.

Specificando ! -y il senso è invertito.

Per concludere un altro esempio che riguarda il servizio di telnet, le prime due regole permettono la connessione verso un server remoto, le ultime due da un host remoto.

 # ipchains -A output -i eth0 -p tcp -s ${IPADDR} 1024:65535 -d 0.0.0.0/0 23 -j ACCEPT [*Invio*]

 # ipchains -A input -i eth0 -p tcp ! -y -s 0.0.0.0/0 23 -d ${IPADDR} 1024:65535 -j ACCEPT [*Invio*]

 # ipchains -A input -i eth0 -p tcp -s 0.0.0.0/0 1024:65535 -d ${IPADDR} 23 -j ACCEPT [*Invio*]

 # ipchains -A output -i eth0 -p tcp ! -y -s ${IPADDR} 23 -d 0.0.0.0/0 1024:65535 -j ACCEPT [*Invio*]

Per ogni porta posso applicare questa regole dove siano entrambe necessarie: spesso la parte server è superflua, mentre quella client può essere sempre necessaria; per esempio: il server di posta non risiede certamente nella firewall, mentre in una firewall si può entrare in telnet o la firewall stessa è il web server.

Per facilitare la comprensione dell'opzione -y spieghiamo la seconda riga: non si accettano connessioni in uscita (pacchetty TCP SYN) verso la porta 23 mentre il server telnet esterno sta comunicando.

Nell'ultima riga si impediscono attacchi dal server telnet interno verso gli host remoti.

Tutte queste regole andrebbero raggrupate in un file (ex: '`/etc/rc.d/rc.firewall`') da avviare ad ogni reboot della macchina oppure ad ogni modifca. Un semplice esempio dove la firewall abbia due schede di rete, la prima con IP statico (accesso ad Internet) e la seconda con IP locale:

```
# file /etc/rc.d/rc.firewall

#!/bin/sh

echo "Starting firewall..."

# Indirizzo della rete locale
LOCAL_IF="eth0"
LOCAL_NET="192.168.1.0/24"
LOCAL_IP="192.168.1.1"

# Indirizzo ip statico assegnato alla nostra scheda di rete.
# L'interfaccia potrebbe essere sostituita con ppp0 se la connessione avviene
 con protocollo PPP
EXT_IF="eth1"
EXT_NET="xxx.xx.xx.xx/255.255.255.240"
EXT_IP="xxx.xx.xx.xx"

# Interfaccia di loopback
LO_IF=lo

# Porte privileged ed unprivileged
PRIVPORTS="0:1023"
UNPRIVPORTS="1024:65535"

# Definizione di range di porte
SSH_PORTS="1020:1023"

# Inserimento moduli ip_masq
/sbin/modprobe ip_masq_ftp
/sbin/modprobe ip_masq_raudio
/sbin/modprobe ip_masq_irc
/sbin/modprobe ip_masq_quake
/sbin/modprobe ip_masq_cuseeme
/sbin/modprobe ip_masq_vdolive

# Settiamo il timeout per il mascheramento
#   3600 secondi per la connessione TCP
#       10 secondi per il traffico dopo che il TCP/IP "FIN" packet è
stato
 ricevuto
#     160 secondi per il traffico UDP
#
/sbin/ipchains -M -S 7200 10 160
```

```
# Pulizia di tutte le regole
/sbin/ipchains -F

# Politiche di default
/sbin/ipchains -P input DENY
/sbin/ipchains -P output REJECT
/sbin/ipchains -P forward REJECT

# Pacchetti per l'interfaccia di loopback
/sbin/ipchains -A input  -i ${LO_IF} -j ACCEPT
/sbin/ipchains -A output -i ${LO_IF} -j ACCEPT

# Accetta tutto il traffico per la rete locale
/sbin/ipchains -A input  -i ${LOCAL_IF} -s ${LOCAL_NET} -j ACCEPT
/sbin/ipchains -A output -i ${LOCAL_IF -d $LOCAL_NET} -j ACCEPT

# Mascheramento del traffico interno
/sbin/ipchains -A forward -i ${EXT_IF} -s ${LOCAL_NET} -j MASQ

# Rifiuta tutti i pacchetti spoofed, ossia quelli che hanno indirizzo EXT_IP
/sbin/ipchains -A input  -i ${EXT_IF} -s ${EXT_IP} -j DENY

# La stessa cosa si potrebbe fare per gli indirizzi in classe A, B, C ed il
 loopback.
# Un esempio per gli indirizzi in classe C.

/sbin/ipchains -A input -i ${EXT_IF} -s 192.168.0.0/16 -j DENY
/sbin/ipchains -A input -i ${EXT_IF} -d 192.168.0.0/16 -j DENY
/sbin/ipchains -A output -i ${EXT_IF} -s 192.168.0.0/16 -j DENY
/sbin/ipchains -A output -i ${EXT_IF} -d 192.168.0.0/16 -j DENY

# Blocca connessione NFS (TCP ed UDP) provenienti dall'esterno
/sbin/ipchains -A input -i ${EXT_IF} -p tcp -y -d ${EXT_IP} 2049 -j DENY
/sbin/ipchains -A output -i ${EXT_IF} -p tcp -y -d 0.0.0.0/0 2049 -j REJECT

/sbin/ipchains -A input -i ${EXT_IF} -p udp -d ${EXT_IP} 2049 -j DENY
/sbin/ipchains -A output -i ${EXT_IF} -p udp -d 0.0.0.0/0 2049 -j REJECT

# Accetta connessioni sicure attraverso il protocollo SSH
# Host che acceddono ad un server SSH remoto
/sbin/ipchains -A output -i ${EXT_IF} -p tcp -s ${EXT_IP} ${UNPRIVPORTS} -d
 0.0.0.0/0 22 -j ACCEPT
/sbin/ipchains -A input  -i ${EXT_IF} -p tcp ! -y -s 0.0.0.0/0 22 -d ${EXT_IP}
 ${UNPRIVPORTS} -j ACCEPT

/sbin/ipchains -A output -i ${EXT_IF} -p tcp -s ${EXT_IP} ${SSH_PORTS} -d
 0.0.0.0/0 22 -j ACCEPT
/sbin/ipchains -A input  -i ${EXT_IF} -p tcp ! -y -s 0.0.0.0/0 22 -d ${EXT_IP}
 ${SSH_PORTS} -j ACCEPT

# Accetta connessioni sicure da hosts remoti
/sbin/ipchains -A input  -i ${EXT_IF} -p tcp -s 0.0.0.0/0 ${UNPRIVPORTS} -d
 ${EXT_IP} 22 -j ACCEPT
/sbin/ipchains -A output -i ${EXT_IF} -p tcp ! -y -s ${EXT_IP} 22 -d 0.0.0.0/0
 ${UNPRIVPORTS} -j ACCEPT

/sbin/ipchains -A input  -i ${EXT_IF} -p tcp -s 0.0.0.0/0 ${SSH_PORTS} -d
 ${EXT_IP} 22 -j ACCEPT
/sbin/ipchains -A output -i ${EXT_IF} -p tcp ! -y -s ${EXT_IP} 22 -d 0.0.0.0/0
 ${SSH_PORTS} -j ACCEPT

echo "done."
```

Le distribuzione Red Hat dalla versione 7 includono un utility grafica con la quale ogni utente può costruire la propria firewall. Di seguito propongo una lista delle porte da filtrare comunque per rendere la vostra rete sicura.

Tabella 3.1. Porte e servizi da filtrare

Servizio	socket	porta
'**DNS**'	socket	53 (TCP)
'**tftpd**'	socket	69 (UDP)
'**link**'	socket	87 (TCP)
'**lpd**'	socket	515 (TCP)
'**uucpd**'	socket	540 (TCP)
'**openwindows**'	socket	2000 (UDP e TCP)
'**X windows**'	socket	6000+ (UDP e TCP)

3.2 Transparent Proxying

Il *transparent proxying* permette di inoltrare il traffico per un host esterno richiesto da un client di una rete locale verso un server locale, tipicamente un proxy.

Sostanzialmente non cambia molto rispetto all'utilizzo visto in precedenza, l'utilizzo comunque resta limitato alle politiche in entrata ('**-I**'), diversamente sarà restituito un messaggio di errore.

Questa tecnica è molto sfruttata in congiunzione con un *proxy server* dove tutto il traffico HTTP finisce nella cache del proxy risparmiando banda Internet. Semplificando, se il nostro collegamento al provider avviene attraverso una linea dedicata, come una ADSL a 640 Kbps, e dobbiamo gestire più client (o Network Computer che si voglia) si capisce che la linea può saturarsi velocemente. Non resta che obbligare il personale ad utilizzare il proxy server.

La configurazione del browser però non è a portata di tutti gli utenti, molto spesso si dimentica, si cambia e si rischia di avere in parte risolto il problema; meglio quindi una soluzione drastica. Perché non rendere tutto trasparente per l'utente finale?

L'istruzione per inoltrare il traffico WEB al proxy è la seguente:

 # /sbin/ipchains -A accept -p tcp -S 192.168.0.0/16 -d 0.0.0.0/0 80 -j REDIRECT 8080 [*Invio*]

Il traffico TCP del network 192.168.0.0/16 destinato all'esterno verso la porta 80 (WWW server), viene reinderizzato alla porta 8080 della *firewall* (-j REDIRECT 8080). Questa porta viene generalmente usata per i proxy server.

Tuttavia questa politica deve essere attivata prima delle altre in entrata per questo uno *script* modificato rispetto al precedente pronto per essere utilizzato in congiunzione con Squid, il proxy server più utilizzato.

```
# file /etc/rc.d/rc.firewall

#!/bin/sh

echo "Starting firewall..."

/sbin/ipchains -F

/sbin/ipchains -P input REJECT
/sbin/ipchains -P output ACCEPT
/sbin/ipchains -P forward DENY

/sbin/ipchains -A input -p tcp -s 192.168.0.0/16 -d 0.0.0.0/0 80 -j REDIRECT
8080
/sbin/ipchains -A input -p all -j ACCEPT

/sbin/ipchains -A forward -i ppp0 -s 192.168.1.0/24 -j MASQ
/sbin/ipchains -A forward -i ppp0 -s 192.168.2.0/24 -j MASQ

echo "done."
```

3.3 Configurazione di Windows

Più client Windows dovranno essere configurati assegnando (Pannello di controllo->Rete) numero IP, server *DNS* ed infine l'IP del *gateway*. Dopo il riavvo del sistema si può navigare tranquillamente sfruttando tutte le utility (Netscape o Explorer compresi) standard fornite acquistando il sistema operativo; sarà quindi possibile lanciare una sessione di telnet, di ftp effettuare un ping (ovviamente se concesso dalla firewall) sfruttando la LAN locale, il Server Linux (con il masquerade attivo), ed un unico accesso verso il provider.

Capitolo 4

Stunnel

4.1 Introduzione

Stunnel è una applicazione costruita per garantire il passaggio cifrato di tutte le informazioni di una connessione tra un servizio client locale e un demone presente in un host remoto e viceversa. Il sistema di cifratura utilizzato è lo SSL (*Security Socket Layer)*.

Stunnel cifra e spedisce attraverso un canale che potremmo definire sicuro il contenuto di una connessine in chiaro. Per questo motivo trova impiego nella realizzazione di VPN (*Reti Private Virtuali*). Stunnel cifra connessioni che sfruttano protocolli tipo POP3, SMTP, IMAP...

Stunnel può essere eseguito in modalità demone e in modalità client.

4.2 Stunnel

`stunnel` [***options***] `program` [`--` ***progname args***]

Alcune opzioni

`-D level`

Aumenta il livello per le informazioni di debug. Il valore di level è compreso tra 0 e 7 equivalente al modello syslog.

`-C cipherlist`

Seleziona i metodi di cifratura consentiti. La lista viene separata dal carattere ":". Esempio DES-CBC-SHA:IDEA-CBC-MD5

`-c`

Avvia Stunnel in modalità client. Il servizio remoto usa lo SSL.

`-p pem`

Specifica il percorso del certificato in formato pem.

`-d [host:]port`

Avvia Stunnel in modalità demone. Se host non viene indicato allora tutti gli indirizzi Ip faranno riferimento al server locale.

`-v level`

Verifica il certificato. I valori ammissibili sono 1, 2 o 3.

`-V`

Visualizza informazioni sul pacchetto Stunnel.

`-h`

Visualizza l’aiuto testuale.

4.2.1 Esempi pratici

Nel primo esempio creiamo un tunnel tra due PC per garantire una connessione sicura con il servizio Telnet.

Nella macchina server digitiamo il seguente comando:

stunnel -d 992 -r localhost:23 [*Invio*]

La porta 992 è riservata per la connessione telnet via SSL. Il nome del nodo localhost si poteva omettere.

Nella macchina client digitiamo il seguente comando:

stunnel -c -d 992 -r 172.17.1.254:992 [*Invio*]

L'indirizzo Ip assegnato è quello dell'ipotetico server dove è avviato il servizio '**telnetd**' e attivo il servizio Stunnel in ascolto nella porta 992.

Nel prossimo esempio creiamo un tunnel tra due nodi per garantire una connessione sicura tra un server MySQL ed il client presente in un server remoto.

Il demone '**mysqld**' opera sulla porta 3306, il tunnel verrà creato sulla porta immediatamente successiva: la 3307.

stunnel -d 3307 -r localhost:3306 [*Invio*]

Nella macchina client digitiamo il seguente comando:

stunnel -c -d 3306 -r 172.17.1.254:3307 [*Invio*]

Nel prossimo esempio creiamo una connessione cifrata per il demone web che interroga la porta 443.

stunnel -d 443 -r 80 [*Invio*]

Il nodo remoto può interrogare il server con il protocollo https.

Nel prossimo esempio creiamo un tunnel per il servzio VNC.

stunnel -d 7777 -r 5901 [*Invio*]

Si può quindi eseguire il demone '**vncserver**'.

vncserver :1 [*Invio*]

Nella macchina client digitiamo il seguente comando:

$ **stunnel -c -d 5901 -r 172.17.1.254:777** [*Invio*]

Capitolo 5

DNS *(Domain Name System)*

5.1 Introduzione

Il DNS è un programma che si occupa di risolvere i nomi di dominio.

Le informazioni delle risorse sono memorizzate nel programma server, che si occupa di inviarle ai client che ne fanno esplicita richiesta.

L'informazione che il risolutore fornisce riguardano le caratteristiche dei nodi della rete: indirizzo Ip, indirizzo del server di posta, e così via. La memorizzazione delle informazioni avviene in modo gerarchico ed esistono enti autoritativi che ne garantiscono la corretta gestione e manutenzione.

I nodi principali di questa organizzazione sono chiamati domini. Ogni dominio dispone di sottodomini. Al dominio viene assegnato un nome unico che lo identifica. L'insieme di tutti i domini genera lo spazio dei nomi di dominio.

L'organizzazione dei domini è suddivisa in zone. Ad ogni zona viene assegnato un file di dati differente dalle altre. I server di dominio hanno potere autoritativo nella zona di pertinenza e limitatamente solo a questa. Ad una zona possono appartenere più nomi di dominio.

È fondamentale chiarire la differenza che intercorre tra il concetto di dominio e il concetto di zona.

Consideriamo il dominio '**`inf.besta`**', questo potrebbe contenere nodi come '**`c1.lab-1.inf.besta`**' oppure '**`c2.lab-2.inf.besta`**' anche se la zona '**`inf.besta`**' ha potere autoritativo solo sui domini '**`lab-1.inf.besta`**' o '**`lab-2.inf.besta`**'.

La zona può quindi includere solo una parte del dominio e delegare ad altri la parte rimanente. Queste parti sono comunque contigue nello spazio dei domini. Ogni voce in un DNS rappresenta un dominio anche se esso non ha parti terminali.

Il dominio viene costruito con due o più nomi separate da un punto.

I dominio radice vengono chiamati *root* e sono gli unici domini che non sono sottodomini di domini. La lista di questi domini radice viene caricata all'avvio di ogni server DNS.

5.2 Risolvere i nomi con il file di host

5.2.1 /etc/hosts

I nomi dei nodi di un dominio possono essere risolti facilmente modificando il file '`/etc/hosts`' e costruendo una tabella come in listato 5.1. Ad ogni indirizzo numerico si fa corrispondere un nome di dominio ed eventualmente un alias. Ogni nodo dovrà contenere lo stesso file di '`hosts`' correttamente compilato ed ogni modifica dovrà essere distribuita a tutti i nodi contemporaneamente.

Il problema di realizzare una soluzione di questo tipo cresce con il numero di nodi della rete. Per questo motivo è preferibile adottare un servizio DNS.

Listato 5.1. File '`/etc/hosts`'.

```
127.0.0.1     localhost.localdomain localhost
172.17.1.1    c1.lab-1.inf.besta    c1
172.17.1.2    c2.lab-1.inf.besta    c2
172.17.1.3    c3.lab-1.inf.besta    c3
172.17.2.1    s1.lab-2.mat.besta    s1
172.17.2.2    s2.lab-2.mat.besta    s2
172.17.2.3    s3.lab-2.mat.besta    s3
```

5.3 Risolvere i nomi con BIND

5.3.1 /etc/host.conf

Il demone che fornisce il servizio DNS distribuito con il pacchetto BIND è Named. Named viene attivato eseguendo il demone '**named**'.

Il sistema di risoluzione dei nomi sfrutta anche il servizio DNS se nel file '`/etc/host.conf`' si aggiunge la voce '**bind**' nella riga con la voce '**order**'.

```
# file /etc/host.conf

order  bind, hosts, nis
multi  on
```

La voce '**order**' impone l'ordine di ricerca per risolvere i nomi; mentre, la voce '**multi**', consente ad un host di essere associato a più indirizzi IP. Esistono altre voci che consentono di controllare l'esatta corrispondenza del numero IP con il nome mnemonico ('**nospoof**') e la possibilità di elimare il nome di dominio dal nome mnemonico ('**trim**').

5.4 /etc/resolv.conf

Il file '`/etc/resolv.cond`' contiene l'indirzzo Ip del fornitore di servizio per la risoluzione dei nomi. Ogni riga contiene o commenti, che iniziano con il carattere "#" oppure parole chiave per inidicare una lista di domini o una lista di risolutori di dominio.

Per indicare più di un risolutore è necessario inserirne uno per righa. Generalmente il primo viene genericamente identificato come *nameserver primario*, il secondo come *nameserver secondario* ed il terzo come *nameserver terziario*.

Per facilitare la risoluzione dei nomi della rete interna è sufficiente indicare una lista di domini di ricerca.

In listato 5.3 un esempio.

Listato 5.3. File '`/etc/resolv.conf`'.

```
nameserver 172.17.1.254
nameserver 172.17.1.253
search     inf.besta mat.besta
```

Nella prima riga si imposta il risolutore primario, mentre nella seconda il risolutore secondario. La terza riga contiene uno o più di domini di ricera separati da uno spazio.

`nameserver`

Imposta l'indirizzo Ip del DNS.

