Installare e configurare servizi
Installazione di applicazioni client/server - Introduzione a Windows 2000 printing

Introduzione a *Windows* 2000 *printing*

- **Terminologia**.
- **Prerequisiti**.

Microsoft Windows 2000 offre all'amministratore tutta una serie di tecnologie e strumenti che rendono molto semplice e flessibile la gestione e configurazione di una stampante condivisa.

Una volta che sia stato configurato un *server* di stampa è possibile configurare *client computer* che eseguono *Windows* 95, *Windows* 98, o *Microsoft Windows* NT 4.0 oltre che, ovviamente, *Microsoft Windows* 2000.

Prima di vedere però come condividere una stampante, come assegnare i permessi di stampa, come installare i *client* ed altri settaggi avanzati, bisogna innanzitutto analizzare la terminologia usata ed i prerequisiti necessari affinché tutto ciò sia realizzabile.

Introduzione a Windows 2000 printing - Terminologia

I seguenti termini definiscono quello che un ambiente di stampa in *Windows* 2000:

- ***Print device***. La periferica *hardware*, cioé la stampante fisica. *Windows* 2000 distingue le due seguenti tipologie di *Print Device*:
  - ***Local print devices***. Periferica *hardware* connessa ad una porta locale del *server* (COM, LPT, ...).
  - ***Network-interface print devices***. Periferica *hardware* connessa al *print server* tramite la rete piuttosto che tramite una porta locale. Richiede una propria scheda di rete ed un proprio indirizzo di rete.
- ***Printer***. Il *software* tramite il quale il sistema operativo dialoga con la stampante.
- ***Print server***. Il *computer* su cui il *printers* e i *drivers* per i *client* risiedono. Tale macchina riceve le richieste di stampa dei *client*, e le processa prima di inviarle alla stampante. Ovviamente si incarica anche di verificare che il *client* abbia opportuni permessi.
- ***Printer driver***. Uno o più *files* che contengono le informazioni che permettono a *Windows* di convertire i comandi di stampa in comandi comprensibili alla specifica stampante (*rendering*).

Introduzione a Windows 2000 printing - Prerequisiti

Di seguito i prerequisiti *hardware* richiesti:

- Una macchina che svolga il ruolo di *Print Server* eventualmente dedicato. Tale macchina può essere:
  - *Windows* 2000 *Server*, *Windows* 2000 *Advanced Server*, o *Windows* 2000 *Datacenter Server*. In tal caso è possibile supportare oltre che un elevato numero di connessioni comtemporanee, anche *clients Macintosh*, *UNIX*, e *NetWare*.
  - *Windows* 2000 *Professional*. In tal caso il numero di connessioni contemporanee è 10, compresi i *client UNIX*.
- Memoria RAM a sufficienza per processare le richieste di stampa.
- Spazio disco a sufficienza per memorizzare i documenti. Bisogna garantire che il *print server* abbia spazio disco a sufficienza, per poter memorizzare i documenti che devono essere stampati.

Aggiungere un printer

- **Aggiungere e condividere un printer per un Local Print Device**.
- **Configurare i clients**.

Quando si installa e condivide una stampante è possibile scegliere tra un *printer* che faccia riferimento ad un *Local Print Device* ed un *printer* che faccia riferimento ad un *Network Print Device*. Questa seconda soluzione è quella ideale in ambienti che prevedono un elevato numero di *client*.

Quando si installa un *printer*, bisogna inoltre verificare che tutti i *client* siano propriamente configurati, in particolare che essi abbiano a disposizione i *driver* relativi alla loro versione del sistema operativo.

Ricordiamo che i *driver* sulla macchina *client* sono necessari affinché quest'ultima possa permettere alle applicazioni di sapere come sarà la stampa del documento (*WYSiWYG, What You See Is What You Get*) e sia possibile configurare le varie proprietà della stampante. Invece il *rendering* del documento avviene invece sul *print server*.

Aggiungere e condividere un printer per un local print device

Per aggiungere e condividere una stampante bisogna utilizzare un *account* utente che appartenga al gruppo *Administrator* su quello che sarà il *print server*.

Per installare e condividere un *Printer* relativo ad un *Local Print Device* selezionare le seguenti opzioni durante l'esecuzione del *wizard Add Printer* che troviamo in `Start\Setting\Printers`:

- ***Local printer***. Selezionando questa opzione stiamo dichiarando di voler configurare il *Print Server*.
- ***Use the following port***. La porta sul *Print Server* a cui è connessa la *print Device*. È possibile selezionare una delle porte esistenti o aggiungerne una nuova, come ad esempio una porta *hardware* non-standard.
- ***Manufacturers e Printers***. Selezionare il *Printer Driver* corretto per la stampante. Se la nostra stampante non è individuabile utilizzando le diverse possibilità in *Manufacturers* e *Printers*, bisogna preoccuparci di fornire tale *driver* o quantomeno uno compatibile.
- ***Printer name***. Un nome che identifica la stampante per gli utenti. Tale nome può essere lungo fino a 31 caratteri ed ovviamente si consiglia che sia il più possibile intuitivo ed esplicativo.
- ***Default printer***. Se questa stampante è la stampante di *default* per le varie applicazioni. Durante l'installazione della prima stampante, tale opzione non viene proposta.
- ***Shared as***. Il nome di condivisione che gli utenti, se opportunamente autorizzati, possono utilizzare per connettersi alla stampante tramite la rete. Tale nome viene visualizzato quando l'utente sfoglia le risorse del *server*. Assicuriamoci che il nome scelto sia correttamente visualizzabile da tutti i *client* (alcuni *client* potrebbero supportare solo nomi in formato 8.3, e dunque per tali *clients* il nostro nome verrà troncato per rispettare tale formato).
- ***Location and Comment***. Informazioni riguardo la stampante.
- ***Do you want to print a test page?*** Permette di verificare che l'installazione sia terminata cons successo. In tal caso selezionare *Yes* nella finestra di dialogo che viene proposta. In caso contrario selezionare No.

Configurare i clients

I seguenti *clients* scaricano automaticamente i *drivers*:

- *Windows* 95.
- *Windows* 98.

- *Windows* NT.

I *client* che eseguono altri sistemi operativi *Microsoft* richiedono l'installazione dei *drivers*.

I *client* non-*Microsoft* richiedono l'installazione di:

- *driver* sul *client*.
- Appropriato servizio sul *server*.

Dopo aver installato e condiviso la stampante sul *Print Server*, è possibile configurare i *computer clients* affinché possano utilizzare tale stampante.

La procedura di installazione del *client* varia in base al sistema operativo installato, ma in ogni caso è richiesta la presenza del *printer driver* sulla macchina *client*.

Distinguiamo i seguenti casi, oltre ai *client Microsoft Windows* 2000:

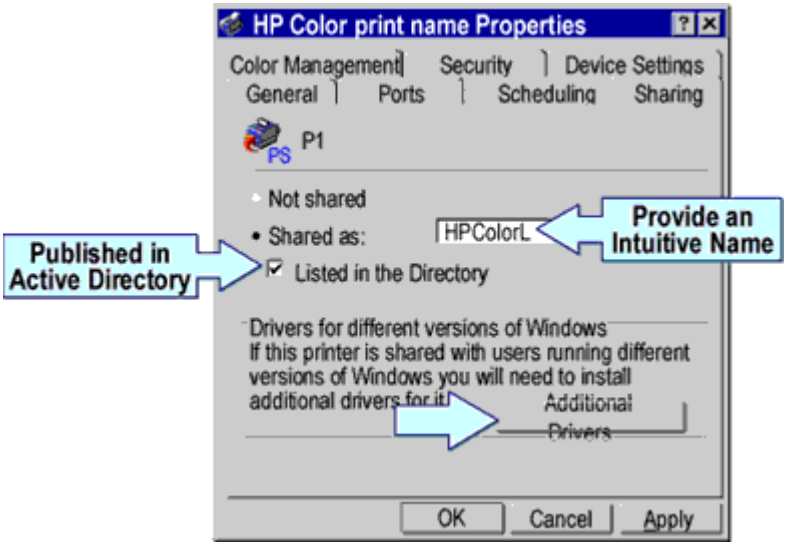
- ***Client Windows* 95, *Windows* 98, o *Windows* NT 4.0**. Gli utenti di tali *client* devono semplicemente connettersi alla stampante condivisa. Essi scaricheranno automaticamente l'appropriato *Printer Driver*, che sarà stato reso disponibile sul *Print Server*. Inoltre, solo per i *client Windows* NT 4.0, se aggiorniamo la copia del *Printer Driver* memorizzato sul *Print Server*, al prossimo utilizzo della stampante sul *client* essi scaricheranno la copia aggiornata del *Printer Driver*. I *client Windows* 95, *Windows* 98 dovranno provvedere invece manualmente.
- ***Client Microsoft* Non *Windows* 95, *Windows* 98, o *Windows* NT 4.0**. In questo caso, affinche sia possibile utilizzare la stampante condivisa, tali *client* devono connettersi ad essa e bisogna installare manualmente i *Printer Driver* opportuni.
- ***Client* Non *Microsoft***. In questo caso, affinche sia possibile utilizzare la stampante condivisa, tali *client* devono connettersi ad essa, bisogna installare manualmente i *Printer Driver* opportuni e bisogna installare dei servizi aggiuntivi sul *Print Server*.
    - ***Client Machintosh***. Bisogna installare il servizio *Macintosh Services for Macintosh*, compreso in *Windows* 2000 ma non installato di *default*.
    - ***Client UNIX***. Bisogna installare il servizio *UNIX TCP/IP Printing*, compreso in *Windows* 2000 ma non installato di *default*, che contiene il *Line Printer Daemon* (LPD).
    - ***Client Novell Netware***. Bisogna installare il servizio *File and Print Services for NetWare*, non compreso in *Windows* 2000.

Configurare un network printer

- **Condividere un printer esistente**.
- **Definire un printer pool**.
- **Impostare le priorità di un printer**.
- **Impostare i permessi di stampa**.

Concludiamo ora le nostre osservazioni inerenti la stampa in *Microsoft Windows* 2000, analizzando come condividere un *printer* esistente, definendo quali sono i permessi di stampa ed analizzando, infine, due configurazioni avanzate che riguardano la realizzazione di un *Pool* di stampanti per realizzare una configurazione performante e *fault tolerant*, e la configurazione della priorità associata ad un *printer* per poter privilegiare la stampa di documenti critici.

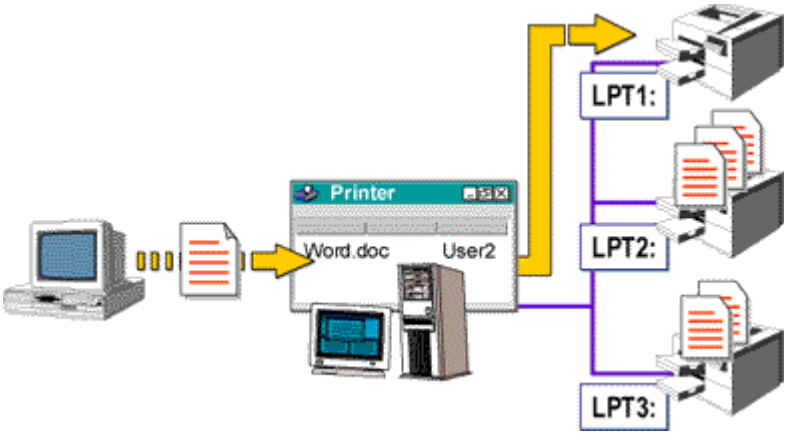
Condividere un printer esistente



Per condividere una stampante già installata su un *server*:

- Accedere alla cartella *Start\Setting\Printers*.
- Selezionare la stampante desiderata.
- Cliccare con il tasto destro e scegliere *Sharing*.
- Specificare le seguenti opzioni:
  - Selezionare *shared AS* e specificare il nome di condivisione.
  - Decidere se pubblicare la stampante in *Active Directory* selezionando *Listed in The Directory* (di *default* è selezionato).
  - Aggiungere *Printer Driver* aggiuntivi per *client Windows* 95, *Windows* 98, o *Windows* NT 4.0 selezionando *Additional Drivers* .

Definire un printer pool



Un *Printer Pool* consiste di un solo *Printer* che è connesso a più di una *Print Device* tramite più porte del *Print Server*. Le *Print Device* possono essere sia *Local* che *Network Interface*, e non necessariamente devono essere identiche, ma tutte compatibili con lo stesso *Printer Driver*.

L'utente invierà le sue stampe all'unico *Printer* che provvederà e distribuirle sulle varie *Print Device* in maniera del tutto trasparente al *client*.

L'utilizzo di un *Printer Pool* presenta i seguenti vantaggi:

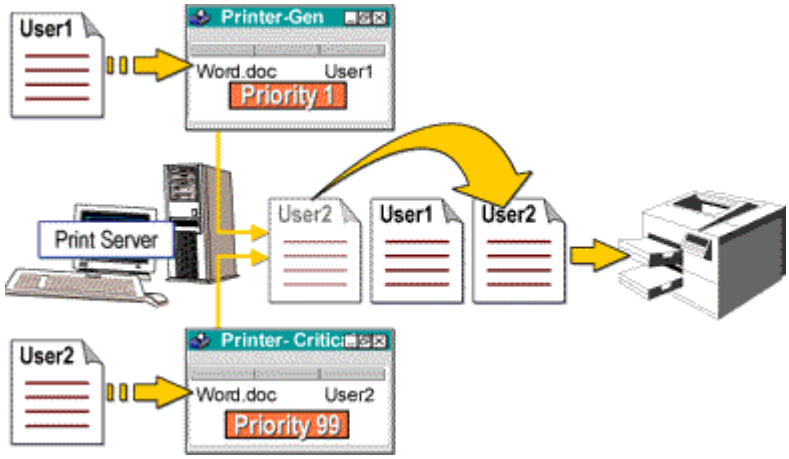
- Diminuisce il tempo di permanenza dei documenti sul *Print Server*.
- Semplifica l'amministrazione poiché consente di amministrare più *Print Device* tramite un

unico *printer*.
- Crea una configurazione *Fault Tolerant*.