`search`

Accetta uno o più nomi di dominio. Ha la precedenza sulla voce '**domain**'.

`domain`

Accetta solamente un nome di dominio; viene ignorata se è presente contemporaneamente una riga con la voce '**search**'.

5.5 /etc/named.conf

Il file '`/etc/named.conf`' è di configurazione per il servizio Named. In listato 5.4 è proposta una semplice configurazione di Named.

Il file è diviso in commenti e parole chiave. I commenti iniziano con il carattere "#" o è possibile adottare lo stile del linguaggio di programmazione C/C++.

Listato 5.4. File '`/etc/named.conf`'.

```
options {
          directory "/var/named";
          allow-query { any; };
};

zone "." {
          type hint;
          file "named.ca";
}

zone "0.0.127.in-addr.arpa" {
          type master;
          file "named.local";
}

zone "17.172.in-addr.arpa" {
          type master;
          file "zone/172.17.1";
}

zone "inf.besta" {
          type master;
          file "zone/inf.besta";
};

zone "mat.besta" {
          type master;
          file "zone/mat.besta";
};
```

La zona di tipo '**.**' contiene i server DNS principali, ossia quelli radice dove vengono inoltrate tutte le richieste di risoluzione se il DNS interrogato non è auoritativo per quella zona.

La zona di tipo '**master**' è autoritativa per il dominio di competenza, ossia i domini gestiti dal server DNS.

5.5.1 Parole chiave in /etc/named.conf

Le parole chiave iniziano un blocco di regole che Named carica all'avvio del servizio. Il confine di questi è racchiuso tra parantesi graffe "{ *blocco* }" e termina con il carattere ";".

acl
: Definisce un nome che contiene un gruppo di inidirizzi IP. Il suo scopo è fondamentalmente legato ai diritti di accesso al servizio Named.

logging
: Servizio per memorizzare su file le informazioni sulle attività svolte da Named.

options
: Configurazione globale del servizio Named.

zone
: Definisce una zona.

include

Inserisce un file.

5.5.1.1 acl

Definisce un nome simbolico che viene utilizzato principalmente per il controllo dell'accesso al servizio Named (*Access Control Lists*).

Sintassi d'uso:

```
acl nome_acl {
        lista_indirizzi_ip
};
```

Il nome simbolico deve essere definito prima di utilizzarlo. Named definisce internamente delle liste di accesso.

none

Nessun numero Ip viene assegnato al nome simbolico.

any

Tutti i numeri Ip sono assegnati al nome simbolico.

localhost

Sono assegnati i numeri Ip delle interfacce di rete del nodo locale.

localnets

Qualsiasi nodo o rete verso il quale il nodo locale ha un'interfaccia.

5.5.1.2 logging

Definisce un sistema di memorizzazione delle informazioni sulle attività svolte da Named particolarmente interessante.

Sintassi d'uso:

```
logging {
  [ channel nome_canale {
    ( file percorso
        [ versions ( number | unlimited ) ]
        [ size dimensione ]
      | syslog tipo_di_programma
      | stderr
      | null );
    [ severity ( critical | error | warning | notice |
                 info | debug [ livello ] | dynamic ); ]
    [ print-category yes | no; ]
    [ print-serverity yes | no; ]
    [ print-time yes | no; ]

  }; ]
  [ category nome_categoria {
    nome_canale ; [ nome_canale ; ...]
  }; ]
  ...
};
```

Con i metodi di uscita definiti in '**channel**' sono riportati in '**category**' per stabilire il tipo e la quantità di informazioni da registrare.

La memorizzazione delle informazioni viene attivata solamente quando l'intero file di configurazione '`/etc/named.conf`' viene letto senza errori.

In '**`channel`**' si definisce il percorso del file di memorizzazione o evenutalmente altri canali di uscita e il tipo di informazione restituita (Il default è info).

In '**`file`**' si possono definire la dimensione massima del file di attività ('**`size`**') e il numero totale di revisione dello stesso file una volta riaperto. Se si superano le dimensioni massime indicate Named non registra più niente fino a quando non verrà riaperto. Le revisioni vengono indicate come *nomefile.x*, dove *x* vale al massimo 99 se si specifica '**`unlimited`**'.

Con '**`syslog`**' al posto di '**`file`**' le informazioni da registrare vengone gestite dal demone '**`syslogd`**'. Solo i messaggi che rientrano almeno nel livello imposto da '**`severity`**' vengono registrati; tutti gli altri scartati.

Il campo '**`stderr`**' invia i messaggi nell'uscita appropriata per gli errori. È opportuno l'utilizzo solo per il debug quando Named è avviato come processo in primo piano.

Il campo '**`null`**' invio tutti i messaggi al file di dispositivo nullo.

I campi con '**`print-`**' registrano informazioni dettagliate come la data e la categoria.

Di seguito un esempio di definizione di canale:

```
logging {
        channel "canale_query" {
                file "query_inf_besta.log" versions 4 size 1M;
                severity        info;
                print-category  yes;
                print-severity  yes;
                print-time      yes;
        };

        channel "canale_client" {
                file "client_in_besta.log" versions 4 size 1M;
                severity        debug 3;
                print-category  yes;
                print-severity  yes;
                print-time      yes;
        };

        category "client" {
                "canale_client";
        };

        category "queries" {
                "canale_query";
        };
};
```

I canali così definiti registrano le informazioni per le categorie '**`client`**' e '**`queries`**'.

Una query effettuata da un cliente registra questa traccia nel file di registro '`query_inf_besta.log`'.

```
Sep 07 18:02:53.557 queries: info: client 172.17.1.100#32770: query: linux.inf.besta IN A
```

Si possono individuare le tre colonne che contraddistinguono il tempo esatto in cui è stata risolto un dominio, il tipo di categoria ed il livello di informazione richiesto.

La traccia lasciata, invece, nel file di registro '`client_inf_besta.log`' è la seguente:

```
Sep 07 18:10:18.062 client: debug 3: client @0x40375268: accept
Sep 07 18:10:18.063 client: debug 3: client @0x40375728: accept
Sep 07 18:10:18.063 client: debug 3: client @0x40375be8: accept
Sep 07 18:10:18.065 client: debug 3: client 172.17.1.254#32787: UDP request
Sep 07 18:10:18.065 client: debug 3: client 172.17.1.254#32787: notify
Sep 07 18:10:18.065 client: debug 3: client 172.17.1.254#32787: send
Sep 07 18:10:18.065 client: debug 3: client 172.17.1.254#32787: sendto
Sep 07 18:10:18.065 client: debug 3: client 172.17.1.254#32787: senddone
Sep 07 18:10:18.065 client: debug 3: client 172.17.1.254#32787: next
Sep 07 18:10:18.065 client: debug 3: client 172.17.1.254#32787: endrequest
Sep 07 18:10:18.569 client: debug 3: client 172.17.1.254#32787: UDP request
Sep 07 18:10:18.569 client: debug 3: client 172.17.1.254#32787: notify
Sep 07 18:10:18.569 client: debug 3: client 172.17.1.254#32787: send
Sep 07 18:10:18.570 client: debug 3: client 172.17.1.254#32787: sendto
Sep 07 18:10:18.570 client: debug 3: client 172.17.1.254#32787: senddone
Sep 07 18:10:18.570 client: debug 3: client 172.17.1.254#32787: next
Sep 07 18:10:18.570 client: debug 3: client 172.17.1.254#32787: endrequest
```

I canali predefiniti in Named sono quattro (vedi listato 5.10).

Listato 5.10. I canali predefiniti in Named.

```
channel "default_syslog" {
        syslog daemon;
        severity info;
};

channel "default_debug" {
        file "named.run";
        severity dynamic;
};

channel "default_stderr" {
        stderr;
        severity info;
};

channel "null" {
        null;
};
```

Con il metodo '**`category`**' si inviano i messaggi da registrare nei canali opportuni. Le categorie sono definite in Named e con gli aggiornamenti potrebbero subire variazioni o nuove aggiunte.

Se per una categoria non vengono definiti dei canali, i messaggi di questi vengono inviati alla categoria di '**`default`**'. Se non viene specificata nessuna categoria è usata quella di '**`default`**' così definita:

```
category "default" {
        "default_syslog"; "default_debug";
};
```

Una categoria non gradita può essere inviata la canale '**`null`**'.

```
category "update" {
        "null";
};
```

Per la lista completa delle categoria vedi tabella 5.1.

Tabella 5.1. Le categorie di Named.

Nome categoria	Comportamento
default	La categoria base dove confluiscono tutte le categorie che non hanno ricevuto una configurazione.
general	Ogni informazione che non è stata registrata in una categoria viene memorizzata qui.
database	Messaggi relativi al database interno gestito da Named per memorizzare zone e dati di cache.
security	Richieste accettate e respinte.
config	Informazioni associate alla sintassi del file di configurazione.
resolver	Messaggi relativi alla risoluzione dei nomi da parte del demone.
xfer-in	Messaggi relativi al trasferimento delle zone.
xfer-out	Messaggi relativi all'invio delle zone.
notify	Messaggi relativi al protocollo NOTIFY.
client	Richieste da parte dei fruitori del servizio.
network	Normali operazioni di rete.
update	Informazioni associate all'aggiornamento dinamico.
queries	Messaggi associati alle query.
dispatch	Invio dei pacchetti in entrata ai processi del server dove verranno elaborati.
dnssec	Protocolli DNSSEC e TSIG.
lame-servers	Messaggi relativi ai *Lame server*.

`directorty`

Indica la directory dove risiedono i file di configurazione per il servizio di BIND;

`zone`

Indica una zona nella quale il server è responsabile.

5.5.1.3 options

5.6 `/etc/bind`

Questa directory contiene tutte le informazioni riguardanti le zone gestite, ossia di competenza per questo dominio. Ogni file contiene dei campi; indicheremo questi campi con il nome di *resource records* o più sinteticamente RR.

Il formato riga è il seguente: '**`{origin} {ttl} {class} {record type} {record specific data}`**'.

`owner`

Indica il dominio al quale il record è destinato; in caso di omissione fa riferimento all'ultimo inserito.

`ttl`

Indica il tempo di vita (*time of live*) e dà validità all'informazione recuperata dal server; in caso di omissione fa riferimento al volare dato nel record '**`SOA`**'.

`class`

Indica la famiglia di protocollo utilizzato;praticamente assume sempre il valore '**`IN`**'; possibili valori sono '**`IN`**' e '**`CH`**'; in caso di omissione fa riferimento al valore dato nel record '**`SOA`**'.

`Record type`

Descrive il tipo di risorsa: '**`A`**', '**`SOA`**', '**`PTR`**', '**`CNAME`**', '**`HINFO`**', '**`MX`**' oppure '**`NS`**'.

`Record specific data`

sono dati associati alle risorsa precedente.

Il campo più importante, SOA, descrive la zona di autorità ed è seguita da un certo numero di record. Le eventuali parentesi servono per raggruppare più dati.

5.6.1 `/etc/bind/db.inf.besta`

Un esempio di un file che gestisce una zona:

```
;                               Origin          Contact
$TTL 604800
@       IN              SOA     inf.besta.      root.router-lab1.inf.besta.  (
                                        1997022700 ; Serial
                                        28800      ; Refresh
                                        7200       ; Retry
                                        604800     ; Expire
                                        86400 )    ; Minimum
@       IN              NS      router-lab1.inf.besta.
@       IN              MX 10   router-lab1.inf.besta.

c1-lab1.inf.besta.      A       172.17.1.1
c2-lab1.inf.besta.      A       172.17.1.2
c3-lab1.inf.besta.      A       172.17.1.3
c4-lab1.inf.besta.      A       172.17.1.4
router-lab1.inf.besta.  A       172.17.1.254
www.inf.besta.          CNAME   c1-lab1.inf.besta.
mail.inf.besta.\011        CNAME   router-lab1.inf.besta.
```

origin

Il nome del host primario.

contact

L'indirizzo di posta elettronica della persona da contattare adetta alla manutenzione della zona.

serial

Il numero di versione del file di zona, viene usato solamente dai file server secondari.

refresh

Indica in secondi quanto deve attendere un server secondario prima di aggiornare i dati dal server primario.

retry

Indica in secondi il periodo di tempo prima che il server secondario riprovi il collegamento se il primo tentativo non fosse andato a termine.

expire

Indica in secondi il periodo dopo il quale il secondario scarta i dati del primario se non fosse stato in grado di contattarlo.

minimum

Riguarda i singoli campi '**RR**' che non hanno il parametro ttl predefinito.

Il carattere @ all'inizio del file indica l'origine, alternativamente si usa la direttiva $ORIGIN.

Oltre al '**SOA**' (Start of authority) ed '**NS**' (name server) vale la pena di citare la direttiva MX (mail exchanger) seguita da un numero che ne indica la priorità, più il nome di un host che fungerà da mail server primario per quella zona od host (campo name che precedeva).

Ad ogni riga contenente un nome mnemonico è associato un indirizzo Ip (legati dal campo '**A**'). Il punto alla fine del nome dell'host indica il nome completo, altrimenti si dovrebbe solo specificare il nome host (Il DNS completa automaticamente il nome host con l'origine: '**c1**' diventa '**c1-lab1.inf.besta**', '**c1-lab1.inf.besta.**' diventa '**c1-lab1.inf.besta**').

Il campo '**CNAME**' equivale ad un alias, ogni host può avere più nomi mnemonici. Una forma alternativa è la seguente:

```
www         CNAME     c1-lab1
```

5.6.2 `/etc/bind/db.172.17.1`

```
$TTL 604800
@       IN      SOA     router-lab1.inf.besta. root.router-lab1.inf.besta.  (
                                1997062000 ; Serial
                                28800      ; Refresh
                                7200       ; Retry
                                604800     ; Expire
                                86400 )    ; Minimum
@               NS      router-lab1.inf.besta.
@               MX      10 router-lab1.inf.besta.

1               PTR     c1-lab1.inf.besta.
2               PTR     c2-lab1.inf.besta.
3               PTR     c3-lab1.inf.besta.
4               PTR     c4-lab1.inf.besta.
254\011\011 PTR\011 router-lab1.inf.besta.
```

Il file punta ad uno speciale dominio (‘**`1.17.172.in-addr.arpa`**’) che viene utilizzato dal DNS per il *reverse mapping*, ossia dall’indirizzo numerico trova il nome menemonico corrispondente.

Segue un esempio del file ‘`named.ca`’ che contiene i name servers radice.

```
# file /etc/bind/db.root

; This file holds the information on root name servers needed to
; initialize cache of Internet domain name servers
; (e.g. reference this file in the ‘‘cache  .  &lt;file&gt;’’
; configuration file of BIND domain name servers).
;
; formerly NS.INTERNIC.NET
;
.    3600000  IN  NS  A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.  3600000    A   98.41.0.4
;
; formerly NS1.ISI.EDU
;
.    3600000    NS  B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.  3600000    A  128.9.0.107
;
; formerly C.PSI.NET
(omissis)
; End of File
```

Il file sopracitato è fondamentale e permette di risolvere indirizzi non presenti nella zona di autorità (in poche parole per "uscire verso la rete"), permette di raggiungere i vari name server presenti nel dominio di root (.); non va modificato, ma si può scaricare presso *Internic*.

Una zona può diventare slave per un’altra zona. Invece di crare un file di zona lo si può scaricare da un altro server.

```
# file /etc/bind/named.conf
...
zone "mat.besta" {
  type slave;
  file "bind/db.mat.besta";
  masters { 172.17.2.10;};
...
}
```

Capitolo 6

DHCP (*Dynamic host configuration protocol*)

Il DHCP è un protocollo che consente di passare le informazioni di una configurazione di un nodo in una rete TCP/IP. L'impostazione è ereditata dal protocollo BOOTP (*Bootstrap Protocol*), al quale sono state aggiunte funzionalità per la ricerca di risorse riutilizzabili e nuove possibilità di configurazione.

6.1 Introduzione

Il DHCP è diviso in due parti: la prima che si occupa di spedire le informazioni di un nodo al server del servizio; la seconda che si occupa di fornire le risorse di rete al nodo che le richiede. Il documento così generato viene utilizzato sia dal fornitore di servizio (dove è in esecuzione il server Dhcp), che dall'utilizzatore (dove è in esecuzione il client Dhcp).

L'assegnazione dell'indirizzo di rete da parte del server avviene in tre modi differenti. Il primo modo avviene in automatico (*automatic allocation*), dove il Dhch fornisce in modo permanente un indirzzo IP. Il secondo avviene in modalità dinamica (*dynamic allocation*) dove l'indirzzo di rete viene assgnato solo per un intervallo limitato di tempo. Il terzo avviene in modalità manuale (*manual allocation*) dove l'indirizzo IP viene assegnato dall'amministratore di rete.

Questi metodi possono coesistere entrambi in una rete.

La modalità dinamica è indispensabile quando gli indirizzi IP non sono sufficienti a coprire l'insieme delle risorse richieste. Per questo motivo questi indirzzi possono venire riutilizzati e condivisi, non contemporaneamente, tra più nodi.

Il servizio Dhcp non registra i nomi di dominio nel server di Dns.

6.2 '**Dhcpd**'

Il demone '**Dhcpd**' realizza il servizio Internet Dhcp e Bootp.

Il servizio fornisce, ad un nodo della rete, un indirizzo Ip valido per poter utilizzare i servizi Internet/intranet. Le risorse sono assegnate in modo preventivo dall'amministratore di rete.

Il servizio viene fornito dal demone '**dhcpd**' e l'impostazione generale è la seguente:

`dhcpd` [***options***] [***interfaces***]

L'impostazione generale della sottorete viene gestita dal file di configurazione '`dhcpd.conf`'; le informazioni contenute nel file vengono memorizzate in memoria e quindi ad ogni modifica va riavviato il servizio.

Le richieste a '**Dhcpd**' vengono inoltrate dal cliente '**Dhcpcd**'; ricevuta la notifca il fornitore del servizio garnatisce le risorse per un periodo di tempo stabilito dall'amministratore e che generalmente è predefinito per un periodo di un giorno.

L'attività di assegnazione degli indirizzi Ip viene registrata così, in caso di riavvio del servizio o nel peggiore delle ipotesi di blocco improvviso del sistema, il demone è in grado di riassegnare in modo corretto le risorse. Il file di registro per questo scopo è '`dhcpd.leases`'. Il meccanismo di aggiunta di un nuovo nodo in coda agli altri è assicurato anche in caso di interruzione a metà della scrittura.

6.2.1 '**dhcpd**'

Il demone può essere lanciato da linea di comando o da un file di script. È opportuno fare sempre riferimento alla propria distribuzione per una corretta configurazione del servizio.

Di seguito alcune opzioni.

-p ***port***

Avvia il servizio su un'altra porta differente dalla 67 che è quella predefinita.

-f

Avvia il servizio in primo piano (*foreground*).

-d

Registra le informazioni sullo standard error invece di utilizzare il servizio '**Syslog**'.

-cf ***file_conf***

Analizza un file di configurazione diverso da '`dhcpd.conf`'.