Per creare un *Printer Pool*:

- Accedere alla cartella *Start\Setting\Printers*.
- Selezionare la stampante desiderata.
- Cliccare con il tasto destro e scegliere *Properties*.
- Selezionare la scheda *Ports*
- Selezionare l'opzione *Enable printer pooling*
- Selezionare le porte a cui le *Print Device* sono state connesse.

Impostare le priorità di un printer



Installare su un *Print Server* più *printer* corrispondenti alla stessa *Print Device*, ed assegnare ad ogni *Printer* un diverso livello di priorità permette di stabilire dei livelli di priorità tra dicumenti stampati effettivamente dalla stessa *Print Device*.

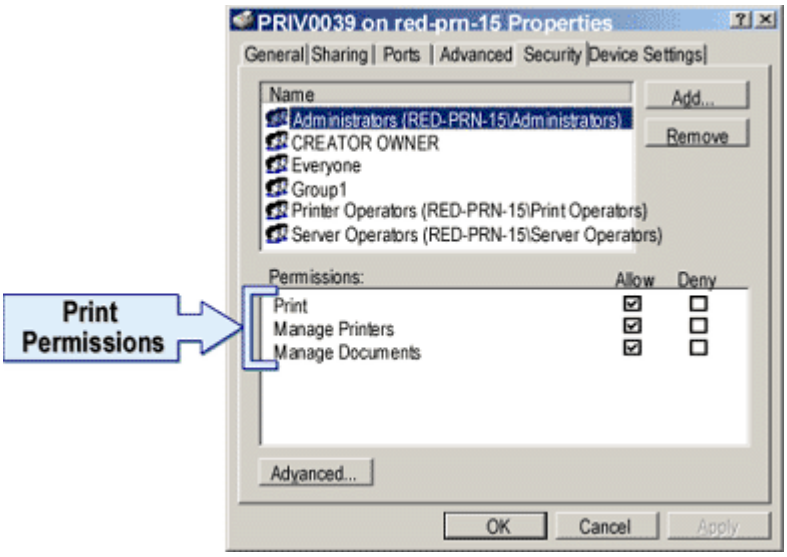
Per impostare *Printers* con priorità diverse:

- Installare più volte lo stesso *Printer* e farlo puntare sempre alla stessa porta fisica.
- Impostare diversi livelli di priorità per ognuno dei *Printer*.
- Condividere i vari *printer* ed assegnare i permessi di condivisione in maniera tale che ogni utente acceda al *printer* con il livello di priorità più opportuno.

Per impostare invece la priorità di un *Printer*:

- Accedere alla cartella *Start\Setting\Printers*.
- Selezionare la stampante desiderata.
- Cliccare con il tasto destro e scegliere *Properties*.
- Selezionare la scheda *Advanced* ed impostare la priorità oppoortuna (da 1 a 99) utilizzando il cursore in *Priority*.

Impostare i permessi di stampa



Esistono tre tipi di permessi per quel che concerne la condivisione di una stampante:

- ***Print***. Stampare documenti, gestire i propri documenti nella coda di stampa, connettersi alla stampante.
- ***Manage Documents***. Gestire tutti i documenti nella coda di stampa oltre ai permessi associati a *Print*.
- ***Manage Printer***. Condividere la stampante, interrompere la condivisione, eliminare la stampante, modificare i permessi oltre ai permessi associati a *Manage Documents*.

Di *default* i gruppi *Administrators* e *Print Operators* hanno il permesso *Manage Printer*, mentre il gruppo *Everyone* ha *Print* ed il proprietario *Manage Documents*.

Per modificare i permessi di condivisione o aggiungerne altri:

- Accedere alla cartella *Start\Setting\Printers*.
- Selezionare la stampante desiderata.
- Cliccare con il tasto destro e scegliere *Properties*.
- Selezionare la scheda *Security*.
- Utilizzare *Add* per aggiungere ulteriori utenti e gruppi ed utilizzare in *Permission* gli opportuni *check box* per assegnare (*Allow*) o negare (*Deny*) un permesso.
- Utilizzare il pulsante Remove per rimuovere gruppi om utenti con i relativi permessi.

DHCP: Nuove Funzionalità

DCHP (*Dynamic Host Configuration Protocol*) è il protocollo ed il relativo servizio che permette la gestione centralizzata ed automatica dei parametri di configurazione relativi al protocollo TCP/IP e necessari ai vari *hosts* presenti sulla rete.

In *Windows* 2000 sono state aggiunte al DHCP tutta una serie di nuove funzionalità, tra cui ricordiamo:

- **Rilevamento di Server DHCP non autorizzati**. È possibile impedire che sulla rete siano attivi *server* DHCP non autorizzati che potrebbero portare all'assegnazione di indirizzi IP duplicati.
- **Integrazione con il DNS** (Domain Name System). Quando il *server* DHCP assegna un indirizzo IP ad un *client* provvede anche alla registrazione parziale o totale delle informazioni necessarie sul *server* DNS se quest'ultimo è abilitato agli aggiornamenti dinamici.

- **Supporto esteso agli scope**. Oltre che *scope* tradizionali è possibile definire **Superscopes** e **Scope Multicast** .
- **Supporto delle Option Classes**. Tramite le *Option Class* è possibile gestire in maniera estremamente flessibile l'assegnazione dei parametri opzionali ai vari *client* basandosi su caratteristiche quali il tipo di sistema operativo o definendone di personalizzate.
- **Assegnazione Automatica degli Indirizzi IP (APIPA)**. Ai *client* DHCP che non riescono momentaneamente a raggiungere il *server* DHCP viene assegnato un indirizzo IP momentaneo appartenente ad un insieme riservato, non utilizzato in Internet.
- *Funzionalità avanzate di monitoraggio e statistica*.
- *Possibilità di ri-registrare i record sul server DNS*. Quando il *client* DHCP rinnova il proprio indirizzo allo scadere dell'intervallo di '*lease*', tale indirizzo viene anche nuovamente ri-registrato sul *server* DNS. Ciò garantisce un certo livello di *fault tolerance* rispetto ai *client* configurati staticamente che effettuano la registrazione automatica sul DNS solo al momento della partenza.

Autorizzare un DHCP Server in Active Directory

Con le precedenti implementazioni del DHCP chiunque poteva installare un *server* DHCP ed inserirlo su una rete, provocando l'assegnazione di indirizzi IP non controllati ed eventuali fenomeni di indirizzi duplicati.

In *Windows* 2000 è necessario autorizzare un *server* DHCP in *Active Directory* prima che tale *server* possa funzionare correttamente.