-if ***file_lease***

Registra le informazioni di lease in file diverso da '`dhcpd.lease`'.

-q

Avvia il demone senza stampare nessuna informazione.

-t

Legge il file di configurazione senza avviare il demone. In questo modo si può controllare l'esatta impostazione del file.

-T

Equivalente a '**-t**'; tuttavia legge il file di lease.

-tf ***file_traccia***

Indica un file dove tutte le operazioni interne del demone vengono registrate. Questa impostazione è utile per ricercare eventuali errori nel servizio dovuti ad improvvisi blocchi.

-play ***file_traccia_playback***

Visualizza il file generato con l'opzione '**-tf**'.

Il servizio potrebbe essere così avviato:

```
# dhcpd -q -cf /etc/dhcpd.conf [ Invio ]
```

Se non viene specificata nessuna interfaccia di rete il demone controlla tutte i dispostivi dove è possibile eseguire un broadcast.

6.3 /etc/dhcpd.conf

Il file fornisce al demone '**dhcpd**' le informazioni riguardo ai nodi di rete fornite dal servizio Dhcp. I commenti nel file sono preceduti per riga dal carattere #; sono ammessi spazi unificatori e tabulazioni per la formattazione del testo. Ad ogni dichiarazioni di sottorete, di nodo o di gruppo seguono dei parametri di configurazione.

Se questi parametri sono globali, devono essere inseriti ad inizio del file di configurazione.

Vediamo in dettaglio.

option domain-name "*nome_dominio***"**

Il nome di dominio assegnato dal servizio Dns.

option domain-name-servers ***nome_dominio*****[,** ***nome_domimio*****]**

La lista di server che forniscono il servizio di Dns.

option routers ***indirizzo_ip*** **|** ***nome_dominio***

Specifica o l'indirizzo numerico o il nome del router della rete o della sottorete.

max-lease-time ***periodo***

Indica il tempo massimo, in secondi, di lease (letteralmente "affitto") assegnato ad un nodo.

default-lease-time ***periodo***

Indica il tempo predefinito, in secondi, di lease assegnato ad un nodo.

Ad ogni sottorete bisogna specificare il gruppo di indirizzi coinvolti dal riutilizzo.

Ogni nodo può essere dichiarato all'interno di un gruppo, isolato o appartenente al collettivo. Di seguito un semplice ed intuitivo esempio.

```
subnet 172.17.1.0 netmask 255.255.0.0 {
        range 172.17.1.30 172.17.1.60;
        range 172.17.1.100 172.17.1.250;
\011option broadcast-address 172.17.255.255;
}
```

Il servizio agisce sui nodi della rete limitati dai confini definiti nel parametri di '**range**'. I nodi della rete possono venire raggruppati in gruppi. Ogni nodo dichiarato può essere seguito da parametri diversi.

La dichirazione di un nodo con indirizzo Ip fisso viene eseguita con il parametro '**hardware ethernet**'. Con esso si fornisce l'indirizzo Mac della scheda di rete che è unico. Il parametro '**fixed-address**' assegna l'indirizzo IP.

```
host c1.lab.inf.besta {
        hardware ethernet 00:00:00:00:00:00;
        fixed-address 172.17.1.101;
}
```

Una sottorete può richiedere configurazioni particolari e differenti da quelli globali.

Un router o un server DNS o un dominio stesso possono essere specificati come parametri di una sottorete; così facendo i parametri predefiniti vengono riscritti.

```
subnet 172.17.2.0 netmask 255.255.0.0 {
        option routers 172.17.1.254;
        option domain-name "mat.besta";
\011option domain-name-servers router.inf.besta, router.mat.besta;
        max-lease-time 7200;
        range 172.17.2.30 172.17.2.60;
}
```

6.3.1 Le liste di Pool

Le liste di pool servono per maneggiare una lista di indirizzi diversamente da un'altra lista di indirizzi anche appartenenti allo stesso segmento di rete. Una situazione simile è abbastanza comune: si immagini una lista di indirizzi assegnata a nodi fissi ed un'altra per nodi non fissi; con esse varia anche il tempo di «affitto» di questi.

Dhcp assegna ai nodi di rete un indirizzo Ip quando questi ne fanno richiesta. Con l'opzione '**deny unknown clients**' questo comportamento viene rifiutato quando i nodi non vengono riconosciuti.

```
subnet 172.17.2.0 netmask 255.255.0.0 {
        option routers 172.17.1.254;
\011option broadcast-address 172.17.255.255;

\011pool {
\011       option domain-name "mat.besta";
               option domain-name-servers router.inf.besta, router.mat.besta;
               max-lease-time 7200;
               range 172.17.2.1 172.17.2.100;
\011       deny unknown clients;
        }

        pool {
               option domain-name "mat.besta";
               option domain-name-servers router.inf.besta, router.mat.besta;
               max-lease-time 500;
               range 172.17.2.101 172.17.2.150;
\011       allow unknown clients;
        }
}
```

NFS e NIS

7.1 Configurazione server

Il termine NFS sta per *Network file system*. Questo protocollo permette di distribuire in modo trasparente file attraverso la rete. La struttura del NFS favorisce il trasporto e la portabilità tra macchine differenti con sistemi operativi differenti.

Trova utilizzi in più campi: sia come vero e proprio file system, sia permette di condividere la directory '`/home`' utile se più utenti devono accedere con la stessa login su client diversi.

La macchina server richiede di operare su un unico file di configurazione '`/etc/exports`' in esso andranno inseriti nelle righe e colonne le directory da esportare, e gli host autorizzati; quest'ultimi se indicati come nome di dominio devono essere risolvibili o attraverso file host o DNS.

```
/home          *.inf.besta(rw,no_root_squash)
*.mat.besta(rw,no_root_squash)
/opt/office52  *.inf.besta(ro)                    *.mat.besta(ro)
```

Nella prima riga la directory '`/home`' viene esportata con permessi di lettura e scrittura ('**`rw`**'). Nella seconda riga la directory '`/opt/office52`' viene esportata con permessi di solo lettura ('**`ro`**').

La directory '`/home`' si presta infatti ad essere esportata per consentire agli utenti di leggere e scrivere sul proprio spazio di disco. è consigliabile quindi permettere l'utilizzo in lettura e scrittura della directory. Fino alla distribuzione 6.2 RedHat installava il server WEB apache con la directory '`httpd`' sotto la directory '`/home`'.

Questo sistema ha portato più di qualche problema per chi utilizzava esportare le home directory. Dalla versione 7.0 tale directory è stata spostata in '`/var/www`'.

L'opzione '**`no_root_squash`**' consente agli amministratori ('**`root`**') delle macchine client di poter accedere alla directory con pieni permessi.

7.2 Configuazione client NFS

Dal lato client biosgna solamente modificare il file '`/etc/fstab`' aggiungendo in coda righe come le seguenti:

```
# file /etc/fstab

# Server                    Mount point       Type     options
192.168.2.52:/home          /home             nfs      bg
192.168.2.52:/opt/office52  /opt/office52     nfs      bg,noauto
```

La prima colonna è composta dall'indirizzo IP del server NFS e la relativa directory da esportare. La seconda colonna il mount point relativo al client; la terza colonna il tipo file system da utilizzare, in questo caso l'unica opzione possibile è '**`nfs`**'. L'ultima colonna contiene una serie di opzioni tipo '**`bg`**', che permette il mount in background della directory anche quando manca il segnale di rete (interruzione momentanea del server, calo di tensione negli HUBS/SWITCHS, si stacca un cavo accidentalmente).

Ad ogni avvio del sistema il mount avverrà in automatico, si può comunque forzare l'operazione '`mount/umount`':

```
# umount /opt/office52 [ Invio ]
```

Disinnesto della directory '`/opt/office52`' sul client.

```
# mount /opt/office52
```

Montaggio della directory '`/opt/office52`' sul client.

L'opzione '**`noauto`**' nella colonna opzioni del file '`/etc/fstab`' non forza il mount all'avvio del sistema.

7.3 NIS

Il NIS consente agli host in una rete di condividere file utenti, file passwords e file di hosts. è un servizio client-server e sfrutta l'RPC per raggiungere lo scopo.

Non serve quindi replicare in ogni macchina il file delle password ed il file di hosts: compilato per un host (server), è disponibile per tutti gli altri host (client).

L'unico accorgimento è di ricompilare il database ogni volta che si effettua una modifca sia al file di password (generalmente quando viene aggiunto un utente), sia al file di host ed a qualsiasi altro file gestito dal NIS.

Studieremo, quindi, solo i casi relativi ai seguenti files: '`/etc/passwd`' ed '`/etc/hosts`'.

7.4 Configurazione

Segliere un nome per il dominmio, ex.: '**`nis.azienda`**', inserirlo nel file '`/etc/sysconfig/network`'.

```
# file /etc/sysconfig/network
NISDOMAIN="nis.azienda"
```

Senza dover riavviare i servizi dirette digitare:

```
# nisdomainname nis.azienda [ Invio ]
```

7.4.1 Configurazione server NIS

Installati i relativi pacchetti nel server e nei vari client lanciare nel server il seguente file:

```
# cd /usr/lib/yp [ Invio ]
```

```
[root@low yp]# ./ypinit -m [ Invio ]
  At this point, we have to construct a list of the hosts which will run NIS
  servers.  dell.localdomain is in the list of NIS server hosts.  Please continue
  to add
  the names for the other hosts, one per line.  When you are done with the
  list, type a control D.
    next host to add:  bella.inf.besta
    next host to add:
```

L'opzione '**`-m`**' indica che si sta costruendo un database NIS primario, infatti è possibile anche crearne di secondari, in caso il primario non fosse disponibile per guasti temporanei. Se non ci sono altri server NIS digitare '**`CTRL+D`**' e quindi se quello che avete digitato è corretto, digitate '**`y`**'.

Il tool crea in modo automatico tutti i file di configurazione nel directory '`/var/yp`' e quelle relativi al dominio nis.azienda in '`/var/yp/nis.azienda`'.

Non resta che lanciare il servizio:

```
# service ypserv start [ Invio ]
```

7.4.2 Configurazione client NIS

Per i client l'operazione risulta più semplice. è sufficiente modificare i file di configurazione '`/etc/yp.conf`':

```
# file /etc/yp.conf

domain nis.azienza server 192.168.1.10
```

Riavviare il servizio su ogni host collegato al server NIS:

```
# service ypbind restart [ Invio ]
```

L'utente per cambiare la propria password di accesso non dovrà più utlizzare il comando '`passwd`' che agisce nel file locale bensì '`yppasswd`' che modifica il file delle password del server remoto NIS.

In assenza di DNS posso compilare un un unico file '`/etc/hosts`' nel server NIS, quindi modicare il file '`/etc/host.conf`' di ogni hosts agiungendo l'argomento '**`nis`**' nella prima riga.

```
# file /etc/host.conf

order hosts, bind, nis
```

Con questo sistema viene anche interrogato il file '`/etc/host.conf`' del server remoto NIS dopo quello locale ed un eventuale DNS. (utile se ho un network di indirizzi statici forniti da un ISP il quale mi fornisce anche il servizio di registrazione del DNS del mio dominio sui loro server).

PPP server

I vari programmi per configurare un server per accessi esterni sono reperibili in tutte le distribuzioni: '`mgetty+sendfax`' e '`pppd`'.

Il primo file da modificare è '`/etc/inittab`', aggiungendo una o piu righe (tante quante riservate all'accesso esterno) nel modo seguente:

```
#-- mgetty begin
S0:23:respawn:/sbin/mgetty ttyS0
S1:23:respawn:/sbin/mgetty ttyS1
```

Questo esempio presume che i modem per l'accesso remoto siano collegati sulle porte seriali standard.

Le schede multi seriale in commercio (costose, dotate di processore ottime per sostituire costosissimi dispositivi RAS) possono assegnare i nomi meno comuni per le porte seriali. Supponiamo che esse siano in una quantità di otto e creino i dispositivi ttyM1, ttyM2, ..., ttyM8.

Il file '`/etc/inittab`' è simile al precedente:

```
#-- mgetty begin
M1:23:respawn:/sbin/mgetty ttyM1
M2:23:respawn:/sbin/mgetty ttyM2
...
M8:23:respawn:/sbin/mgetty ttyM8
```

Successivamente bisogna configurare '`mgetty`' perché attivi il '`pppd`' in modo automatico se presente un pacchetto TCP/IP. Questa aggiunta non compromette, comunque, l'accesso in modalità terminale.

```
# file /etc/mgetty/login.config
/AutoPPP/  -  a_ppp  /usr/sbin/pppd 38400 modem auth -chap +pap login crtscts
lock
```

L'opzione '**`login`**' confronta il database delle password di sistema con quella spedita durante la negoziazione se non c'è nessuna corrispondenza chiude la connessione. è altresì possibile agire direttamente sul file '`pap-secrets`' per impedire l'accesso ad utenti privilegiati come '**`root`**'.

Se si utilizza questa soluzione si deve necessariamente rinominare il file '`/etc/ppp/pap-secrets`' oppure inserire tutti gli utenti abilitati al servizio in quest'ultimo. Contrariamente la connessione viene interrotta.

Il file '`/etc/ppp/pap-secrets`' sarà così composto:

```
# file /etc/ppp/pap-secrets

# Utenti disabilitati
guest   *  "*"  -
master  *  "*"  -
root    *  "*"  -
support *  "*"  -
stats   *  "*"  -

#Utenti abilitati
utente1  *  ""  *
utente2  *  ""  *
```

Per consentire a tutti gli utenti con accesso al sistema di accedere anche con servizio PPP si deve togliere nel file precedente l'opzione '**`+pap`**' e sostituirla con '**`-pap`**'. Il file '`/etc/pap-secrets`' non verrà letto, ricordarsi comunque di non cancellarlo.

Per ogni porta seriale collegata al sistema si aggiunge un file di '`/etc/ppp/options.ttySx`'utilizzato per quella connessione (con x indichiamo il numero della porta seriale, quelle standard sono ttyS1 e ttyS2).

```
# file /etc/ppp/options
```

```
# (vuoto)
```

Per ogni porta seriale abilitata all'accesso remoto:

```
# file /etc/ppp/options.ttySx

192.168.1.1:192.168.1.10
dns-addr 192.168.1.1
```

L'ultima riga assegna l'indirizzo IP del server DNS alle macchine WinXX. La penultima riga esplicita l'indirizzo locale (quello del server) e remoto (quello assegnato all'host).

Lo script seguente viene lanciato dopo la avvenuta connessione.

```
# file /etc/ppp/ip-up o ip-up.local

#Questo file si avvia ad ogni esecuzione del comando pppd
#!/bin/sh
#Lo script è chiamato con i seguenti argomenti
#Arg  Name    Example
#$1  Interface name  ppp0
#$2  The tty    tyS1
#$3  The link speed  38400
#$4  Local IP number  12.34.56.78

PATH=/usr/local/sbin:/usr/sbin:/sbin:/usr/local/bin:/usr/bin:/bin
export PATH
sendmail -q &
```

L'aggiunta di '`sendmail -q`' consente di inoltrare tutta la posta eventualmente presente nella cache; il '`sendmail`' è in grado di controllare la cache a scadenze regolari e spedire la posta se presente, ma questo provocherebbe continue connessioni. Vedremo, successivamente, come configurare '`sendmail`' durante il boot del sistema, perché inoltri la posta solo durante la connessione verso il server provider.

8.1 Accesso distinto

Se desideriamo scindere gli accessi esterni da quelli interni (telnet, ftp, ...) si deve procedere in modo sostanzialmente diverso. Prima di tutto si cancella l'opzione '**`login`**' nel file '`/etc/mgetty+sendfax/login.config`'. è sufficiente quindi inserire riga per riga nel file '`/etc/ppp/pap-secrets`' gli utenti e le rispettive password.

```
# file /etc/ppp/pap-secrets
guest    *  "*"    -
master   *  "*"    -
root     *  "*"    -
support  *  "*"    -
stats    *  "*"    -

utente1  *  <password>  *
utente2  *  <password>  *
```

Le righe con (-) impediscono l'accesso all'utente (utile se è utilizzato in congiunzione con l'opzione '**`login`**').

PPPoE (*PPP over Ethernet*)

Il PPPoE è un protocollo che consente di mantenere una connessione PPP attiva su una rete di tipo Ethernet. Il sistema di funzionamento consente ad una traccia PPP di insere incapsulata in una traccia di tipo Ethernet. La fornitura di accesso con linee ADSL si basa in buona parte su questo protocollo.

L'accesso avviene in modo sistematico con uno scambio di informazioni tra il nodo che spedisce una richiesta broadcast per identificare se esiste un identificatore di accesso che gli possa garantire la connessione. Questi pacchetti vengono chiamati PADI (*PPPoE Active Discovery Initiation*). A questo l'ultimo il concentratore di accesso risponde con un pacchetto di tipo PADO (*PPPoE Acrive Discovery Offer*).

La fase si conclude ed incomincia una richiesta di connessione da parte del nodo; i pacchetti sono di tipo PADR (*PPPoE Active Discovery Request*). I pacchetti di risposta del concentratore di accesso sono di tipo PADS (*PPPoE Active Discovery Session-Confirmation*).

L'accesso si conclude con la fase di sessione tra il nodo ed il concentratore di accesso. Le tracce dei pacchetti PPP vengono incapsulati sulle tracce di quelli Ethernet.

9.1 Pppoe

Sintassi

`pppoe -A` [***opzioni_pppoe***]

`pppd pty "pppoe` [***opzioni_pppoe***]`"` [***opzioni_pppd***]

Pppoe è la parte client che consente di attivare una connessione PPPoE. Il programma che viene distribuito è '**`pppoe`**'.

La prima forma viene utilizzata solamente per ricevere informazioni concernenti ai concentratori di accesso e non deve essere impiegata per la connessione.

Alcune opzioni

Opzione	Descrizione
-I ***interfaccia***	Indica il nome della interfaccia di rete da utilizzare che deve essere attiva prima della connessione. Generalmente viene identificata con '**`eth0`**' o '**`eth1`**'.
-T ***timeout***	Il tempo in secondi prima che la connessione termini in assenza di traffico. Questo valore è generalmente quattro volte quello di '**`lcp-echo-interval`**'.
-D ***nome_file***	Memorizza le informazioni sul traffico generato nel file di registro ***nome_file***.
-V	Visualizza il numero di versione di '**`pppoe`**' e termina il programma.
-A	Questa opzione non deve essere utilizzata in congiunzione con il demone '`pppd`'. Vengono inviati dei pacchetti PADI e viene stamapato il nome dei concentratori di accesso ad ogni pacchetto PADO ricevuto.
-S ***nome_servizio***	Specifica il nome del servizio desiderato. In questo modo la connessione avviene solo con i concentratori di accesso che possono garantire il servizio.
-C ***nome_concentratore***	Specifica il nome del concentratore di accesso se questo è noto.
-U	Identifica con un valore unico legato al nodo i pacchetti di ricerca; questo facilita l'impiego di PPPoE quando ci sono più servizi contempoaranei attivi nello stesso istante.
-s	Sincronizza i pacchetti PPP incapsulati da PPPoE. Questo meccanismo implica l'opzione '**`sync`**' nel servizio '**`pppd`**'. Il carico complessivo nel nodo potrebbe diminuire, tuttavia in macchine più lente potrebbe causare la perdita di pacchetti.
-m ***MSS***	Taglia il valore del segmento TCP al valore specificato in ***MSS***. Il valore è generalmente più basso di quello Ethernet a causa del sovraccarico generato da PPPoE. Se il computer funge da router per una rete interna è preferibile utilizzare il valore di 1412 che garantisce un piccolo guadagno in velocità.