Questo poiché quando il servizio DHCP parte, contatta *Active Directory* per verificare se appartiene alla lista dei *server* autorizzati. In seguito a tale controllo possono verificarsi le seguenti condizioni:

- Il *server* DHCP è autorizzato e dunque il servizio parte correttamente.
- Il *server* DHCP non è autorizzato e dunque il servizio registra un errore nel log di sistema e non risponde a nessuna delle richieste dei *client*

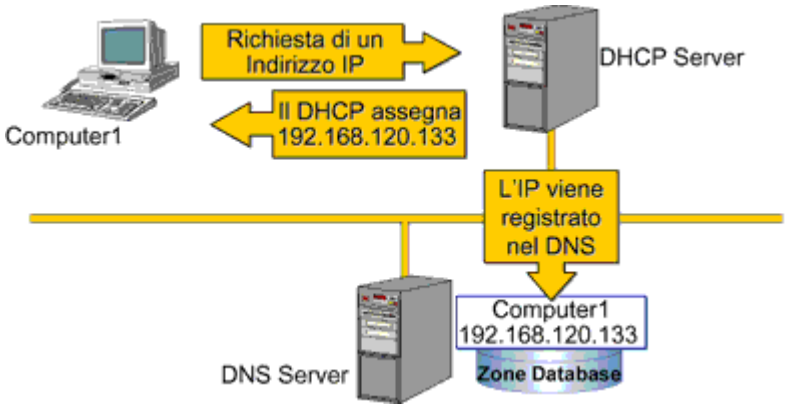
Periodicamente il DHCP continua a controllare se sono avvenute modifiche all'elenco dei *server* autorizzati.

Per autorizzare un *server* DHCP in *Active Directory*, utilizzare il *tool* DHCP presente in *Start\Programs\Administrative Tools* e facendo click con il tasto destro su DHCP scegliere *Manage authorized servers*, selezionare *Authorize* ed inserire il nome o l'indirizzo IP del *server* da autorizzare.

Per autorizzare un *server* DHCP bisogna appartenere al gruppo *Enterprise Admins* presente nel dominio radice della foresta.

Aggiornamento Dinamico del Server DNS

Di *default*, un *server* DHCP *Windows* 2000 è abilitato ad eseguire l'aggiornamento di un *server* **DNS** per il quale sia attivo l'aggiornamento dinamico. Di conseguenza, il *server* DHCP aggiorna automaticamente il record PTR per i *client Windows* 2000 che provvedono invece all'aggiornamento del record di tipo A.



Per configurare il *server* DHCP affinché aggiorni automaticamente il *server* DNS, utilizzare il *tool* DHCP presente in *Start\Programs\Administrative Tools* e facendo click con il tasto destro sul *server* in questione scegliere *Properties* e selezionare la scheda DNS.

Sono disponibili le seguenti opzioni:

- ***Automatically update DHCP client information in DNS***. Abilita la funzionalità di aggiornamento dinamico del DNS per il *server* DHCP. Se tale opzione non è selezionata, nessun'altra opzione risulta essere disponibile.
- ***Update DNS only if DHCP client requests***. Aggiorna il record A ed il record PTR in base a quelle che sono le richieste del *client* durante il processo di assegnazione dell'indirizzo IP. Questa opzione è selezionata di *default* e determina che il *client Windows* 2000 provvede alla registrazione del record A ed il *server* DHCP alla registrazione del record PTR. Per *client* diversi da *Windows* 2000 non avviene nessun aggiornamento di tipo dinamico.
- ***Always update DNS***. Indipendentemente dalla richiesta del *client* il DHCP si fa carico del completo processo di registrazione sul *server* DNS.
- ***Discard forward (name-to-address) lookups when lease expires***. Rimuove i record A e PTR relativi ad un *client* per il quale un certo indirizzo IP non è più valido.
- ***Enable updates for DNS clients that do not support dynamic update***. Abilita l'aggiornamento dinamico sia del record A che del record PTR da parte del *server* DHCP per i *client* DHCP non *Windows* 2000.

Se i *client* sono *Windows* 2000 ed il *server* e *Microsoft Windows* NT 4.0 DHCP saranno i *client Windows* 2000 a dover aggiornare sia il record A che il record PTR.

Configurare i DHCP Scopes in Windows 2000

Windows 2000 estende le funzionalità di un *server* DHCP supportando oltre allo *scope* tradizionale anche i cosiddetti **Superscopes** e **Scope Multicast** .

Tramite queste due nuove funzionalità è possibile assegnare, ad esempio, indirizzi IP a reti fisiche che contengono più di una sottorete logica.

Oltre a queste nuove tipoligie di *scope* è comunque presente un *wizard* che semplifica estremamente il processo di creazione di un qualsiasi tipo di *scope*.

Configurare uno Scope

Windows 2000 utilizza il *wizard* '*Create Scope*' per rendere più semplice il processo di creazione di uno *scope*.

Per avviare tale *wizard* utilizzare il *tool* DHCP presente in *Start\Programs\Administrative Tools* e fare doppio click sul *server* che ospiterà lo *scope*. Fare click con il tasto destro sul *server* e selezionare *New Scope*.

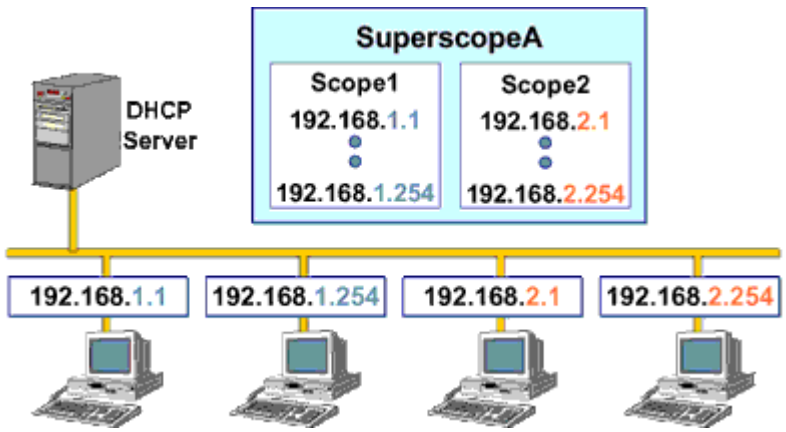
Verranno richieste le seguenti informazioni:

- Nome dello *scope*.
- *Range* di indirizzi IP da assegnare.
- *Subnet mask* da associare agli indirizzi definiti nel *range*.
- Eventuale intervallo di esclusione del *range*.
- Durata dell'assegnazione.
- Parametri DHCP comuni, tra cui:
  - Indirizzo dei *gateways* di *default*.
  - Nome del dominio DNS ed indirizzo dei *server* DNS.
  - Indirizzo IP di ogni *server WINS*.

Di *default* uno *scope* non è attivo. Per attivarlo, cliccare con il tasto destro sullo *scope*, selezionare *All Tasks* e quindi *Activate*.