Opzione	Descrizione
-p ***file***	Memorizza il PID del processo PPPoE nel file ***file***.
-e ***sess:mac***	Salta la fase di ricerca dell'identificatore di accesso passando direttamente a quella di sessione. Non utilizzare questa caratteristica a meno che non si stia configurando un servizio PPPoE-Server.
-n	Apre un socket di ricerca. Non utilizzare questa caratteristica a meno che non si stia configurando un servizio PPPoE-Server.
-k	Termina una sessione attiva inviando una finestra PADT.
-d	Invia una ricerca, quindi stampa le informazioni ricevute ed esce dal servizio.
-h	Visualizza le informazioni sulla versione di PPPoE.

9.1.1 Esempi pratici

 # pppd pty "pppoe -I eth 0 -T 80 -m 1412" noipdefault defaultroute hide-password lcp-echo-interval 60 lcp-echo-failure 3 noauth mtu 1492 usepeerdns user *userid* [*Invio*]

La ***userid*** dovrà essere memorizzato con la relativa password nel file '`/etc/ppp/chap-secrets`' e '`/etc/ppp/pap-secrets`'. Generalmente ***userid*** e password sono fornite dal gestore di accesso.

```
# file /etc/ppp/chap-secrets

utente      *       password
```

Il servizio Dns viene assegnato dal provider ('**`usepeerdns`**'); se si disponde di un risolutore interno togliere questa opzione.

Nelle distribuzioni GNU/Linux il servizio PPPoE è distribuito con script di installazione e configurazione che evitano l'avvio automatico.

Capitolo 10

DIALD (Dial on demand)

10.1 Introduzione

L'utility in esame non ha mai riscosso grande successo, questo è dovuto alla scarsa documentazione esistente e alla difficile implementazione delle tecniche di "filtraggio". La sperimentazione mi è costata molti giorni di prova, un esame molto approfondito dei file di log, qualche arrabbiatura, decine di reset e qualche scatto telefonico...

Esiste infatti incompatibilità tra la presenza contemporanea del servizio '`named`' e '`diald`'.

Non può, inoltre, essere sottovaluto che in un server a cui è anche delegato il compito di gestire posta, web, samba, DNS ed altro ancora provochi, per presenza di pacchetti TCP, UDP lungo la LAN, (come un client Windows 95 che cerca il server WINS) far partire improvvisamente il collegamento (anche con scadenza di pochi minuti).

Solo un lavoro certosino ha permesso di risolvere questi problemi e evitarne i danni.

10.2 /etc/diald/diald.conf

```
# file /etc/diald/diald.conf

include /etc/diald/phone.filter

device /dev/modem
speed 115200
mode ppp

local 19x.18x.22x.x
remote 19x.18x.22x.x

defaultroute
modem
crtscts

connect "/usr/sbin/chat -v -f /etc/ppp/chatscript"
redial-timeout 5
fifo /etc/diald/diald.ctl
```

File di configurazione ottimizzato per l'accesso con un terminarore ISDN. Le righe sono state modificate o aggiunte rispetto al file che si installa con la distribuzione. Da notare la prima riga, indispensabile se si implementa un servizio DNS nel server. Il file di '`/etc/ppp/options`' non deve contenere argomenti già implementati tipo: '**`connect, defaultroute, proxyarp`**' (da non usare con il DNS), '**`crtscts...`**'

Chairiamo che '**`local`**' e '**`remote`**' indicano i numeri IP del server locale ed il numero del server remoto rispettivamente. Se non si possiede un numero IP statico allora si ricorre ad una scorciatoia:

```
# file /etc/diald/diald.conf
...
local 192.168.1.1
remote 192.168.1.2
dynamic
...
```

Alla macchina locale si assegna il numero fittizio 192.168.1.1 a quella remota il numero 192.168.1.2 ed in più si aggiunge l'opzione '**`dynamic`**'.

Il passo impegnativo consiste nell'ispezionare ed eventualmente modificare il file '`phone.filter`' respon-

sabile delle politiche e dei tempi di connessione, quello che seguirà ha subito qualche modifica per evitare continui collegamenti quando un client WinXX cercava un altro client presente nella stessa LAN (host XXX che cerca host YYY) oppure un server WINS.

```
# REGOLE PER I PACCHETTI TCP/IP
# Aspetta 60 secondi per la connessione, se l'host non risponde fai cadere il
collegamento
accept tcp 60 tcp.syn

# Il named-xfer non fa partire la connessione;
ignore tcp tcp.dest=tcp.domain
ignore tcp tcp.source=tcp.domain

# Per evitare che il netbios attvi la connessione
ignore tcp tcp.source=tcp.netbios-ns,tcp.dest=tcp.netbios-ns

# I pacchetti vuoti fanno comunque cadere il collegamento (<40 byte)
ignore tcp ip.tot_len=40,tcp.live

# Il collegamento http resta attivo almeno 2 minuti
ignore tcp !tcp.live,tcp.dest=tcp.www
ignore tcp !tcp.live,tcp.source=tcp.www

accept tcp 120 tcp.dest=tcp.www
accept tcp 120 tcp.source=tcp.www

# Una volta che il link non è più attivo, si chiude la
connessione. Se non c'è nessuna connessione ignora completamente la
regola (keepup)
keepup tcp 30 !tcp.live
ignore tcp !tcp.live

# Taffico ftp
accept tcp 120 tcp.dest=tcp.ftp
accept tcp 120 tcp.source=tcp.ftp

# Non indispensabile in quanto non è definito in /etc/services
accept tcp 120 tcp.dest=tcp.ftp-data
accept tcp 120 tcp.source=tcp.ftp-data

# Tieni il collegamento comunque acceso per 5 minuti
accept tcp 300 any

# REGOLE PER I PACCHETTI UDP
# Ignora rwho
ignore udp udp.dest=udp.who
ignore udp udp.source=udp.who

# Ignora RIP
ignore udp udp.dest=udp.route
ignore udp udp.source=udp.route

# Ignora NTP e timed
ignore udp udp.dest=udp.ntp
ignore udp udp.source=udp.ntp
ignore udp udp.dest=udp.timed
ignore udp udp.source=udp.timed

# Attiva il collegamento quando si è in presenza contemporanea di due
servizi named
accept udp 60 udp.dest=udp.domain,udp.source=udp.domain

# Ignora le richieste provenienti da netbios
ignore udp udp.dest=udp.netbios-ns
ignore udp udp.source=udp.netbios-ns

# Ignora routed e gated
ignore udp tcp.dest=udp.route
```

```
ignore udp tcp.source=udp.route

# Dai ai pacchetti ICMP 60 secondi
accept icmp 60 any
```

Le variabili sono dichiarate in un altro file: ‘`diald.defs`’.

CallBack

11.1 Introduzione

Il Callback consente ai dipendenti che lavorano in remoto (generalmente sul server) di farsi richiamare dai modem del Server aziendale.

Esistono costose apparecchiature hardware (RAS) che consentono la stessa cosa, però anche con la soluzione qui proposta si possono raggiungere discreti risultati con costi contenuti.

I pacchetti utilizzati al solito: '`mgetty`' e '`pppd`'.

11.2 Configurazione

Modificare il file '`/etc/mgetty/login.config`' ed inserire in coda tante righe quanti sono gli utenti abilitati al servizio di callback.

```
# file /etc/mgetty/login.config
...
call_utente1    -   -     /usr/sbin/callback -x3 -S 0422xxxxx
call_utente2    -   -     /usr/sbin/callback -x3 -S 0422yyyyy
call_generic    -   -     /usr/sbin/callback -x3 -S
```

L'opzione '**`-x`**' attiva il livello di dettaglio del debug, i valori consentiti variano da 0 a 5.

L'opzione '**`-S`**' utilizza la stesso modem che ha risposto per la richiamata verso l'utente.

L'utente "call_generic" non ha numero di telefono quindi il callback chiederà di inserirne uno che utilizzarà per la richiamata.

Per avvalersi del servizio è sufficiente digitare al login, ovviamente utilizzando un collegamento via modem, uno di questi utenti.

> Durante il callback la comunicazione rimane comunque client-server verso un unico verso, ossia il computer che ha chiamato fungerà comunque da client, cambia solo lo scatto sulla bolletta... Si trasferisce dal dipendente all'azienda!

Un altro file di configurazione '`/etc/mgetty/callback.config`' imposta i parametri del software di callback.

```
# file /etc/mgetty/callback.config

# Dispositivi seriali abilitati al callback
dialout-devices ttyM0:ttyM2

# Tempo (in secondi) che intercorre tra due tentativi di chiamata
retry-time 30

# Tempo (in secondi) che intercorre per la prima chiamata
delay 10

# Tempo (in secondi) massimo per la chiamata
max-time 90

# Stringa di inizializzazione del modem
modem-init "" ATZ

# Velocità  della porta seriale
```

```
speed 115200

# Livello di debug
debug 4

# Prefisso per la chiamata
dial-prefix ATDT1055
```

11.2.1 Callback via terminale

Attivare il software '`minicom`', digitare la stringa '**`ATDT0422xxxxxx`**'.

```
Welcome to minicom 1.83.1

Press CTRL-A Z for help on special keys

ATZ
OK
ATDT0422123456
CONNECT

RedHat Linux release 7.0 (Guinnes)
Kernel 2.2.16-22smp on a i686
login: call_utente1
```

Al prompt digitare uno degli utenti abilitati al callback: impostati nel file '`/etc/mgetty/login.config`'.

Digitare '**`ALT-H`**' (Hang Up), resettare il modem con '**`ATZ`**', quindi digitare '**`AT&C1A`**' ed attendere lo squillo del telefono.

Alla risposta del vostro modem apparirà il prompt del server remoto. Entrare come utente registrato.

11.2.2 Callback via PPP

La procedura non cambia molto rispetto alla precedente, si attiverà in questo caso una connessione PPP.

Creare un paio di file di scripts '`/etc/ppp/peers/callback-ppp`' per il servizio '`pppd`' e '`/etc/ppp/chat-ppp`' per il programma '`chat`'.

Di seguito due semplici ed utili esempi:

```
# file /etc/ppp/peers/callback-ppp
115200
/dev/modem
connect "/usr/sbin/chat -v -f /etc/ppp/chat-ppp"
debug
defaultroute
noauth
user pincopallino

# file /etc/ppp/chat-ppp
ABORT BUSY
ABORT VOICE
ABORT "NO DIALTONE"
ABORT "NO ANSWER"
" ATZ
OK ATDT0422xxxxxx
CONNECT \d\d
ogin: \q\dcall_utente1
callback: \q0422xxxxx
TIMEOUT 90
RING AT&C1A
CONNECT
```

Attivare la connessione con:

pppd call callback-ppp [*Invio*]

L’autenticazione per l’utente pincopallino avviene dopo l’utlimo ‘**CONNECT**’.

FAX

Mgetty è stato costruito per essere usato con modem/fax compatibili hayes. Le sue funzionalità sono estese all'inizializzazione del modem, la risposta e l'Invio di fax. Quest'ultima caratteristica verrà spiegata in questa sezione.

12.1 File `/etc/mgetty+sendfax/mgetty.config'

File di configurazione per 'mgetty' consente di inizializzare il modem.

```
# file /etc/mgett+sendfax/mgetty.config

# imposta il livello globale di debug a 4
debug 4

# imposta l'id del fax locale
fax-id 39 0422 xxxxx

# imposta la velocità della porta seriale
speed 38400

# configurazione del modem collegato alla porta ttyS0
# utile in presenza di dispositivi che richiedono stringhe di configurazione
specifiche
port ttyS0
  debug
  init-chat "" ATZ OK
  data-only y
```

12.2 File `/etc/mgetty+sendfax/faxspool.rules'

Nell'installazione di mgetty questo file viene memorizzato con '`/etc/mgetty+sendfax/faxspool.rules.sample`'; per prima cosa digitare:

```
[root@low /etc/mgetty+sendfax]#cp faxspool.rules.sample faxspool.rules
```
[*Invio*]

Modificare quindi il file.

```
# file /etc/mgetty+config/faxspool.rules

# file name suffix -> file format mapping
# after the SUFFIX key word, name the extention, and then the file format:
#
# the following mappings are already built-in into faxspool:
#
SUFFIX .g3 g3
SUFFIX .asc ascii
SUFFIX .txt ascii
SUFFIX .ps ps
SUFFIX .lj lj
SUFFIX .pcl lj
#

# file format -> conversion command rules
#
# $1 is set to the input file name, $2 is the base name for the
# output file(s), and $3 is "-n", if normal resolution has been requested
#
```

```
# Warning: better use absolute path names here, as "faxspool" might be
#          called without a reasonable value for $PATH
#
#
# Again, some examples that are similar to what is already built in:
#

#
# postscript files (using faxg3 driver)
FILTER ps /usr/bin/gs -sDEVICE=faxg3 -r204x198 -sOutputFile=$2%03d -dNOPAUSE -q
-dSAFER - <$1

#
# ASCII files, filter via ghostscript
FILTER ascii /usr/bin/gs -sDEVICE=dfaxhigh -sOutputFile=$2%03d -dNOPAUSE -q --
gslp.ps -fCourier-Bold10 -B $1

#
# G3 files are just filtered through "g3cat" to clean errors
FILTER g3 /usr/bin/g3cat $1 >$2.001
```

Ad ogni suffisso corrisponde un filtro, l'utente più smaliziato può configurarsene di propri. Non è necessaria la configurazione e la presenza di questo file.

12.3 File `/etc/mgetty+sendfax/sendfax.config`

Sendfax invia fax in formato g3 con faxmodem in classe 2.

```
# file /etc/mgetty+sendfax/sendfax.config

# imposta il dettaglio del log
verbose y

debug 5

# imposta una o più linee di uscita per i fax
fax-devices ttyS0
#fax-devices ttyS0:ttyS2

# imposta il numero di fax da inviare al destinatario
fax-id 39 0422 xxxxx

# prova ad inviare la pagina per tre volte e continua anche se l'ultima fallisce
max-tries 3
max-tries-continue y

# come per mgetty se alcuni modem richiedono configurazioni particolari
#port ttyS0
#  modem-handshake AT&K4

# il modem non lavora correttamente con classe 2.0, prova quindi con classe 2
port ttyS1
  modem-type cls2
```

12.4 File `/etc/mgetty+sendfax/fax.allow'

Nel file '`/etc/mgetty+sendfax/fax.allow`' inserire gli utenti abilitati all'Invio del fax, nel file '`/etc/mgetty+sendfax/fax.deny`' gli utenti non abilitati.

```
# file /etc/mgetty+sendfax/fax.allow

umbertoz
toni
```

12.5 Preparazione ed invio del fax

Conclusa questa breve fase di configurazione è possibile inviare fax, l'operazione si conclude in due semplici passaggi. Nel primo i fax vengono convertiti dal formato originario (.txt, .ps) nel formato .g3, nel secondo inviati agli utenti.

faxspool -F "Umberto Zanatta" 0422xxxxx file1.txt [*INVIO*]

```
spooling to /var/spool/fax/outgoing/F000002...
spooling file1.txt...
conegliano.txt is format: ascii
GNU Ghostscript 6.51 (2001-03-28)
Copyright (C) 2001 artofcode LLC, Benicia, CA.  All rights reserved.
This software comes with NO WARRANTY: see the file COPYING for details.
Loading NimbusMonL-Regu font from /usr/share/ghostscript/fonts/n022003l.pfb...
2279788 828969 1622424 332422 0 done.
Loading NimbusMonL-Bold font from /usr/share/ghostscript/fonts/n022004l.pfb...
2340076 890731 1622424 338024 0 done.
Printing file1.txt
Page height = 70.

Putting Header lines on top of pages...

Fax queued successfully. Will be sent at next ''faxrunq'' run.
```

Se l'operazione termina correttamente non resta che spedirlo con il comando '`faxrunq`'.

faxrunq [*Invio*]

```
processing F000002/JOB...
/usr/sbin/sendfax -v 0422xxxxx f1.g3 f2.g3 f3.g3
Trying fax device '/dev/ttyS0'... OK.
Dialing 0422xxxxx...
```

La coda dei fax può essere visualizzata con il comando '`faxq`'.

faxq [*Invio*]

```
F000002/JOB: queued by umbertoz. 3 page(s) to 0422xxxxx.
```

Per eliminare un fax in coda controllare l'id del job quindi digitare (supponiamo che l'id del job sia F000002):

faxrm F000002 [*Invio*]

Capitolo 13

Server WEB

Breve panoramica sul servizio più conosciuto e soprattutto quello che, a mio parere, ha permesso a LiNUX di crescere sia in prestazioni che in popolarità come sistema operativo per Internet: il server WEB.

è prassi considerare Apache "Il server WEB", è molto diffuso ed ha conquistato una buona fetta di mercato che sfiora o supera il 60% di tutte le installazioni. è un software multipiattaforma facile da configurare, tuttavia presenta molti limiti: più per le prestazioni, mentre la stabilità è a dir poco, ottima.

Non è infatti gestito a threads bensì lancia più copie del servizio all'aumentare delle richieste, RedHat sembra intenzionata a lanciare un nuovo Web Server di nome TUX, che si appoggia ai servizi HTTP presenti nel kernel 2.4.x.

Viene integrato con molti moduli, tra quelli più importanti ricordiamo: '**`mod_php, mod_dav, mod_perl, mod_ssl.`**'

Supporta i Virtual Host, l'esecuzione dinamica di script cgi (quest'ultimi in origine erano monopolio del Perl, ora per gli applicativi WEB si preferiscono linguaggi di script evoluti come il PHP e l'ASP).

"Il Perl è morto? Viva il Perl."

Varieremo di poco il file di configurazione per rendere accessibile le pagine WEB create dagli utenti, per creare domini virtuali, per scaricare i Log di sistema in un database SQL (PostgreSQL o MySQL, ma in generale tutti i DataBase in standard SQL).

13.1 Apache 1.3.xx

Il file di configurazione è '`httpd.conf`' e risiede nella directory '`/etc`', tuttavia quest'ultima può variare in base alla distribuzione.

Le pagine WEB risiedono nella directory '`/var/www/html`' per il nuovo filesystem standard, tuttavia anche in questo caso possono sussistere differenze.

Ad ogni modifica del file di configurazione il servizio dovrà essere riavviato.

Per una distribuzione RedHat:

`service httpd restart` [*Invio*]

Per una distribuzione SuSE:

`rcapache restart` [*Invio*]

13.2 File '`httpd.conf`'

Di seguito alcune direttive presenti nel file di configurazione, quelle che generalmente vengono modificate per ottimmizzare l'uso del servizio o per aggiungere domini, directory, permessi di accesso, percorsi per le immagini ed icone, scripts e quant'altro. Ogni linea verrà spiegata con un commento e con un esempio completo. Le direttive sono inserite in contesti, generalmente directory, file di configurazione del server o domini virtuali.

```
# file /etc/httpd/conf/httpd.conf

ServerAdmin webmin@inf.besta
ServerName www.inf.besta

DocumentRoot "/var/www/html"

Listen 192.168.2.53:80
Listen 192.168.2.54:80
```

```
Alias  /img/  /var/www/html/img/
ScriptAlias /cgi-bin/ /var/www/html/cgi-bin/

<Directory /var/www/html/test>
 Options Indexes Includes FollowSymLinks
 AllowOverride AuthConfig
 Order allow, deny
 Allow from all
</Directoty>
```

Tabella 13.1. Direttive Apache.

Direttiva	Contesto	Commento
ServerAdmin	Server conf, Virtual Host	Indirizzo email del responsabile del server WEB (il webmaster).
ServerName	Server conf, Virtual Host	Il nome dell'host, l'indirizzo (se non si dispone di un nome di dominio vali
DocumentRoot	Server conf, Virtual Host	Il percorso dove sono memorizzati tutti i documenti, o meglio le pagine HT
BindAddress *|indirizzo|dominio	Server conf	Se l'argomento è "*" il server web è in ascolto su ogni indirizzo della macc
Listen [indirizzo:]porta	Server conf	Gli indirzzi IP e la porta dove il serer è in ascolto. Per default il server asco
Port number	Server conf	La porta standard per il servizio HTTP è la 80. In assenza della direttiva Lis
Alias	Server conf, Virtual Host	Consente ai documenti di essere posizionati in altre parti del filesystem oltr
ScriptAlias	Server conf, Virtual Host	Identico comportamento assunto da Alias, in questo directory vengono mer
<Directory></Directory>	Server conf, Virtual Host	Raccoglie un gruppo di direttive che agiscono esclusivamente in questo dire
<Location></Location>	Server conf, Virtual Host	è simile alla precedente, e viene processata dopo quest'ultima. Il percorso r

13.3 Autenticazione utenti con Apache

Un directory può avere restrizioni d'accesso a determinati utenti o gruppi. è sufficiente creare il file '**`.htaccess`**' all'interno del directory stesso.

Prima cosa modificare il file '**`/etc/httpd.conf`**' nel seguente modo:

```
# file /etc/httpd/conf/httpd.conf

<Directory "/var/www/html">

#
# This may also be "None", "All", or any combination of "Indexes",
# "Includes", "FollowSymLinks", "ExecCGI", or "MultiViews".
#
# Note that "MultiViews" must be named *explicitly* "OptionsAll";
# doesn't give it to you.
#
    Options Indexes Includes FollowSymLinks

#
# This controls which options the .htaccess files in directories can
# override. Can also be "All", or any combination of
"Options", "FileInfo",
# "AuthConfig", and "Limit"
#
    AllowOverride AuthConfig

#
# Controls who can get stuff from this server.
#
    Order allow,deny
    Allow from all
</Directory>
```

La linea con '**AllowOverride None**' deve essere cambiata in '**AllowOverride AuthConfig**'. Ora con l'utilizzo del programma '**htpasswd**' è possibile creare un database di utenti con relativa password da poter utilizzare per i nostri accessi.

```
# htpasswd -c /etc/http/.htpasswd webmin Invio
New Password:
Re-type new password:
Adding password for user webmin
```

L'opzione '**-c**' crea il file se non esiste, deve essere usato solamente la prima volta.

13.3.1 File '**.htaccess**'

Supponiamo di limitare l'accesso al directort '**/var/www/html/admin**' al solo utente '**webmin**'.

```
# file /var/www/html/admin/.htaccess

AuthName "Admin web"
AuthType Basic
AuthUserFile /etc/httpd/.htpasswd
Require user webmin
```

Oltre all'utente webmin potrei aggiungerne altri oppure sostituire il tutto con la riga '**Require valid-user**'. Ogni utente presente nel file delle password può accedere in quella zona riservata del sito.

Capitolo 14

Proxy server

Il proxy server è un servizio che, generalmente, integra o sostituisce la firewall e permette un accesso indiretto della proprio network verso Internet. Il client quindi comunicherà direttamente con il proxy e questo a sua volta con il router. Il proxy è indispensabile nelle reti dove non sia presente un mascheramento delle sottoreti, e permette di monitorare il traffico dei client verso la rete, il file di log può essere analizzato da prodotti come webalizer. La disponibilità di configurare un'ampia cache su disco, riduce notevolmente il traffico generato dai client ed abbassa i costi di connessione. Le capacità di firewall per questi prodotti o sono disponibili già nel nucleo del prodotto o vengono aggiunte tramite plug-in commerciali e non.

14.1 Squid

Squid è un proxy server ad alte prestazioni che supporta i protocolli HTTP, FTP e gopher. Squid consiste in due programmi distinti: squid il server principale e dnsserver che funge da Domain name system. Il numero di processi dnsserver eseguiti è configurabile per raggiungere la piena efficienza secondo le necessità. I protocolli SSL/HTTPS/TLS sono pienamente supportati in modalità tunneling tra client e server.

Tabella 14.1. Argomenti file di configurazione di Squid.

Opzione	Valori possibili/Esempio	Commento ed uso
'**`http_port`**' port	3128	Quale porta utilizzare per le richieste HTTP. N
port	3130	Il numero dell porta dove Squid invia le richie
'**`tcp_outgoing_address`**' indirizzo	10.1.2.3	Usato per le connessioni ad un server remoto
udp_incoming_address indirizzo	10.1.2.3	Usato dal socket ICP per ricevere i pacchetti d
udp_outgoing_address indirizzo	10.1.2.3	Usato per i pacchetti ICP spediti ad altre cach
cache_peer hostname type http_port icp_port [option]	proxy1.inf.besta parent 3128 3130	Permette di specificare altri proxy server nella
cache_peer_domain cache-host domain [domain ...]	proxy.inf.besta .besta	Usato per limitare le richieste alla cache. In qu
neighbor_timeout	6	Quanto tempo attendere prima di cercare l'inf
cache_stoplist	cgi-bin ?	Oggetti che non devono mai essere memorizz
cache_mem mem	4 MB	Quanta memoria riservare alle pagini in transi
maximum_object_size mem	4096 KB	Oggetti che superano questa dimensione non v
minimun_object_size mem	2048 KB	Oggetti inferiori a questa dimensione non ven
cache_dir type directory name MBytes level-1 level	ufs /var/spool/squid 2048 16 256	è possibile inserire più righe cache_dir per dis
emulate_httpd_log on|off	on	La cache emula i file di log tipo di un server h
httpd_accel_host hostname	www.inf.besta	Attiva lo httpd accelerator del server specifica
httpd_accel_port port	81	La porta dove il server httpd reale è in ascolto
httpd_accel_with_proxy on|off	on	Attivare http_accel_host disattiva la cache. Se

14.2 Access control list

La direttiva '**`acl`**' definisce un tipo di accesso. Si possono abilitare gruppi di utenti, gruppi di numeri IP, orari di accesso, protocolli e così via.

Consente due forme utilizzabili:

```
#file /etc/squid.conf

acl aclname acltype stringa
acl aclname acltype "file"
```

Se si usa file gli argomenti vanno inseriti uno per linea.

La situazione più comune che si presenta sia a scuola che in azienda è consentire l'accesso alla cache solo alle macchine e agli utenti autorizzati.

Prima di iniziare bisogna attivare un programma per l'autenticazione esterna. Se si vuole separare l'am-

ministrazione degli utenti web da quella UniX, è propensivo l'uso del tradizionale nsca inserire nella riga corrispondente:

```
#file /etc/squid.conf

...
authenticate_program    /usr/sbin/ncsa_auth
/usr/local/etc/squid/passwd
...
```

Il file contenente le password deve essere generato con l'utility '`htpasswd`' fornita con il server web apache, tuttavia se gli utenti UniX sono anche quelli web è preferibile la riga seguente:

```
#file /etc/squid.conf

...
authenticate_program    /usr/sbin/pam_auth              /etc/passwd
...
```

Non resta che aggiornare le acl. Per essere subito operativi aggiungere nelle acl la riga seguente:

```
#file /etc/squid.conf

...
acl     password        proxy_auth                      REQUIRED
acl     secure_users    proxy_auth                      root bin nobody
...
```

Nella seconda si possono aggiungere tutti gli utenti critici.

Per abilitare solamenente i computer o i numeri IP della propria rete aggiungere nelle acl una riga tipo la seguente e sostituire con il vostro numero di rete:

```
#file /etc/squid.conf

...
acl     network1         src
192.168.1.0/255.255.255.0
acl     network2         src
192.168.2.0/255.255.255.0
...
```

Non resta che attivare le politiche; e negare tutto ciò che non è stato diversamente indicato ('**`http_access deny all`**').

```
#file /etc/squid.conf

...
http_access             deny                            !password
http_access             deny                            secure_users
http_access             allow                           localhost
http_access             allow                           network1
http_access             allow                           network2
http_access             deny                            all
...
```

Le acl vengono verificate partendo dalla prima, per questo una volta autenticato l'utente si deve verificare che la macchina client chiamante sia compresa nelle lista di accesso autorizzata.

14.3 Configurare i Browser

Tutti i *browser* in commercio possono essere configurati per accedere al proxy server.

Il *proxy server* può agire in modo distinto sui servizi *HTTP, FTP, GOPHER*. Gli ultimi non sono molto usati, mentre il *caching* delle pagine *WEB* è certamente più utile. Mettere in cache un file scaricato da un utente via *ftp* può essere rischioso se l'operazione non è stato portata a termine in modo completo.

Premesso questo è sufficiente selezionare il menu delle opzioni e nelle *Proprietà->Avanzate* (Communicator) o nella modalità di *Connessione* (Internet Explorer) scegliere il menu relativo al proxy. Inserire quindi l'indirizzo numerico dell'*host* che ospita il proxy (ex.: 10.0.1.1, 192.168.1.1, ...) e nella casella affianco la porta. Quest'ultima generalmente è la 8080, ma nel caso di Squid è la 3128.

Con *lynx* basta entrare nel file di configurazione '**`/etc/lynx.cfg`**' e modificare le righe seguenti:

```
...

http_proxy:http://10.0.1.1:8080/
https_proxy:http://10.0.1.1:8080/
ftp_proxy:http://10.0.1.1:8080/
gopher_proxy:http://10.0.1.1:8080/
...
```

14.3.1 Squid e Transparent Proxying

Il *transparent proxying* è già stato trattato in precedenza, ci limiteremo quindi a spiegare le modifiche al file di configurazione di Squid per rendere attivo la cache; ricordo che sempre nel paragrafo relativo al *transparent proxying* è stato motivato il perché di questa scelta.

Questa tecnica è molto valida e, prima vista, più veloce di quella precedente pur tuttavia con qualche limitazione,comunque per me ininfluente.

1. Prima di tutto nel file di log '`/var/log/squid/access.log`' non verrano più riportati gli URL bensì il numero IP; questo perché l'indirizzo è determinato mediante una chiamata alla funzione '**`getsockname`**'.
2. Questo metodo riguarda solo il protocollo HTTP.
3. Non si può però tentare di reinderizzare la porta 21 (FTP) verso il proxy server per aggirare il problema, perché il browser comunicherebbe con un protocollo sbagliato; questo succede perché comunque ogni richiesta di proxying fatta dal browser viene convertita in una HTTP.

Le modifiche da effettuare al file '`/etc/squid.conf`' precedente riguardano solamente due opzioni che qui riporto; per il significato dei termini fare riferimento alla 14.1.

```
--------------------------------
# OPZIONI PER HTTPD ACCELERATOR
--------------------------------

httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
```

Capitolo 15

Multi router traffic grapher

MRTG (*Multi router traffic grapher*) è un applicativo che monitorizza il carico in punto rete. Vengono generate immagini con il formato GIF che rappresentano un andamento reale del traffico generato.

Il codice è basato sul linguaggio Perl e sul linguaggio C ed è multi piattaforma.

Lo script in Perl usa il protocollo SNMP per leggere e contare il traffico dei router, mentre il programma in C genera gli output grafici delle reti monitorizzate. Per visualizzare questi grafici è sufficiente un browser per web.

Oltre ai report giornalieri, MRTG genera statistiche settimanali, mensili e annuali. Il sistema di log è decisamente stabile e senza occupare molto spazio su disco, tiene traccia degli ultimi due anni di attività.

MRTG non impone limiti di traffico, né di monitoraggio dei dispositivi. Qualsiasi variabile del protocollo SNMP è controllabile. Carico del computer, sessioni di loggin degli utenti ecc.

15.1 CfgMaker

I file di configurazione '**`mrtg.cfg`**' possono essere generati automaticamente dal comando '`cfgmaker`'.

`cfgmaker` [***options***] ***community@router*** [***community@router***]

Le opzioni del comando sono le seguenti:

`--ifref=nr`
: Riferimenti all'interfaccia con il numero (default)

`--ifref=ip`
: Riferimenti all'interfaccia con il numero Ip

`--ifref=eth`
: Riferimenti all'interfaccia con il numero Ethernet

`--ifref=descr`
: Riferimenti all'interfaccia con la descrizione della stessa

`--ifref=name`
: Riferimenti all'interfaccia con il nome della stessa

`--ifref=type`
: Riferimenti all'interfaccia con il tipo

`--ifdesc=nr`
: Descrizione dell'interfaccia con riferimento il numero

`--ifdesc=ip`
: Descrizione dell'interfaccia con riferimento il numero Ip

`--ifdesc=eth`
: Descrizione dell'interfaccia con riferimento il numero Ethernet

`--ifdesc=descr`
: Descrizione dell'interfaccia con riferimento la descrizione della stessa

`--ifdesc=name`
: Descrizione dell'interfaccia con riferimento il nome della stessa

--ifdesc=alias

Descrizione dell'interfaccia con riferimento l'alias dell'interfaccia

--ifdesc=type

Descrizione dell'interfaccia con riferimento il tipo di interfaccia

--if-filter=f

Verifica ogni interffacia con un filtro '**f**' per decidere quale interfaccia includere o meno nell'insieme. Attualmente il filtro '**f**' è valutato come un'espressione scritta in Perl ed il valore '**vero**' o '**falso**' viene usato per accettare o rifiutare l'interfaccia

--if-template=templatefile

Un file che contiene codice perl con lo scopo di rimpiazzare il contenuto dell'interfaccia con un altro. Caratteristica in via di sperimentazione.

--host-template=templatefile

Un file che contiene codice perl con lo scopo di rimpiazzare i contenuti sia dell'interfaccia sia dell'host stesso con altri. Caratteristica in via di sperimentazione.

--global "x: a"

Aggiunge una voce di configurazione.

--no-down

Non visualizza lo stato dell'interfaccia.

--show-op-down

Visualizza le interfacce che non operano più.

--descint

Descrive l'interfaccia oltre all'analisi del traffico.

--subdirs=format

Assegna ad ogni router una sottodirectory, l'argomento "format" accetta le variabili HOSTNAME e SNMPNAME. Il formato può essere '**--subdirs=HOSTNAME**' o '**--subdirs="HOSTNAME (SNMPNAME)"**'.

--noreversedns

Disabilita la risoluzione dei nomi di dominio.

--community=cmty

Assegna la stringa "cmty" come nuova community invece di quella di default "public".

--snmp-options=:[<port>][:[<tmout>][:[<retr>][:[<backoff>][:<ver>]]]]

Le opzioni SNMP da aggiungere a tutti i router. I router, successivamente, possono riscrivre tutte le opzioni qui assegnate.

--dns-domain=domain

Imposta il dominio da aggiungere a tutti i router.

--nointerfaces

Non genera nessuna linea di configurazione per le interfacce, salta la configurazione della stessa e non avvia nessun codice template dell'interfaccia.

--interfaces

Genera le linee di configurazione per le interfacce. Comportamento predefinito.

--help

Visualizza i messaggi di aiuto.

--man

Visualizza tutta la documentazione.

--version

Visualizza il numero di versione di ‘cfgmaker’.

--output=FILE

File di uscita, il default è lo standard output.

15.1.1 Script di esempio

Creiamo un file di esempio per la macchina ‘**router1.inf.besta**’.

```
# cfgmaker --global "WorkDir: /var/www/html/mrtg" --ifref=ip
public@router1.inf.besta > mrtg.cfg [ Invio ]
```

La community è public, mentre la directory dove verranno memorizzati i file html è ‘**/var/www/html/mrtg**’. Il numero IP dell’interfaccia viene preso come riferimento per i grafici.

L’esempio successivo crea un file di configurazione per due router sempre con la community public.

```
# cfgmaker --global "WorkDir: /var/www/html/mrtg" --ifref=ip
public@router1.inf.besta public@router2.inf.besta > mrtg.cfg [ Invio ]
```

15.2 IndexMaker

L’applicazione ‘indexmaker’ crea un file di indice per i siti monitorizzati da MRTG.

indexmaker [*options*] *regexp* mrtg.cfg
[*community@router*]

Le opzioni del comando sono le seguenti:

--output-filename

File di uscita, il default è lo standard output.

--filter (title|pagetop|name)(=~|!~)regexp

Possono essere impostati più filtri. Ognuno di essi viene confrontato con una sezione specifica del file ‘**mrtg.cfg**’.

--title text

Imposta il titolo del file di configurazione generato.

--bodyopt text

Il valore dell’argomento text viene aggiunto al tag BODY.

--columns number

Visualizza il grafico in una tabella con i numeri di colonna.

--sort title|name|descr|original

Riordina i grafici nella pagina per titolo, nome, descrizione ll’interfaccia o lascia invariato.

--enumerate

Aggiungi una sequenza di numeri al titolo di ogni pagina.

--width number

Imposta la larghezza del grafico.

--height number

Imposta l'altezza del grafico.

--show day|week|month|year|none

Imposta il grafico da visualizzare nella pagina indice.

--section h1|title|name|description|portname

Imposta il titolo da visualizzare per i grafici in ogni pagina.

--rrdviewer path

Imposta il percorso dello script cgi che genera il grafico.

--prefix path

Il file generato da '`indexmaker`' viene memorizzato nella directory di lavoro (WorkDir). Il prefisso path imposta il percorso che deve essere raggiungibile dalla WorkDir.

15.2.1 Script di esempio

Creiamo un file di indice per il file di configurazione '**`mrtg.cfg`**'.

```
# indexmaker mrtg.cfg > index.html [ Invio ]
```

Capitolo 16

MySQL Server

MySQL server si presenta come un database adatto esclusivamente per applicazioni orientate al WEB. Didatticamente sarebbe più indicato usare PostgreSQL in quanto esso offre funzioni avanzate come i Trigger le transazioni.