Configurare un Superscope

In una rete basata su *Windows* NT 4.0, gli indirizzi IP assegnati ai *client* DHCP dovevano essere definiti in modo che ad ogni sottorete logica corrispondesse una singola sottorete fisica e viceversa.



L'implementazione del DHCP in *Windows* 2000 prevede, tramite i *Superscopes*, la possibilità che ad un'unica sottorete fisica siano associate più sottoreti logiche *Windows* NT 4.0 *Service Pack* 2 supporta i *Superscopes*.

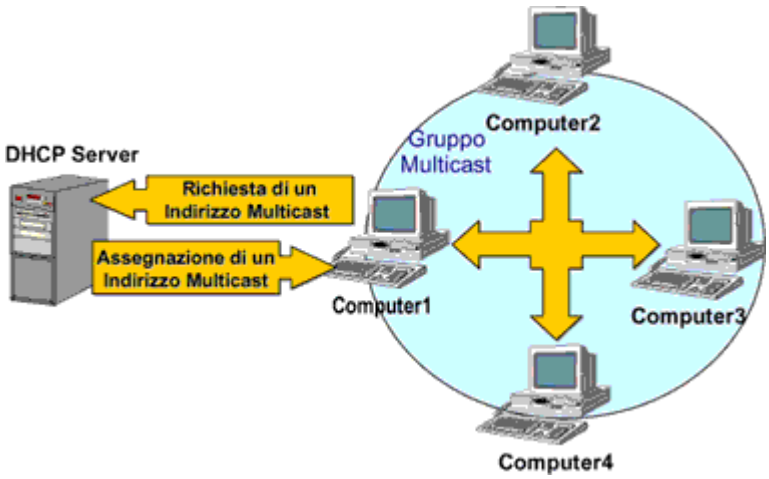
Ad esempio, è possibile utilizzare i *Superscopes* nelle seguenti situazioni:

- Bisogna aggiungere ad una sottorete più *host* di quanti pianificato in fase di progettazione e configurazione del *server* DHCP.
- Bisogna sostituire un *range* di indirizzi esistente con un nuovo *range*.

Per configurare i *Superscopes* utilizzare il *tool* DHCP presente in *Start\Programs\Administrative Tools* e fare doppio click sul *server* che ospiterà il *Superscopes*. Fare click con il tasto destro sul *server* e selezionare *New Superscope*. Verranno richiesti il nome e quali sono gli *scope* esistenti da includere.

Configurare un Multicast Scope

Ricordiamo che gli indirizzi IP il cui primo ottetto varia in un intervallo che va da 224 a 239 (Classe D) sono i cosiddetti Indirizzi *Multicast*, cioé quella classe di indirizzi IP che vengono utilizzati per indirizzare, non un singolo *host*, ma un insieme di *host* ben definito. Tali *host* hanno la caratteristica di appartenere allo stesso Gruppo *Multicast* identificato appunto da uno di tali indirizzi. Tramite tale indirizzo è possibile raggiungere con un solo messaggio diretto un numero di macchine maggiore di uno.



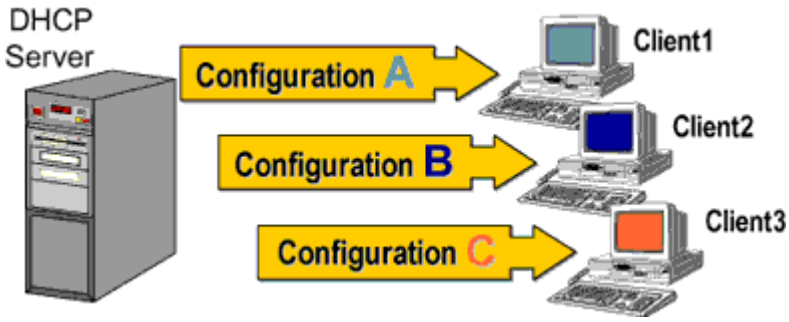
Tipicamente le applicazioni basate sull'utilizzo di flussi audio e videoconferenza si basano su tecnologie *multicast*.

Tramite la definitione di *Multicast scopes* è possibile automatizzare e gestire in maniera centralizzata l'assegnazione di tale tipologia di indirizzi ai *client* che ne necessitano.

Per configurare i *Multicast scopes* utilizzare il *tool* DHCP presente in *Start\Programs\Administrative Tools* e fare doppio click sul *server* che ospiterà il *Superscopes*. Fare click con il tasto destro sul *server* e selezionare *New Multicast Scope*.

Verranno richiesti il nome e il *range* di indirizzi *multicast*. Attivare lo *scope* quando verrà richiesto.

Le Option Classes



In una rete basata su *Microsoft Windows* NT 4.0, il *server* DHCP fornisce gli stessi parametri di configurazione opzionali per tutti i *client* di uno stesso *server* o, al massimo, appartenenti allo stesso *scope*. Inoltre, utilizzando le *reservation* è possibile specificare parametri di configurazione particolari per uno specifico *client* utilizzando il suo *MAC address*.

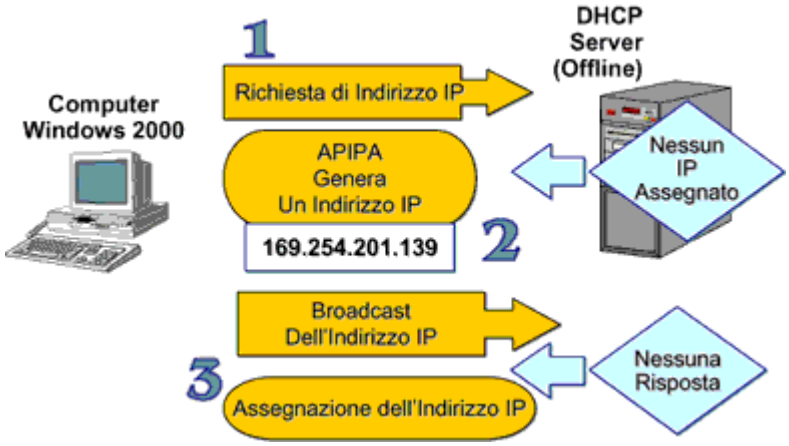
Al di là di questi ambiti (*server*, *scope*, singola macchina) non è possibile definire un qualsivoglia sottoinsieme di macchine, basandosi su una qualche caratteristiche ad esse comune, e specificare delle opzioni riservate a tali macchine.

L'implementazione del DHCP in *Windows* 2000, supporta le *Option Classes*, che permettono, appunto, di definire insiemi di macchine in base ad una qualsiasi caratterizzazione, e riservare a tale insieme parametri di configurazione specifici. Esempi di *Option Classes* possono essere tutte le macchine che abbiano una certa versione del sistema operativo, o le macchine di un certo tipo (*desktop*, *laptop*) o tutte le macchine abilitate all'accesso ad Internet, eccetera ...