MySQL invece è molto flessibile per gestire Database anche voluminosi dove sia richiesto un rapido accesso alle colonne delle tabelle. Inzialmente mi sono avvicinato al PostgreSQL versione 7, apprezzandone la facilità di installazione, la facilità con cui si creano database ed utenti; per contro è molto pesante per applicazioni WEB, esempi di script (PHP, Perl) sono quasi tutti con MySQL, in quanto è il più usato.

Come dire: insegnate il PostgreSQL (è più flessibile), se dovete sviluppare "codice" per applicativi WEB (dove non sono richieste le transazioni) usate MySQL.

16.1 Amministrare MySQL

Installati i pacchetti la prima operazione seguente è assegnare una password all'ammistratore del server SQL.

```
# mysqladmin -u root password password [ Invio ]
```

Se verrà cambiata in un secondo momento digitare invece:

```
# mysqladmin -p -u root password password [ Invio ]
  Enter password: old password
```

Possiamo quindi creare un database di prova, dove effettueremo alcuni test.

```
# mysqladmin -p create prova_db [ Invio ]
  Enter password: password
```

L'opzione -p chiede di digitare la password all'utente che sta tentando di accedere al database, in questo caso root. Per cambiare utente inserire l'opzione -u.

```
# mysqladmin -p -u utente create prova_db [ Invio ]
  Enter password: password
```

Per accedere al database appena creato digitare:

```
# mysql -p prova_db [ Invio ]
  Enter password: password

  Welcome to the MySQL monitor.  Commands end with ; or \g.
  Your MySQL connection id is 4 to server version: 3.23.32

  Type 'help;' or '\h' for help. Type '\c' to clear the buffer

  mysql>
```

L'utility mysql è equivalente a psql di PostgreSQL, benché meno sofisticata. Se l'accesso è avvenuto correttamente possiamo costruire la nostra prima tabella.

```
  mysql> CREATE TABLE prova_table (id_articolo INT PRIMARY KEY NOT NULL,
  desc_articolo VARCHAR(200));
  Query OK, 0 rows affected (0.03 sec)
  mysql>
```

Inseriamo un paio di valori:

```
mysql> INSERT INTO prova_table VALUES (1, 'SCARPE DA TENNIS');
Query OK, 1 row affected (0.05 sec)
mysql> INSERT INTO prova_table VALUES (2, 'SCARPE DA GINNASTICA');
Quey OK, 1 row affected (0.00 sec)
```

Semplice query:

```
mysql> SELECT * FROM prova_table;
+-------------+----------------------+
| id_articolo | desc_articolo        |
+-------------+----------------------+
|           1 | scarpe da tennis     |
|           2 | scarpe da ginnastica |
+-------------+----------------------+
2 rows in set (0.00 sec)
```

Per uscire dal database basta gigitare \q. Aggiungiamo ora un utente generico oltre a quello già presente che è root. MySQL permette di gestire politche di accesso sia per quanto riguarda tutti i database, sia per quanto riguarda specifici database. Rispetto al PostgreSQL trovo questa caratteristica più completa, sebbene l'amministrazione risulti più complicata. Per questo è bene conoscere in modo approfondito come MySQL gestisce gli utente ed i permessi di accesso.

16.2 Sicurezza in MySQL

Installato ed avviato il server di database, MySQL generera un database mysql che contiene cinque tabelle: db, host, user, tables_priv e columns_priv. Sono chiamata tabelle di sicurezza, in quanto MySQL decide chi è autorizzato ad accedere al database e con quali privilegi.

Per accedere al database mysql digitare:

```
# mysql -p mysql [ Invio ]
  Enter password: password
```

16.2.1 Tabella user

La tabella user contiene le seguenti colonne:

Tabella 16.1. Argomenti tabella user.

Campo Tipo	Null	Chiave	Default	Extra
Host	char(60)		PRI	
User	char(16)		PRI	
Password	char(16)			
Select_priv	enum('N','Y')			N
Insert_priv	enum('N','Y')			N
Update_priv	enum('N','Y')			N
Delete_priv	enum('N','Y')			N
Create_priv	enum('N','Y')			N
Drop_priv	enum('N','Y')			N

Campo Tipo	Null	Chiave	Default	Extra
Reload_priv	enum('N','Y')			N
Shutdown_priv	enum('N','Y')			N
Process_priv	enum('N','Y')			N
File_priv	enum('N','Y')			N
Grant_priv	enum('N','Y')			N
References_priv	enum('N','Y')			N
Index_priv	enum('N','Y')			N
Alter_priv	enum('N','Y')			N

Nelle colonne Host e User si può inserire il carattere jolly % il quale permette di sostituire più caratteri; esempio: %base completa firstbase e secondbase.

Il terzo campo password imposta la password per quell'utente, con la funzione *password()* quest'utlima può essere codificata. I campi seguenti accettano solamente due valori (il tipo è un ENUM): Y o N, assicurano vari livelli di richieste di accesso al database. Per *default* sono tutti impostati su N.

Di seguito un breve esempio. Accediamo al database mysql quindi digitiamo:

```
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2 to server version: 3.23.32

Type 'help;' or '\h' for help. Type '\c' to clear the buffer

mysql> INSERT INTO user (Host, User, Password) VALUES ('localhost',
'admin_db', password('db_pass'));
Query OK, 1 row affected (0.01 sec)
mysql>
```

L'utente admin ora può accedere al database SQL, tuttavia non ha possibilità di interagire con MySQL perché i suoi permessi sono tutti impostati su N. Supponiamo che l'unico compito dell'utente admin sia di creare database. Quindi digitiamo:

```
mysql> UPDATE user SET Create_priv='Y' WHERE User='admin_db';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

Questa ed altre modifiche si propagano per tutti i database esistenti, in un server possono esistere più database. Queste modifiche vanno quindi usate con *cums salis granis*, preferibile impostare i permessi per utenti nei rispettivi database. Per gestirne di multipli si usa la tabella db.

16.2.2 Tabella db

Questa tabella imposta i permessi non per l'intero server ma per i database presi singolarmente. L'unica differenza rispetto alla tabella user sta nel campo Password sostituito con il campo Db.

Tabella 16.2. Argomenti tabella db.

Campo Tipo	Null	Chiave	Default	Extra
Host	char(60)		PRI	
User	char(16)		PRI	
Db	char(16)			
Select_priv	enum('N','Y')			N
Insert_priv	enum('N','Y')			N
Update_priv	enum('N','Y')			N
Delete_priv	enum('N','Y')			N
Create_priv	enum('N','Y')			N
Drop_priv	enum('N','Y')			N
Reload_priv	enum('N','Y')			N
Shutdown_priv	enum('N','Y')			N
Process_priv	enum('N','Y')			N
File_priv	enum('N','Y')			N
Grant_priv	enum('N','Y')			N
References_priv	enum('N','Y')			N
Index_priv	enum('N','Y')			N
Alter_priv	enum('N','Y')			N

L'ordine di precedenza di questa tabella viene dopo di quella user. Ricordiamo che quella user si applica a tutti i database, ora si evince la necessità di controllare l'accesso degli utenti singolarmente da questa tabella e non da quella precedente.

16.2.3 Tabella host

L'ultima tabella da esaminare è la host. Simile alle precedenti, non ci sono campi User o Password. Permette di verificare l'acceso all'host in modo incrociato con la tabella db.

Tabella 16.3. Argomenti tabella host.

Campo	Tipo	Null	Chiave	Default	Extra
Host	char(60)		PRI		
Db	char(16)				
Select_priv	enum('N','Y')			N	
Insert_priv	enum('N','Y')			N	
Update_priv	enum('N','Y')			N	

Campo	Tipo	Null	Chiave	Default	Extra
Delete_priv	enum(’N’,’Y’)			N	
Create_priv	enum(’N’,’Y’)			N	
Drop_priv	enum(’N’,’Y’)			N	
Grant_priv	enum(’N’,’Y’)	Â	Â	N	Â
References_priv	enum(’N’,’Y’)			N	
Index_priv	enum(’N’,’Y’)			N	
Alter_priv	enum(’N’,’Y’)			N	

Se l’utente passa la verifica di accesso nella tabella db con valore vuoto nel campo Host, allora il controllo definitivo è affidato a quest’ultima tabella.

Le modifiche ai permessi vengono attivate solamente digitando questo comando:

```
# mysqladmin -p reload [ Invio ]
```

Capitolo 17

PostgreSQL Server

PostgreSQL è un database relazionale non ANSI SQL compatibile, però lo sviluppo prevede ad ogni step l'aderenza a questo standard. Diversamente dai DataBase con sviluppo commerciale PostgreSQL sfrutta un nuovo approccio incorporando nel suo motore nuovi tipi di dati come: classi, funzioni, ereditarietà; anche i trigger, come le regole consentono maggiore potenza e flessibilità. Queste caratteristiche fanno di PostgreSQL un database del tipo object-relational.

Queste caratteristiche sono state studiate ed inserite in altri database.

17.1 Amministrare PostgreSQL Server

L'amministrazione dei permessi di accesso ai database è garantita da un file data/pg_hba.conf. Da utente root diventare l'utente postgres con il comando su -. Entrare nel directory data ed aprire il file.

Questo file viene letto ogni qualvolta vi è una richiesta di connessione ad un database, non è quindi necessario riavviare PostgreSQL ad ogni modifica. Ogni riga prevede dei record; nella prima colonna si identifica il tipo di record: host, hostssl o local. Nelle successive: database, range di indirizzi IP, sottorete e tipo di autenticazione.

Il record local accetta solamente le colonne database e tipo di autenticazione.

```
# file data/pg_hda.conf

local   all                                     trust
host    all     192.168.1.0     255.255.255.0   trust
```

Accesso consentito a tutti gli host verso qualsiasi database (all). Questo sistema ne garantisce l'accesso, singoli permessi per utente e di gruppo avvengono con il comando GRANT di SQL.

```
# file data/pg_hda.conf

# Rifiuta la connessione solo all'host 192.168.1.1
host    all     192.168.1.1     255.255.255.255 reject
host    all     192.168.1.0     255.255.255.0   trust

# Accetta la connessione solo con password: in chiaro e criptata
host    all     192.168.1.0     255.255.255.0   password
host    all     192.168.2.0     255.255.255.0   crypt

# Accetta la connessione per il database esempio solo con protocollo ident
host    esempio 192.168.1.1     255.255.255.255 ident   utenti
```

L'ultima riga prende la mappa utenti presente nel file data/pg_ident.conf, la quale associa agli utenti ident quelli creatu con PostgreSQL. Esiste una mappa speciale sameuser e non presente in data/pg_ident.conf che associa ad ogni utente lo stesso nome per PostgreSQL.

Le password fornite non sono quelle UniX, bensì si trovano nel file base/pg_shadow; le password vanno fornite nel momento in cui si crea un utente. Altri sistemi di autenticazione sono previsti via tipo krb4 e tipo krb5.

17.2 Gestione utenti e database

Il comando createuser crea un nuovo utente di PostgreSQL.

```
bash-2.04$ createuser umbertoz
Shall the new user be allowed to create databases? (y/n) n
Shall the new user be allowed to create more new users? (y/n) n
CREATE USER
bash-2.04$
```

L'utente viene abilitato o disabilitato per la creazione di database o di più utenti. Ulteriori opzioni vengono passate per linea di comando come l'opzione -d che abilita la creazione di nuovi database, l'opzione -a che abilita la creazione di nuovi utenti, l'opzione -P che associa una password all'utente. Solo gli utenti abilitati con l'opzione usesuper set nel file pg_shadow possono crearne di nuovi.

```
bash-2.04$ createuser -d -a -P toni
Enter password for user "toni":
Enter it again:
CREATE USER
bash-2.04$
```

Tabella 17.1. Argomenti programma psql.

Opzione	Commento
-h	Specifica il nome della macchina dove un processo PostgreSQL è attivo.
-p	Specifica il numero della porta locale TCP/IP dove PostgreSQL è in ascolto.
-e	Stampa gli avvisi che createuser genera.
-q	Non visualizzare nessun avviso.
-d	Abilita il nuovo utente alla creazione di nuovi database.
-D	Nega al nuovo utente la creazione di nuovi database.
-a	Abilita il nuovo utente alla creazione di nuovi utenti.
-A	Nega al nuovo utente la creazioned di nuovi utenti.
-P	Il nuovo utente viene associato ad una password per l'accesso a PostgreSQL.
-i	Assegna uno userid diverso da quello assegnato; opzione non necessaria ma alcune volte richiesta.

Le opzioni -h, -p e -e vengono passate al comando psql con il medesimo significato.

Il comando createdb crea un nuovo database di PostgreSQL. L'utente che ha eseguito il comando diventa il possessore del database.

```
bash-2.04$ createdb prova
CREATE DATABASE
bash-2.04$
```

Come per il comando createuser ulteriori opzioni vengono fornite con linea di comando.

```
bash-2.04$ createdb -h db.inf.besta prova
CREATE DATABASE
bash-2.04$
```

Tabella 17.2. Argomenti programma createdb.

Opzione	Commento
-h	Specifica il nome della macchina dove un processo PostgreSQL è attivo.
-p	Specifica il numero della porta locale TCP/IP dove PostgreSQL è in ascolto.
-U	Nome utente per la connessione.
-W	Forza l'inserimento di una passaword.
-e	Stampa gli avvisi che createuser genera.
-q	Non visualizzare nessun avviso.
-D	Imposta il percorso non standard dove sono creati i database.
-E	Imposta lo schema di codifica dei caratteri usati per questo database.

Le opzioni -h, -p, -U e -W vengono passate al comando psql con il medesimo significato.

17.3 Primi passi con PostgreSQL

Il programma psql è usa un'interfaccia testuale per interagire con PostgreSQL ed analizzare i risultati delle query. Si è dimostrato altresì un valido strumento didattico: completo, funzionante e da sfruttare anche con i PC i286. Oltre ad accettare comandi SQL accetta un gran numero di metacomandi che permettono di stampare, generare codice HTML, lanciare script... Per lanciarlo è sufficiente conoscere il nome del database, l'host e la porta di accesso.

```
$ psql prova [ Invio ]
  Welcome to psql, the PostgreSQL interactive terminal.

  Type:  \copyright for distribution terms
         \h for help with SQL commands
         \? for help on internal slash commands
         \g or terminate with semicolon to execute query
         \q to quit

  prova=[num   ]
```

Se PostgreSQL non è attivo nella macchina locale si sfrutta l'opzione -h e per scegliere il database l'opzione -d; la porta di accesso con l'uso dell'opzione -p.

```
$ psql -h db.inf.besta -d prova [ Invio ]
```

Per ottenere la lista dei database registrati:

```
bash-2.04$ psql -l
         List of databases
   Database   |  Owner   | Encoding
--------------+----------+-----------
 discografia | postgres | SQL_ASCII
 prova       | postgres | SQL_ASCII
 template1   | postgres | SQL_ASCII
(3 rows)
```

La colonna Owner visualizza l'utente che ha creato il database e quindi ne ha pieno accesso in lettura e scrittura. Per accesso a database prottetti da password si usa l'opzione -W e per connettersi con utente diverso da quello con cui si accesso al terminale UniX l'opzione -U. Queste ultime opzioni non hanno effetto se nel file data/pg_hba.conf AUTHTYPE è impostata su trust.

Capitolo 18

Interbase

Interbase è un database relazione della INPRISE (http://www.inprise.com/) azienda legata a Borland. Dalla versione 6.0 è stata rilasciata licenza Open Source per chiunque voglia sviluppare ulteriormente le caratteristiche di questo prodotto. Subito il popolo del free software ha ricompilato i sorgenti e creato una versione parallela all'interbase 6.0 chiamata firebird e coordinata dalla IBPHOENIX (http://www.ibphoenix.com). Comparando Interbase ad altri prodotti Open Source tipo MySQL o meglio ancora PostgreSQL, non possiamo che apprezzarne le caratteristiche di velocità aderenza allo standard SQL-92 ed anche al nuovo SQL-3.

La Inprise distribuisce due versioni funzionati, una basata sul kernel classico (CS), una basata su kernel superserver (SS). Consiglio vivamente di installare la seconda versione, infatti a lavora a thread, diversamente dalla prima che lavaro a processi. I "difetto" di "origine" si notano, ricordiamo che Interbase fino alla versione 5.6 era proprietario e multipiattaforma, e l'avvio in una macchina con Linux Redhat 7, posso garantire che non è alla portata dell'utente medio.

Nonostante la presenza di file rpm, questi risultano incompleti e necessitano di supporto manuale. Si tratta di un prodotto UniX più che di un prodotto LiNUX. I forum ed il supporto tecnico non sempre lavorano come dovrebbero. Mezza giornata di lavoro e si aggiusta tutto! Difetti di gioventù esclusi, prendete in seria considerazione l'utilizzo di questa piattaforma di sviluppo. Molte utility di corredo, a volte a pagamento, sono create per ambiente Windows, però sono interfacce clients il server può essere installato senza difficoltà in ambiente LiNUX.

18.1 Installazione

I manuali in formato PDF liberamente scaricabili da Internet sono completi, e spiegano come installare e configurare Interbase sia su piattaforma Windows che UniX. Per questo motivo sono spessi, e scoraggiano l'utente a sfogliarli. Le informazioni sono sparse per questo vanno esaminati attentamente prima di procedere all'avvio del server di database. La prima parte dei vari capitoli riguarda l'uso in ambiente Windows, successivamente viene spiegata la stessa procedura per ambiente UniX (generalmente SUN).

Installare il pacchetto rpm della versione super server. Creare quindi l'utente interbas.

```
# useradd -d /opt/interbase -u 28 interbas [ Invio ]
```

Anche se può risultare una operazione non necessaria è vivamente consigliata per evitare che il server SQL si avvii con i permessi di amministratore, deleterio per la sicurezza in un network. La seconda operazione consiste nel modificare i permessi del directory dove viene installato interbase, questo directory è '`/opt/interbase`'.

```
# cd /opt/interbase [ Invio ]# chown -R interbas.interbas [ Invio ]
```

Ai file e directory presenti viene assegnato come proprietaro l'utente interbas. Interbase usa le variabili ISC_USER ed ISC_PASSWORD per individuare l'utente e relativa password per l'accesso al database. L'amministratore viene individuato nell'utente SYSDBA e la password inziale è masterkey. Per avviare e fermare il server SQL dobbiamo quindi servirci di questo super utente. INPRISE non ha fornito un file da inserire in /etc/rc.d/init.d per far partire il servizio, lasciando ad una pagina del manuale un file di script da copiare. Questo file può andar bene per una SUN, certo per LiNUX ha bisogno di alcune modifiche, per questo ho cercato di modificarlo per renderlo standard e di livello comparabile agli altri servizi preinstallati. Premetto che ho incoltrato alcune difficoltà in quanto il prodotto in caso di errore non riporta messaggi di errore da interpretare in console.

```
# file /etc/rc.d/init.d/ibserver

#!/bin/sh

# ibserver script - Start/stop the InterBase daemon

# chkconfig: 345 85 15
```

```
# description: Starts and stops the InterBase backend daemon that handles \
#              all database requests.
# processname: ibmgr
# pidfile: /var/run/ibserver.pid
#

. /etc/init.d/functions

: ${INTERBASE:=/opt/interbase}
: ${ISC_USER:=SYSDBA}
: ${ISC_PASSWORD:=masterkey}

start() {
  echo -n "Starting ibserver service: "
  su -l interbas -c '/opt/interbase/bin/ibmgr -start -forever 2>/dev/null'
  pid=‘pidof ibserver‘
  touch /var/lock/subsys/ibserver
  echo $pid > /var/run/ibserver.pid
  success
  echo
}

stop() {
  $INTERBASE/bin/ibmgr -shut -user SYSDBA -password $ISC_PASSWORD 2>/dev/null
  ret=$?
  if [ $ret -eq 0 ]; then
    action "Stopping ibserver service: " /bin/true
  else
    action "Stopping ibserver service: " /bin/flase
  fi
  [ $ret -eq 0 ] && rm -f /var/run/ibserver.pid
  [ $ret -eq 0 ] && rm -f /var/lock/subsys/ibserver
  return $ret
}

case "$1" in
  start)
    start
    ;;
  stop)
    stop
    ;;
  reload)
    stop
    star
    ;;
  status)
    status ibserver
    ;;
  *)
    echo "Usage: ibserver {start|stop|status|restart}"
    exit 1
esac

exit 0
```

La funzione *stop()* non funziona come dovrebbe, infatti in caso di errore (servizio non attivo), il server risponde con in messaggio di errore, ma restituisce sempre valore nullo. Dovrebbe rispondere con valore negativo. Questo script deve essere protetto con maschera 700, in quanto la password del super utente è in chiaro.

Copiato lo script in /etc/rc.d/init.d/ usare il comando chkconfig per aggiungere lo script nel livelli 3 e 5.

```
# chkconfig --add ibserver Invio
# chkconfig --level 35 ibserver on Invio
```

Aggiungere nel file /etc/hosts.equiv e quindi copiare questo file in /etc/gds_hosts.equiv la riga seguente:

```
# file /etc/hosts.equiv

localhost interbas
```

Un bug attualmente presente su questa versione di Interbase e che si manifesta con la Red Hat 7.0 può non avviare l'Intebase. L'errore c

18.2 Gestione utenti

L'utility per gestire gli utenti con interbase è gsec. Il comando può essere avviato sia se si è superutent (root) sia come utente interbas. Il secondo caso è quello consigliato, per evitare di utilizzare root come utente.

```
# su - interbas Invio
bash-2.04$ gsec -user SYSDBA -password password
GSEC>
```

Se state utilizzando per la prima volta interbase sicuramente l'unico utente presente è SYSDBA. Verifichiamo con il comando di.

```
GSEC>di Invio
      user name                     uid   gid     full name
------------------------------------------------------------------------------
SYSDBA                             0     0
```

Per gestire gli utente questa utility offre una serie di comandi con relativi parametri, i comandi possono anche essere abbreviati: esempio di sta per display, può essere utilizzato uno o l'altro indifferentemente.

add <name> [<parameter> ...]	Aggiunge un nuovo utente
delete <name>	Cancella un utente
display <name>	Visualizza i dati di un utente
modify <name> <parameter> [<parameter> ...]	Modifica i parametri di un utente
quit	Esci

I parmaetri disponibili sono i seguenti: -pw <password>, -uid <uid>, -gid <uid>, -fname <firstname>, -mname <middlename>, -lname <lastname>. Per aggiungere un nuovo utente digitare il seguente comando:

```
GSEC> add umbertoz -pw password -fname Umberto -lname Zanatta Invio
GSEC> di Invio
          user name                     uid   gid     full name
------------------------------------------------------------------------------
SYSDBA                             0     0
UMBERTOZ                           0     0      Umberto  Zanatta
GSEC>
```

Per modificare la password di umberto digitare:

```
GSEC> mo umbertoz -pw nuovapassword Invio
GSEC>
```

Rispetto al database MySQL visto in precedenza, per i permessi su tabelle e database bisogna usare i comandi standard SQL GRANT e REVOKE.

OpenLDAP

LDAP (*Lightweight directory access protocolo*) è un protocollo per accedere a servizi basati su directory generalmente in standard X.500. Il servizio di trasferimento delle directory è basato su protocollo Tcp/Ip ed il modello è di tipo client/server. I dati delle directory possono essere distribuiti su più server LDAP i quali formano una DIT (*Directory Information Tree*). Il server interrogato restituisce al client un puntatore alla risorsa, che così può essere facilmente recuperata.

19.1 X.500

19.2 Introduzione

Il modello di informazioni di LDAP è basato su indentità dette *entry*. Ogni entry rappresenta una collezione di attributi che è riferita ad uno ed uno solo DN (*Distinguished Name*). Un attributo ha come riferimento principale un *tipo* cui sono associati uno o più *valori*. La sintassi dei valori è strettamente legata alla natura stessa del tipo: un nome di persona, un indirizzo di posta ecc.

Queste informazioni sono organizzate in una struttura gerarchica ad albero dove, alla radice risiedono i vertici dell'organizzazione fino a scendere alle unità organizzative esempio: stato, uffici, persone e così via. L'albero è generalmente organizzato come un dominio Internet; questo facilita la ricerca delle directory LDAP utilizzando il servizio Dns.

Ogni entry possiede un attributo speciale denominato «classe oggetto» o ***objectClass*** che controlla quali sono gli attributi richiesti ed ammessi. Il valore associato allo ***objectClass*** definisce lo *schema* della entry, quindi le sue regole.

La entry viene referenziata attraverso il suo DN, il quale è costruito con il nome proprio della entry detto RDN (*Relative Distinguished Name*) concatenato con tutte le entry precedenti nella struttura gerarchica.

```
uid=umbertoz,ou=lab,dc=inf,dc=besta
```

Far riferimento al DN di una entry assegnata all'utente '**`umberto`**'. Gli RDN sono: '**`uid`**', '**`ou`**' e '**`dc`**'. LDAP sostanzialmente serve per ricercare informazioni all'interno dell'albero delle directory; tuttavia è possibile aggiungere, modificare o cancellare le entry. La possibilità di costruire filtri consente di semplificare o affinare i criteri di ricerca.

19.3 Servizi

L'attività delle directory in LDAP è distribuita su due servizi: '**`slapd`**' e '**`slurpd`**'. Il primo è il server delle directory, il secondo consente di replicare le informazioni di un server principale o master ai server secondari.

19.3.1 Slapd

Slapd è il demone per il server delle directory: risponde alle richieste dei client e distribuisce le informazioni ricercate. Sfrutta il servizio Tcp/Ip in ascolta sulla porta predefinita 389.

```
# /etc/init.d/slapd start
```

Avvia il demone '**`slapd`**' in una distribuzione Debian GNU/Linux.

I file di configurazione principali sono '`/etc/ldap/ldap.conf`' e '`/etc/ldap/slapd.conf`'. Il primo fornisce le informazioni predefinite ai client relative al servizio LDAP; ogni utente può rifiutarle e sostituirle con il file '`.ldaprc`' nella propria home directory. Il secondo contiene le informazioni globali per il servizio '**`slapd`**' e '**`slurpd`**'.

19.3.2 Slurpd

Slurpd è il demone che distribuisce le informazioni da un server LDAP ad un altro. La replica è garantita da un registro di replicazione prodotto da Slapd che Slurpd si incarica di propagare agli altri server.

```
# /etc/init.d/slurpd start
```

Avvia il demone '**slurpd**' in una distribuzione Debian GNU/Linux.

Se il demone '**slapd**' non produce il registro di replicazione '**slurpd**' non viene avviato.

19.3.3 `# /etc/ldap/slapd.conf`

In listato 19.5 è presentato un semplice file di configurazione per uno schema di directory che fa riferimento all'organizzazione '**inf.besta**'.

Il file è organizzato in una serie di configurazioni globali seguite da configurazioni relative al tipo di motore di database e al database utilizzato per memorizzare l'albero delle directory. Quest'ultimo può essere presente in quantità, dipende da quanti database si utilizzzano e possono sovrascrivere le opzioni definite nella sezione globale. I commenti iniziano con il carattere '#' e gli argomenti sono separati con lo spazio bianco.

```
<configurazioni globali>

backend tipo
database tipo
<configurazioni del database>

database tipo
<configurazioni del database>

...
backend tipo
database tipo
<configurazioni del database>

database tipo
<configurazioni del database>
```

Listato 19.5. File di configurazione di Slapd

```
# Schema e definizione delle objectClass
include         /etc/ldap/schema/core.schema
include         /etc/ldap/schema/cosine.schema
include         /etc/ldap/schema/nis.schema
include         /etc/ldap/schema/inetorgperson.schema

# Impone la corrispondenza tra le entry e le
# classi oggetto di definzione
schemacheck     on

# File pid del demone
pidfile         /var/run/slapd.pid

# Lista delle opzioni da passare al demone slapd
argsfile        /var/run/slapd.args

# Il file di registro per la replicazione delle directory
replogfile\011/var/lib/ldap/replog

# Aumenta le informazioni sul file di registro
loglevel        0

backend ldbm

# Il tipo di database utilizzato, ldbm e' il predefinito
database        ldbm
suffix\011\011"dc=inf,dc=besta"
rootdn\011\011"cn=Manager,dc=inf,dc=besta"
rootpw\011\011{SSHA}toMOC43opY83aCk9JJhcT4sF7s6bzPCr

# Directory dove risiede il database
directory       "/var/lib/ldap"

# Opzioni per l'indicizzazione
index objectClass eq

# Registra la data di modifica della entry
lastmod on

# Liste per il controllo di accesso (ACL)
access to attribute=userPassword
       by dn="cn=Manager,dc=inf,dc=besta" write
       by anonymous auth
       by self write
       by * none

access to *
       by dn="cn=Manager,dc=inf,dc=besta" write
       by * read
```

19.3.4 Slappasswd

`slappasswd` [*options*]

Alcune opzioni

Opzione	Descrizione
-v	Visualizza informazioni dettagliati
-u	Genera una password conforme allo standard RFC2307 per l'attributo '**userPassword**'. Viene impiegato per ragioni di compatibilità con le versioni precedenti di LDAP.
-s [*password***]**	Genera il codice hash della password ***password***.
-h ***scheme***	Sono ammessi i seguenti schemi di codifica: '**CRYPT**', '**MD5**', '**SMD5**', '**SSHA**' e '**SHA**' che è lo schema predefinito.

Opzione	Descrizione
-c *crypt-salt-format*	Specifica il formato del vettore salt quando viene utilizzato lo schema '**CRYPT**'. La stringa deve essere in formato '**sprintf**' e viene generata casualmente. Valori ammissibile possono essere '**%.2s**' con due caratteri o '**1%.8s**' che fornisce otto caratteri per le versioni di '**crypt**' che utilizzano la codifica '**MD5**'. Il valore predefinito è '**%s**' che garantisce un vettore di salt con 31 caratteri.

Slappasswd viene utilizzato per generare una password da poter utilizzare come password per l'attributo '**userPasswrod**' o per l'attributo '**roopwd**' nel file '`/etc/ldap/slapd.conf`'.

slappasswd -s abcd1234 [*Invio*]

```
{SSHA}ctb39cvwzB20syKGlwSQHOvjR6ONWvQv
```

Genera il codice di hash della password '**abcd1234**'.

19.3.5 Aggiungere una entry

Le entry vengono aggiunte al servizio di directory LDAP attraverso un file di dati in formato LDIF (*LDAP Data Interchange Format*). I programmi '**ldapadd**' e '**ldapsearch**' consentono di manipolare le entry.

```
dn: <distinguished name>
<attrdesc>: <attrvalue>
<attrdesc>: <attrvalue>
<attrdesc>:: <base64-encoded-value>
<attrdesc>:< <URI>
```

Per facilitare la scrittura degli attributi una linea può essere continuata inserendo un singolo carattere di tabulazione o uno spazio vuoto in quella successiva. I commenti iniziano con il carattere '#'. Multiple definizioni dello stesso attributo sono permesse.

I valori codificati nella notazione base64 iniziano con '::'; le URI con il carattere ':<'.

Listato 19.8. Un esempio di file LDIF

```
dn: uid=umbertoz,dc=inf,dc=besta
objectClass: top
objectClass: account
objectClass: posixAccount
cn: umbertoz
uid: umbertoz
uidNumber: 1001
gidNumber: 100
homeDirectory: /home/umbertoz
loginShell: /bin/bash
userPassword: {SSHA}UeIpGls27cg0uak7Va2mDHT3SRoZiuV0
```

In listato 19.8 è rappresentato il file '`account.ldif`' in formato LDIF.

 # ldapadd -x -D "cn=Manager,dc=inf,dc=besta" -W -f account.ldif [*Invio*]

Aggiunge le entry organizzate nel file '`account.ldif`'. Alla richiesta di password digitare quella relativa all'amministratore LDAP realtivo al dominio '**inf.besta**'.

 # ldapsearch -x -b "uid=umbertoz,dc=inf,dc=besta" [*Invio*]

```
version: 2

#
# filter: (objectclass=*)
# requesting: ALL
#

# umbertoz, inf, besta
dn: uid=umbertoz,dc=inf,dc=besta
objectClass: posixAccount
cn: umbertoz
uid: umbertoz
uidNumber: 1001
gidNumber: 100
homeDirectory: /home/umbertoz
loginShell: /bin/bash

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

Ricerca nell'albero delle directory relativo al dominio '**inf.besta**' il DN relativo alla '**uid=umberto**'.

19.3.6 Cancellare una entry

Per cancellare una entry presente nel database si può utilizzare un il file '`account-del.ldif`' in formato LDIF come in listato 19.10.

Listato 19.10. Un esempio di file LDIF.

```
dn: uid=umbertoz,dc=inf,dc=besta
changetype: delete
```

 # ldapadd -x -D "cn=Manager,dc=inf,dc=besta" -W -f account-del.ldif [*Invio*]

Elimina la entry relativa all'utente '**umbertoz**'.

19.3.7 Modificare la password

La password relativa all'oggetto '**userPassword**' può essere modificata attraverso il comando '**ldappasswd**'.

 # ldappasswd -x -S -D "cn=Manager,dc=inf,dc=besta" "uid=umbertoz,dc=inf,dc=besta" -W [*Invio*]

```
New password:
Re-enter new password:
Result: Success (0)
```

Sendmail

20.1 Introduzione

Sendmail è diventato nel corso degli anni uno standard de facto per server di posta elettronica nei sistemi Unix. Nonostante non sia ancora un prodotto affidabilissimo offre all'amministratore un'ampia possibiltà di configurazione.

Pochi si saranno cimentati nella lettura delle centinaia di pagine di documentazione, sia per la loro difficoltà, sia perché esiste un metodo molto più semplice per gestire sendmail. Si può, infatti, creare un file che verrà interpretato da un preprocessore ('**m4**') il quale genererà a sua volta il file di configurazione '**sendmail.cf**'.

20.2 Configurazione di Sendmail utilizzando m4 (SERVER)

Studieremo un caso, ove vi sia una intranet d'istituto con un dominio fittizzio ('**inf.besta; mat.besta**'), che disponga di un collegamento con ip statico, e di un dominio registrato al nic. La posta in uscita potrà essere instradata verso un altro server (smart host) oppure gestita localmente. La soluzione preferibile è la prima perché evita che eventuali domini momentaneamente irraggiungibili occupino il servizio di posta (quindi banda) inutilmente.

Il file da modificare è /etc/mail/sendmail.mc di seguito riportiamo un esempio.

```
# file /etc/mail/sendmial.mc

divert(-1)
include('/usr/lib/sendmail-cf/m4/cf.m4')
VERSIONID('linux setup for Red Hat Linux')dnl
OSTYPE('linux')
define('confDEF_USER_ID',''8:12'')dnl
undefine('UUCP_RELAY')dnl
undefine('BITNET_RELAY')dnl
define('confAUTO_REBUILD')dnl
define('confTO_CONNECT', '1m')dnl
define('confTRY_NULL_MX_LIST',true)dnl
define('confDONT_PROBE_INTERFACES',true)dnl
define('PROCMAIL_MAILER_PATH','/usr/bin/procmail')dnl
define('ALIAS_FILE','/etc/aliases')dnl
define('STATUS_FILE', '/var/log/sendmail.st')dnl
define('UUCP_MAILER_MAX', '2000000')dnl
define('confUSERDB_SPEC', '/etc/mail/userdb.db')dnl
FEATURE('smrsh','/usr/sbin/smrsh')dnl
FEATURE('mailertable','hash -o /etc/mail/mailertable')dnl
FEATURE('virtusertable','hash -o /etc/mail/virtusertable')dnl
FEATURE('redirect')dnl
FEATURE('always_add_domain')dnl
FEATURE('use_cw_file')dnl
FEATURE('local_procmail')dnl
FEATURE('access_db')dnl
FEATURE('blacklist_recipients')dnl
FEATURE('masquerade_envelope')
MASQUERADE_AS('linuxdidattica.org')
MASQUERADE_DOMAIN('inf.besta mat.besta')
MAILER('smtp')dnl
MAILER('procmail')dnl
```

File di configurazione standard inserito nella distribuzione Red Hat 7.0 più alcune aggiunte.

Attraverso l'uso delle macro è possibile controllare il modo in cui Sendmail spedisce la posta e come si dovrà comportare in determinate situazioni.

Tabella 20.1. Argomenti file sendmail.mc.