Windows 2000 supporta due tipi di *Option Classes*:

- ***Vendor-defined classes***. Permettono di identificare un *client* dalla versione del suo sistema operativo
- ***User-defined classes***. Permettono di identificare i *client* in base ad una qualche caratteristica personalizzata, ad esempio *desktop* o *laptop*, necessità di accedere ad Internet, locazione geografica e quant'altro possa risultare conveniente.

Automatic Private IP Addressing (APIPA)



Windows 2000 supporta un nuovo meccanismo per l'assegnazione automatica di indirizzi IP nel caso di reti LAN di piccole dimensioni.

Questo meccanismo, denominato '*Automatic Private IP Addressing* (APIPA)', permette di assegnare un indirizzo IP ad un *host* senza ricorrere alla configurazione statica (manuale) e nel contempo senza ricorrere ad un *server* DHCP.

Il suo funzionamento è il seguente:

- Quando *Windows* 2000 parte, il TCP/IP prova a cercare un *server* DHCP che possa fornire un indirizzo IP valido.
- In assenza del *server* DHCP, il *client* non resterebbe sprovvisto di indirizzo IP. Se invece APIPA risulta abilitato, all'*host* viene assegnato un indirizzo IP del tipo 169.254.x.y con *subnet mask* 255.255.0.0.
- Dopo la generazione di tale indirizzo il *client* verifica tramite un *broadcast* che tale indirizzo non sia in uso ed in questo caso lo usa come suo, altrimenti ne genera un altro ed il processo si ripete.
- Il *client* continua ad utilizzare tale indirizzo finché non scopre che un DHCP ha un indirizzo disponibile e valido (la ricerca di tale DHCP viene effettuata ogni 5 minuti).

Per implementare tale funzionalità *Microsoft* si è riservata per l'utilizzo tramite APIPA il *range* di indirizzi da 169.254.0.1 a 169.254.255.254.

Purtroppo APIPA è in grado di generare l'indirizzo IP e la *subnet mask* ma non ogni altro eventuale

parametro (indirizzo del *gateway*, indirizzo dei DNS, indirizzo dei *WINS* ...) di cui il *client* dovesse aver bisogno. Dunque, gli *host* che hanno ricevuto un indirizzo tramite APIPA possono comunicare solo con *host* dello stesso segmento che abbiano ricevuto l'indirizzo IP nello stesso modo.

APIPA è abilitata di *default*. Per disabilitare APIPA, basta aggiungere il valore `IPAutoconfigurationEnabled` di tipo `REG_DWORD` con valore 0 al *path* di registro: `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip\Parameters\Interfaces\_scheda_di_rete`

Configurare un Web Server

Il processo di creazione di un *Web Server* è un processo molto simile a quello relativo alla creazione di un *file server*.

Come nel caso di ogni altra tipologia di *server* bisogna innanzitutto installare il sistema operativo, decicere l'appartenenza ad un gruppo di lavoro (*stand-alone server*) o ad un dominio (*member server*), ed organizzare in maniera opportuna i dischi.

Il passo successivo consiste nell'installare e configurare in maniera appropriata i servizi relativi alla funzionalità di *Web Server*.

Ricordiamo che tramite un *Web Server* un *client* accede a *file* che costituiscono le pagine *Web*, ma anche a complesse applicazioni *client/server* basate su *database*.

Windows 2000 *server* comprendono un *Web Server* che va sotto la denominazione di *Internet Information Services* (IIS) 5.0.

IIS viene installato di *default* come servizio di rete durante l'installazione di *Windows* 2000 e permette di supportare anche configurazioni di *Web Server* abbastanza complesse (*Server Web* Virtuali, Cartelle Virtuali, ...).

Per poter installare e configurare con successo IIS 5.0 su un *member server*, si necessita di:

- *Transmission Control Protocol/Internet Protocol* (TCP/IP).
- Almeno un indirizzo IP.
- *Domain name*. Se si vuole accedere il *server* utilizzando un *Fully Qualified Domain name* (FQDN) piuttosto che il suo indirizzo IP, c'è bisogno di installare e configurare opportunamente un *server* DNS.

Bisogna poi ovviamente provvedere alla creazione e configurazione del *Server Web* tramite la definizione della sua *Directory* Radice e la definizione di tutte le strategie atte a permettere un accesso controllato e sicuro (autenticazione, cifratura ...) al *server* in questione.

Linux - DHCP server

Il protocollo DHCP permette di assegnare in modo dinamico gli indirizzi IP alle macchine della rete locale. Vi sono diversi *server* DHCP disponibili per i sistemi *UNIX*, sia *software* commerciali che *Open Source*. Su *Red Hat Linux* viene utilizzato DHCPd di *Paul Vixie*.

La configurazione della parte *client* del protocollo su *Linux* è stata descritta nell'unità didattica relativa alla configurazione di rete.

Per il funzionamento del protocollo DHCP è necessario che nel *kernel* sia configurato il supporto per il *multicast*. È possibile vedere se una interfaccia è correttamente configurata utilizzando il comando

`ifconfig`:

```
# ifconfig eth0
eth0 Link encap:10Mbps Ethernet HWaddr 00:C0:4F:D3:C4:62
inet addr:183.217.19.43 Bcast:183.217.19.255 Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
RX packets:2875542 errors:0 dropped:0 overruns:0
TX packets:218647 errors:0 dropped:0 overruns:0
Interrupt:11 Base address:0x210
```

Se l'etichetta *MULTICAST* non è presente, si deve ricompilare il *kernel* aggiungendo il supporto per il *multicast*. Alcuni *client* DHCP richiedono che nel *server* esista una *route* per la rete 255.255.255.255. Questa può essere difficoltosa da creare, in quanto alcune versioni del comando *route* per *Linux* insistono nel cambiare l'indirizzo 255.255.255.255 nell'indirizzo di *broadcast* della rete locale. Per verificare se la versione in uso è affetta da questo problema è sufficiente scrivere il comando

```
# route add -host 255.255.255.255 dev eth0
```

se viene restituito un errore 255.255.255.255: *Unknown host*, è possibile provare ad inserire in `/etc/hosts` la seguente linea

```
255.255.255.255 all-ones
```

ed aggiungere la *route* utilizzando la linea

```
/sbin/route add -host all-ones dev eth0
```

Per rendere permanente la modifica bisognerà ovviamente inserire tale comando in uno *script* di avvio.

Il *server* DHCPd viene generalmente eseguito in modalità *standalone*, lanciandolo da uno degli *script* di avvio in `/etc/rc.d`, dopo averlo configurato mediante il *file* `/etc/dhcpd.conf`. Per associare il servizio ad una interfaccia diversa da quella di *default* (`eth0`), è sufficiente scriverne il nome come parametro nella linea di comando (ad esempio: `/usr/local/bin/dhcpd eth1`).

In ambiente KDE è possibile utilizzare per la configurazione l'interfaccia grafica `kcmdhcpd`.