OSTYPE	Specifica il tipo di sistema operativo che il file di macro dovrà elaborare. La sintassi corretta è: OSTYPE('os')
DEFINE	Server per ridefinire alcune opzioni nel file di configuazione di Sendmail. La sintassi corretta è: define('option
FEATURE	Consente di modificare i comportamenti delle caratteristiche predefinite in sendmail. La sintassi corretta è: FE
MASQUERADE_AS	Permette a Sendmail di mascherare il nome di dominio con un altro di diverso. Nel nostro esempio il dominio
MASQUERADE_DOMAIN	Consente di mascherare domini multipli. La sintassi corretta è: MASQUERADE_DOMAIN(domain domain .
MAILER	Specifica quali sono gli agenti di posta che Sendmail dovrà usare per trasportare i messaggi. La sintassi corret

20.2.1 FEATURE

Con l'uso della macro FEATURE come detto è possbile modificare alcuni comportamenti di Sendmail. Di seguito una concisa descrizione di alcuni di questi:

Tabella 20.2. Tabella file m4.

allmasquerade	Se il masquerade (MASQUERADE_AS) è attivo questa feature consente di mascherare anche l'indirizzo e non
always_add_domain	Aggiungi il nome del dominio anche se la posta è distribuita localmente.
local_procmail	Impone a Sendmail di usare procmail come mailer locale per il trasporto dei messaggi.
masquerade_entire_domain	Maschera con il nome di dominio non solo la posta proveniente dall'host locale, ma dagli host interni ad un dor
masquerade_envelope	Generalmente solo la testatina del messaggio è macherata. Usare questa direttiva per mascherare tutti campi del
use_cw_file	Consente a Sendmail di leggere un file che contiene nomi host alternativi per il server di posta. Il file in questio
virtusertable	Consente a domini multipli di essere ospitati su uno stesso server. Il file contente la tabella virtuale conterrà nel
genericstable	Consente agli indirizzi di posta non qualificati di essere confrontati con degli indirizzi presenti in una mappa e c

Visto in modo molto rapido alcune basilari opzioni non resta altro che creare il file '**`/etc/sendmail.cf`**': '**`m4 sendmail.mc > /etc/sednmail.cf`**'

Riavviare quindi il servizio Sendmail:

```
# service sendmail restart Invio
```

Il file '**`sendmail.cf`**' potrebbe essere utlteriormente modificato per evitare che la connessione parta inutilmente e faccia aumentare gli scatti telefonici; per prima cosa il comando di avvio dovrà cambiare da:

```
# file /etc/rc.d/init.d/sendmail
...
daemon /usr/sbin/sendmail $([ "$DAEMON" = yes ] && echo -bd) \
                                  $([ -n "$QUEUE" ] && echo
-q$QUEUE) -os
```

Un'altra riga di '**`sendmail.cf`**' va modificata:

```
# Inizialmente era impostata su False

O HoldExpensive=True
```

In questa mainera si evita lo svuotamento della cache che sarà effettuato successivamente inserendo '**`sendmail -q &`**' nel file '**`ip-up (ip-up.local)`**'.

Per finire nei flag '**`F=mdFMuX`**' nelle righe che contengono' **`Msmtp`**', '**`Mesmtp`**' ... inserire la lellera '**`e`**' (che sta per expensive).

20.3 Configurazione di Sendmail utilizzando m4 (WORKSTATION)

20.4 Aliases

Un modo molto semplice per spedire posta a più persone utilizzando un unico alias consiste nell'editare il file '**/etc/aliases**', quest'ultimo contiene una serie di nicknames che puntano ciascuno a più indirizzi di posta.

```
# file /etc/aliases
Postmaster:    root
net:           utente1, utente2, utente3, utente4, utente5
nome.cognome:  utente1
root:          utente1
```

Partendo da questo file, ed attraverso il comando' **newaliases,**' si genera un altro file di nome '**/etc/aliases.db**' che rappresenta il database utilizzato che utilizzerà sendmail.

Se l'utente '**utente1**' specificasse come destinatario '**ugonet**', il contenuto sarà spedito agli utenti specificati nella seconda colonna.

Un'osservazione per quanto riguarda la terza riga. L'utente '**utente1**' voleva che il suo indirizzo di posta fosse '**nome.cognome**'; purtroppo la userID non può superare la lunghezza massima di otto caratteri quindi, si è preferito utilizzare un alias.

Un buon amministratore non dovrebbe usare mai root come indirizzo di posta per ragionevoli motivi di sicurezza. L'accesso al Server come root deve essere limitato al minimo. Per questi motivi è consigliabile usare sempre un altro indirzzo di mail ed usare un alias che permetta di spedire tutta la posta per '**root**' ad un altro utente, nell'esempio '**utente1**'.

20.5 VIRTUSERTABLE

La feature virtusertable permette di ospitare più domini virtuali su uno stesso server. Il file contenente la tabella è tipo il seguente esempio:

```
# file /etc/mail/virtusertable

umbertoz@esempio.it     umbertoz
toni@esempio.it         toni
@esempio2.it            %1@esempio3.it
vecchio+*@esempio.it    nuovo+%2@esempio.it
```

Le mail indirizzate alla prima colonna vengono reindirizzate all'email della seconda colonna. La parola "%1" sostituisce lo userid, la parola "%2" sostituisce il dettaglio: nel nostro esempio la parola dopo l'asterisco.

Il server deve accettare il relay e la posta in entrata per questi domini, nell'esempio: esempio.it, esempio2.it, esempio3.it. Nel file /etc/mail/local-host-name inserire tutti i domini virtuali (o meglio:tutti i nomi di dominio con cui il server viene riconosciuto), mentre nel file /etc/mail/access compilare le colonne per consentire il relay della posta.

```
# file /etc/mail/access

localhost.localdomain   RELAY
localhost               RELAY
esempio.it              RELAY
esempio2.it             RELAY
```

```
#file /etc/mail/local-host-name

esempio.it
localhost
esempio2.it
```

20.6 GENERICSTABLE

L'utilizzo della feature genericstable è simile a quello degli aliases, diversamente il nome utente e/o il dominio sono modificati per i messaggi in uscita. Sia, quindi, il nome utente che il dominio vengono mascherati. Il file da modificare è /etc/mail/genericstable; la prima colonna contiente l'indirizzo e-mail di origine, la seconda quello nuovo.

```
# file /etc/mail/genericstable

utente1 umbertoz
utente2 antonio
toni    utente3@esempio.it
```

Le aggiunte da effettuare nel file da processare con l'm4 sono le seguenti:

```
FEATURE('genericstable', 'hash -o /etc/mail/genericstable')
GENERICS_DOMAIN('dominio1 dominio2')
```

Solo i domini *dominio1* e *dominio2* saranno eventualmente mascherati con il nuovo userid e/o domainname come da tabella nel file /etc/mail/genericstable. Per creare la mappa aggiungere nel file di script di avvio di sendmail la parola genericstable primo o dopo virtusertable.

```
# file /etc/rc.d/init.d/sendmail

start() {
        # Start daemons.

        echo -n "Starting sendmail: "
        /usr/bin/newaliases > /dev/null 2>&1
        for i in genericstable virtusertable access domaintable mailertable ; do
            if [ -f /etc/mail/$i ] ; then
                makemap hash /etc/mail/$i < /etc/mail/$i
            fi
        done
        daemon /usr/sbin/sendmail $([ "$DAEMON" = yes ] && echo -bd) \
                                $([ -n "$QUEUE" ] && echo -q$QUEUE)
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/sendmail
        return $RETVAL
}
```

Riavviare quindi il servizio.

```
# service sendmail restart Invio
```

20.7 PINE

Come "vi" è definito il tool di amministrazione universale anche pine potrebbe essere definito come il programma di posta universale.

Alcuni di voi potrebbero obiettare preferendo mail! Sinceramente trovo PINE robusto affidabile e didattico per chi inizia.

In una rete con molti host la posta, come precedentemente spiegato, dovrebbe essere centralizzata in un unico server. Per default tutta la posta viene spedita nella cartella '**`/var/spool/mail/nomeutente`**'. In ogni host sarebbe sufficiente montare via NFS questa directory ed il gioco è fatto.

In una rete con molti host non è molto conveniente perché aumenterebbe di molto il traffico per la rete locale. Poi risulterebbe dispendioso controllare la lunghezza dei file di posta di ognuno quindi trovare una soluzione per memorizzare i messaggi di posta nella HOME utente potrebbe permettere di eliminare questi incovenienti; unito all'utilizzo delle quota per il disco non porta velocemente a saturazione i dischi, ognuno è responsabile nel contenere lo spazio della propria mailbox.

è sufficiente installare procmail ed il gioco è fatto!. Questo software trasferisce tutta la posta in entrata verso una directory prestabilita. Può anche comportarsi da filtro. è quindi un buon tool di smistamento della posta in entratra.

Per il nostro scopo è sufficiente creare il file /etc/procmailrc (per ogni host presente in rete, compreso il server) ed inserire le seguenti righe:

```
# file /etc/procmailrc

MAILDIR=$HOME/.mail       # Questa directory deve essere creata (mkdir .mail)
DEFAULT=$MAILDIR/mbox     # Questo file deve essere creato (touch mbox)
LOGFILE=$MAILDIR/from
LOCKFILE=$HOME/.lockmail
```

Ogni utente dovrà crearsi la directory '`.mail/`' ed il file '`mbox`' all'interno della stessa. È conveniente crearli anche nella directory '`/etc/skel`' in modo che ogni nuovo utente trovi "tutto pronto".

Il contenuto della directory '`/etc/skel`' viene copiato nella directory di un nuovo utente al momento della sua aggiunta in un server LiNUX.

Ua directory nascosta è preferibile, spesso non interferisce con i "moderni" software di posta che utilizzano grafica.

Nel profile utente o di sistema cercare individuare la variabile MAIL generalmente impostata su '`/var/spool/mail/$USER`' e cambiarla in:

```
# file /etc/profile

MAIL=$HOME/.mail/mbox
```

Anche PINE deve essere configurato in modo appropriato perché prelevi la posta non dal directory stabilito, bensì da quello imposto. Per ogni host e per il server modificare il file di configurazione '`/etc/pine.conf.fixed`', generalmente vuoto, ed aggiungere la seguente riga:

```
# file /etc/pine.conf.fixed

inbox-path=.mail/mbox
```

Non modificare il file /etc/pine.conf perché viene sovrascritto da quello utente '`$HOME/.pinerc`'.

Capitolo 21

StarOffice

21.1 Introduzione

StarOffice è un pacchetto composto da un elaboratore di testi, un folgio elettronico, un programma per le presentazioni, ritocco di immagini, e un database. L'eseguibile è pienamente distribuibile senza dover pagare costose licenze software.

Dalla prossima versione cambierà nome, da StarOffice ad OpenOffice, e sarà distribuito con licenza GPL. Le caratteristiche sono equiparabili ad altri prodotti più blasonati e sinceramente a costi alti, non sempre a portata di portafoglio e soprattutto può incidere molto sul conto finale di un'aula informatica.

Non penalizza assolutamente l'utente finale e la curva di apprendimento per passare da altri prodotti è calcolabile in un tempo di trenta minuti.

La versione per LiNUX è decisamente stabile, anche se richiede comunque ingenti risorse macchina. Purtroppo sfrutta librerie proprietarie molto "pesanti" e l'integrazione in un unico ambiente (per intenderci un vero e proprio Desktop) aggrava ulteriormente il carico del computer.

Per questi motivi consiglio almeno 128 Mb di RAM e un Pentium II, magari un HD veloce. I tempi di avvio possono superare abbondantemente il minuto!

Con la versione 6.0 il pacchetto verrà proposto anche in versione per GNOME, aspettiamoci quindi grosse novità .

21.2 Installazione mono utente

Per installare il prodotto scaricare le componenti dal sito di StarOffice (‘**`http://www.sun.com/staroffice`**’). Attualemente i file sono tre (per i possessori di RedHat 7.0 nei CD c'è anche il file RPM in lingua inglese): uno è l'eseguibile, uno il database ADABAS, l'ultimo il player.

Sia il database, che il player non sono indispensabili. Ricordo che comunque in caso di installazione andrebbero installati prima dell'eseguibile vero e proprio. Questo per facilitare la configurazione automatica.

Volendo si può avere StarOffice anche con le riviste che ogni tanto lo propongono nei CD in omaggio.

Il file da eseguire è ‘`so-5_2-ga-bin-linux-it.bin`’.

```
# ./so-5_2-ga-bin-linux.it.bin [ Invio ]
```

Consiglio di copiare l'eseguibile nella directory‘ **`/tmp`**’ prima di avviare l'eseguibile.

Il programma chiederà dove installare il pacchetto, consiglio per più motrivi che vedremo in seguito la directory ‘`/opt/office52`’.

Al termine dell'installazione guidata si può lanciare l'eseguibile:

```
$ ./office52/soffice [ Invio ]
```

21.3 Installazione multi utente

LiNUX come sistema operativo è multi utente, spesso più persone lavorano su una stessa workstation, quindi risulta utile poter installare StarOffice in modo adeguato alle esigenze di un sistema operativo multiutente.

Ciò significa che oltre all'installazione generica ne andrà aggiunta un'altra per ogni utente, questo comporta l'occupazione di un paio di Megabytes di RAM nella directory home.

Lanciare il file di setup di StarOffice con l'opzione '**`/net`**':

```
# so-5_2-ga-bin-linux.it.bin /net [ Invio ]
```

Terminata l'installazione l'utente generico dovrà lanciare nuovamente il programma di installazione:

```
$ /opt/office52/program/setup [ Invio ]
```

StarOffice vorrà conferma sull'installazione tipo da adottare: scegliere l'installazione workstation. Ogni utente avrà nella propria home la directory '**`office52`**', nella quale potrà memorizzare i propri lavori e personalizzare il proprio ambiente di lavoro.

> Se si omette l'opzione /net ogni utente dovrebbe installare nella propria home il progtramma intero circa 300Mb!

21.4 Collegare un database PostgreSQL con StarOffice

StarOffice 5.2 come suite per l'automazione d'ufficio viene distribuito con StarBase, il modulo per la gestione di semplici database monoutente. L'interfaccia grafica gestisce la creazione di form per l'inserimento dati e di report per la visualizzazione delle query; combinando queste proprietà con l'affidabilità e la potenza di un database tipo PostgreSQL si possono facilmente superare le limitazioni intrinsiche di StarBase.

21.4.1 Modulo ODBC

ODBC è una specifica aperta per garantire agli sviluppatori di applicazioni un'interfaccia comune per accedera ad una sorgente di dati. Queste sorgenti comprendono server SQL e qualsiasi applicazione che si appoggia ad un driver ODBC.

Tralasceremo spiegazioni approfondite sul modulo '**`ODBC`**'; per i sistemi '**`UniX/LiNUX`**' esiste una soluzione aperta disponibile sottoforma di pacchetto in formato rpm. L'uso è simile a quello disponibile per piattaforme non UniX e l'interfaccia può appogiarsi o a KDE oppure a GNOME.

L'applicativo se non presente nella vostra distribuzione è liberamente scaricabile all'indiridizzo Internet <*http://www.unixodbc.org*>.

L'installazione e la configurazione avvengono eseguendo l'applicativo '`gODBCconfig`' se utilizziamo come interfaccia grafica GNOME. Il modulo di configurazione è molto semplice ed i passi da eseguire sono due: l'installazione di un Driver e di un Data Source (DSN). I driver contengono il codice per accedere al database di lavoro. I Data Source rappresentano un punto di accesso per prelevare i dati. I Data Source sono sia di sistema sia utente. In quest'ultimo caso ogni utente può aggiungere, modificare e rimuovere un Data Source.

21.4.1.1 System DSN

I System DSN sono creati dall'amministratore il quale ne detiene i diritti per la creazione, modifica ed eventuale eliminazione. Il System DSN verrà usato solamente se non ne esiste uno di analogo creato dall'utente, il quale quindi ne detiene la precedenza. Ogni data source viene condiviso con tutti gli utenti.

21.4.1.2 User DSN

Lo User DSN rappresentano i data source personale per ogni utente. L'utente ha la possibilità di crearli, modificarli ed evenutalmente eliminarli. I dati sono separati tra gli utenti e questo permette sicurezza e flessibilità di accesso.

21.4.1.3 Drivers

Sono l'anello di comunicazione tra il software ed il database di lavoro. Generalmente vengono distribuiti con il database stesso dal produttore, tuttavia con unixODBC ne vengono forniti alcuni. Solo l'amministratore può installarli e rimouverli.

La procedura di instalazione è molto semplice; cliccare sul tag drivers come da figura 21.1. Quindi cliccare sul tasto "Add" e completare come da figura 21.2.

Figura 21.1.

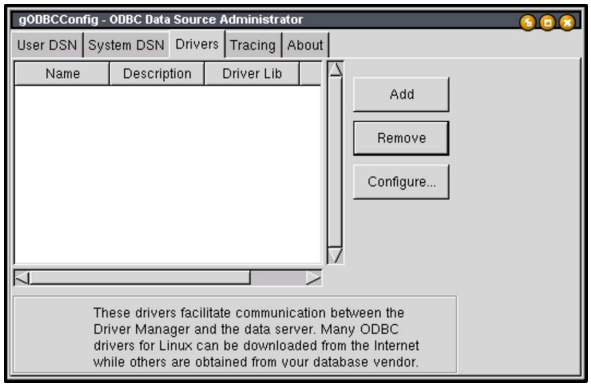
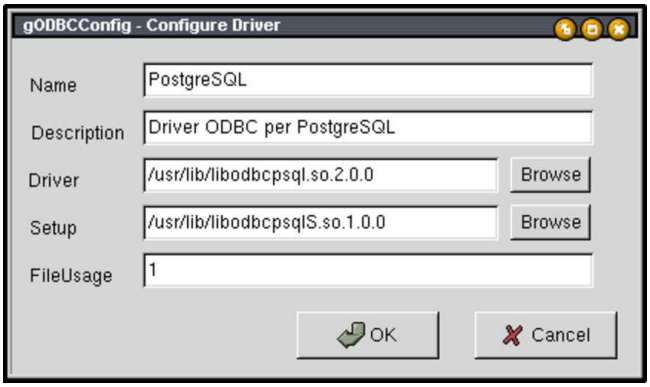


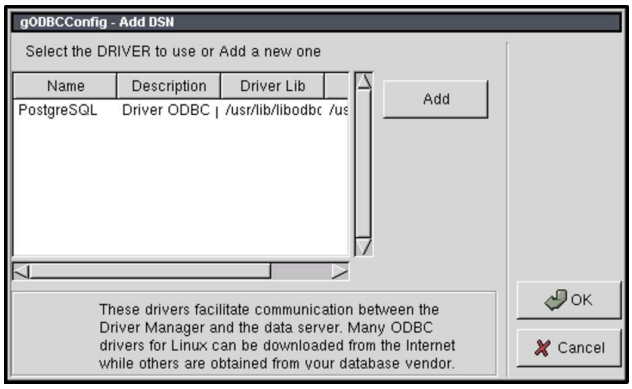
Figura 21.2.



21.4.1.4 Installazione di un System DSN o di uno User DSN

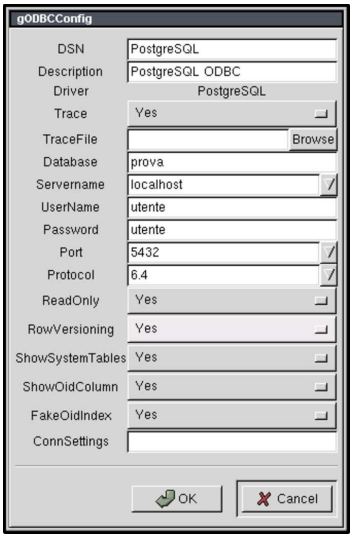
Scegliere il tag System DSN quindi cliccare sul tasto "Add". Scegliere il driver da utilizzare come da figura 21.3. Quindi cliccare sul tasto: "OK"

Figura 21.3.



Completare la finestra di dialogo con almeno il nome del database la UserName e la Password (vedi figura 21.4).

Figura 21.4.



Per lo User DSN le operazione da eseguire sono le medesime. Ad operazione terminata è possibile lanciare StarOffice 5.2 e creare un nuovo DataBase con driver ODBC.