Il seguente esempio mostra come assegnare ai *client* degli indirizzi IP scelti negli intervalli 192.168.1.10-192.168.1.100 oppure 192.168.1.150-192.168.1.200, con periodo di validità (*lease*) di 600 secondi come *default* e di 7200 secondi come periodo massimo. Vengono inoltre passate ai *client* le informazioni relative alla *netmask*, all'indirizzo di *broadcast*, al *gateway* e ai *server* DNS e *WINS* (*netbios-name-servers*).

```
# Sample /etc/dhcpd.conf
default-lease-time 600;
max-lease-time 7200;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.1.255;
option routers 192.168.1.254;
option domain-name-servers 192.168.1.1, 192.168.1.2;
option netbios-name-servers 192.168.1.1;
option domain-name mydomain.org;

subnet 192.168.1.0 netmask 255.255.255.0 {
range 192.168.1.10 192.168.1.100;
```

```
range 192.168.1.150 192.168.1.200;
}
```

È possibile assegnare ad un *client* con dato *MAC address* un indirizzo IP fisso mediante una linea del tipo:

```
host pc222 {
hardware ethernet 08:00:2b:4c:59:23;
fixed-address 192.168.1.222;
}
```

La lista degli indirizzi assegnati ai vari *client* viene mantenuta nel *file* `/var/state/dhcp/dhcpd.leases`. Essa è in formato testo e pertanto mediante semplici *script* può essere utilizzata per tenere sincronizzata la tabella dei nomi locali nel DNS.

Per ulteriori informazioni si consulti il documento DHCP-Mini-*HOWTO*.

Linux - Servizi di stampa

Nei sistemi *UNIX* lo *spooler* lpd può essere configurato in modo da accettare, alla porta TCP/512, richieste di stampa provenienti da *client* remoti. Il controllo degli accessi viene gestito mediante il *file* `/etc/hosts.lpd`. Per configurare una macchina *Linux* in modo da spedire una stampa ad un lpd remoto è necessario definire una stampante di rete in `/etc/printcap`, specificando in modo opportuno il campo `rm` (*remote machine*). Un esempio è il seguente:

```
net|laserjetlp:\
:sd=/var/spool/lpd/net:\
:mx#0:\:sh:\
:rm=printsrv.mydomain.com:\
:rp=lp:\
:if=/var/spool/lpd/net/filter:
```

Per ulteriori informazioni si faccia riferimento alla documentazione di lpd oppure ai seguenti *HOWTO*:

- *Printing-HOWTO*.
- *Printing-Usage-HOWTO*.

Eventuali stampe da e verso sistemi *Windows* possono essere gestite mediante il *software* Samba, descritto in una unità didattica a parte.

Linux - NIS (Network Information System)

È un sistema, inizialmente sviluppato da *Sun* col nome di *Yellow Pages* (`yp`), che permette la gestione centralizzata delle informazioni amministrative in ambiente *UNIX* (*password*, gruppi, *hosts*...), che in questo modo possono essere condivise da tutti i *client* appartenenti ad uno stesso dominio NIS. Pertanto un utente può collegarsi su macchine diverse utilizzando il medesimo *account* senza dover tenere sincronizzate fra i diversi *computer* le tabelle delle *password*. Le tabelle NIS prendono il nome di mappe.

Le tabelle vengono rese disponibili da un *server master* e possono essere replicate su uno o più *server* con funzioni di *backup* (*slave*). La modifica dei dati contenuti nelle zone è possibile da macchine remote opportunamente abilitate.

Linux - Configurazione dei client NIS

Se si vuole abilitare un *client* all'accesso ad un *server* NIS, è necessario attivare il demone `ypbind`, mediante uno *script* di avvio. Il dominio NIS viene impostato mediante il comando *`domainname`*, mentre la modalità di accesso al *server* viene definita nel *file* `/etc/yp.conf` nel seguente modo:

```
domain <dominio-NIS> server <host>
domain <dominio-NIS> broadcast
ypserv <host>
```

Nei *client* è necessario inoltre definire mediante il meccanismo di NSS (*Name Server Switch*), l'ordine con cui deve essere consultato il *server* NIS rispetto ai *file* di sistema. È buona norma consultare per primi i *file* di sistema, in modo da evitare che da un malfunzionamento di questo derivi l'impossibilità di collegarsi alla macchina. Nel caso della tabella delle *password* questo comportamento può essere ottenuto inserendo la seguente linea in `/etc/nsswitch.conf`:

```
passwd: files nis
```

Occorre inoltre aggiungere in coda a `/etc/passwd` la seguente linea:

```
+::::::
```

Di norma gli *account* amministrativi come *root*, *bin*, *wheel*, *demon* ed altri non vengono condivisi tramite NIS, per motivi di sicurezza e di *performance*, ma vengono invece utilizzati quelli definiti nel *file* `/etc/passwd` locale.

Per ulteriori informazioni su NSS e NIS si può fare riferimento anche alla unità didattica relativa all'autenticazione nei sistemi *UNIX*.

Configurazione del *server* NIS

Il *server* NIS per mantenere le informazioni non utilizza i classici *file* di testo, come ad esempio `/etc/passwd`, bensì dei *file* speciali detti mappe, contenenti coppie di valori nella forma chiave-valore, ordinati per chiave in modo da poter ricercare velocemente i dati.

Ad esempio partendo da `/etc/passwd` vengono generate le due mappe passwd.byname e passwd.byuid, che permettono rispettivamente ricerche veloci per nome utente e per numero identificativo (UID).

Le mappe vengono di norma salvate in `/var/nis` oppure in `/var/yp`, a seconda della configurazione che si utilizza. Per generare le mappe a partire dai *file* di sistema ogni implementazione mette a disposizione un comando apposito, ad esempio `makedbm`. Generalmente viene fornito un *Makefile* che permette la rigenerazione della mappe semplicemente spostandosi nella *directory* dove si trovano le tabelle e digitando *make*.

Per definire quali informazioni un *server* NIS può fornire alle macchine appartenenti ad un dominio si utilizza il *file* `/etc/ypserv.conf`, mentre gli *host* di fiducia possono essere impostati mediante `/etc/yp/securenets`.

Il demone che fornisce il servizio di *server* NIS è `ypserver`, che viene attivato in modalità *standalone* mediante uno *script* di avvio. Essendo NIS basato su RPC è necessario che sia attivo nel *server* il servizio *`portmapper`*.

Aggiornamento dei dati nelle mappe NIS

Per l'aggiornamento dei dati nel sistema NIS non è sufficiente utilizzare i classici comandi *UNIX*, ad esempio `passwd`. Vediamo ad esempio cosa succede se l'utente cambia la propria *password* su una macchina del dominio NIS. I casi possibili sono due:

- se la macchina non è il *server*, il *file* locale verrà aggiornato ma non verrà mai utilizzato, in quanto la macchina è configurata per prelevare le *password* via NIS.
- Se la macchina è il *server* NIS, questo dispone potenzialmente della nuova *password* ma le mappe contengono ancora la *password* precedente. Pertanto la nuova *password* funziona solo sul *server*, fintantoché non vengono rigenerate le mappe.

La soluzione consiste nell'utilizzo di comandi specifici per NIS, in questo caso `nispasswd`. Nel *server* è necessario attivare il servizio `rpc.passwdd`, che permette agli utenti di cambiare *password* da remoto. È anche possibile fare in modo che `rpc.passwdd` tenga automaticamente aggiornate le mappe a seguito di una modifica su uno dei *client*. In questo caso il servizio deve essere attivo quindi sia sul *server* che sui *client*.

Configurazione di un *server slave*

Per la configurazione di un *server* di tipo *slave* è sufficiente attivare normalmente le funzioni di NIS *server* (escluso `nispasswd`) e configurare il sistema come un *client* del *server master*. Per la replica delle mappe si utilizza il comando

```
/usr/lib/yp/ypinit -s <master-NIS>
```

generalmente mediante uno degli *script* `/usr/lib/yp/ypxfr*`.

NIS+

NIS+ è una estensione di NIS che permette una miglior gestione di domini con un numero elevato di *client*. Con questa nuova implementazione è stata anche migliorata la sicurezza dell'intero sistema con una riprogettazione dei meccanismi di scambio dati. Allo stato attuale esistono implementazioni di NIS+ disponibili come *Open Source* solo per quanto riguarda la parte *client*. Se si vuole quindi realizzare una rete NIS+ è necessario preventivare l'utilizzo di un *server Sun*.

LDAP

LDAP (*Lightweight Directory Access Protocol*) è un altro sistema di gestione centralizzata di informazioni, il quale offre un approccio più generico rispetto a NIS, in quanto non si limita ai soli dati amministrativi ma è utilizzabile anche per gestire dati generici quali *bookmarks* o indirizzari.

L'aggettivo *lighweight* (leggero), suggerisce una progettazione del sistema che privilegia le prestazioni. L'architettura di LDAP è infatti studiata per avere tempi di risposta molto veloci nella ricerca e lettura dei dati, a scapito di una maggiore lentezza nelle operazioni di scrittura e aggiornamento. Questo non rappresenta un problema grave, in quanto le operazioni di scrittura avvengono di norma meno frequentemente rispetto a quelle di lettura (si pensi al numero di volte in cui la propria agenda telefonica viene consultata rispetto a quelle in cui viene modificata).

LDAP gestisce insiemi di dati strutturati secondo una gerarchia ad albero (*directory*), nei quali i singoli elementi possono essere oggetti che possiedono diversi attributi. Gli elementi di una *directory*, nodi intermedi o foglie dell'albero, possono anche essere riferimenti ad altre *directory* residenti su altri *server* LDAP. Ogni elemento deve avere un nome che permetta di individuarlo in modo univoco nella *directory*.

Alcuni dei tipi possibili utilizzabili per definire gli attributi di un oggetto sono i seguenti:

- *bin*: dato binario;
- ces (*case exact string*): stringa in cui vengono distinte le lettere maiuscole dalle minuscole;
- cis (*case ignore string*): stringa in cui le lettere maiuscole e minuscole sono trattate allo stesso modo;
- tel: numero telefonico;
- dn (*distinguished name*): nome univoco.

Un *server* che fornisce in *UNIX/Linux* il servizio LDAP è `slapd`. Esso mantiene i dati in un formato denominato LDBM, simile al DBM ma ottimizzato per le ricerche (per ognuno degli attributi degli oggetti contenuti in una *directory* viene creato un indice).

Linux - Network File System

Il protocollo NFS (*Network File System*), originariamente sviluppato da *Sun Microsystems*, permette di condividere (esportare) *filesystem* verso altre macchine *UNIX*. Applicazioni possibili sono la condivisione delle *home directory* degli utenti fra un *cluster* di macchine oppure la realizzazione di sistemi *diskless*, in cui il *filesystem* di *root* viene montato da un *server* di rete.

In *Linux* per utilizzare la parte *client* è sufficiente avere un *kernel* compilato con il supporto per NFS. Il *mounting* avviene mediante un comando del tipo:

```
# mount -t nfs server01:/disk2 /test
```

dove `server01` è il nome del *server* e `/disk2` la *directory* da montare. Il parametro `-t` è opzionale. Come con gli altri tipi di *filesystem*, è possibile automatizzare il *mount* utilizzando la tabella `/etc/fstab`.

Per quanto riguarda la parte *server*, NFS utilizza come trasporto il protocollo RPC (*Remote Procedures Call*). È pertanto necessario verificare di avere attivo il *portmapper* e i demoni `rpc.mountd` (gestore del montaggio) e `rpc.nfsd` (gestore delle richieste dei *client*).

È possibile definire i *filesystem* da esportare mediante la tabella `/etc/exports`, secondo la seguente sintassi:

directory indirizzo(opzioni)

Gli indirizzi delle macchine a cui permettere il *mount* dei *filesystem* possono essere specificati anche mediante espressioni regolari. Ad esempio, per esportare in lettura e scrittura la *directory* `/disk2` verso una singola macchina si può utilizzare la seguente linea:

```
/disk2 lnxserver001(rw, no_root_squash)
```

L'opzione `no_root_squash` specifica che l'utente *root* del *client* deve essere mappato nell'utente *root* del *server* (di *default* verrebbe mappato nell'utente *nobody*).

Per esportare il CDROM a tutte le macchine della rete locale con indirizzi 192.168.10.x si può utilizzare la seguente linea:

```
/mnt/cdrom 192.168.10.0/255.255.255.0(ro, insecure)
```

Dopo aver modificato `/etc/exports` è necessario riavviare il *server* NFS.

Per ulteriori dettagli si faccia riferimento al documento NFS-*HOWTO*.

Linux - Condivisione di filesystem e stampanti con altri sistemi operativi

Coda è un *filesystem* di rete simile a NFS, che supporta operazioni a *filesystem* disconnesso, il *caching* persistente e altre funzionalità avanzate che lo rendono particolarmente adatto ad un utilizzo su reti lente o inaffidabili e su *computer* portatili. Esso è incluso di serie nelle versioni di *Linux* a partire dalla 2.2. (Ulteriori informazioni)

Linux permette di condividere *filesystem* e stampanti anche secondo protocolli propri di altre piattaforme, come *Apple Macintosh*, mediante il pacchetto *Netatalk* (*Netatalk*, *Netatalk*) o *Novell* NCP (si veda a proposito il documento IPX-*HOWTO*). Volendo condividere *filesystem* con macchine *Windows* si può utilizzare il pacchetto Samba, che verrà descritto in una unità didattica a parte.

Linux - AUTOFS (automount)

Nei sistemi *UNIX* ogni disco contenente un *filesystem* per essere accessibile deve essere montato in una *directory* mediante il comando *mount* (oppure al *boot* mediante `/etc/fstab`). `Autofs` permette di automatizzare tale l'operazione anche a sistema funzionante, in modo da non dover mantenere sempre montati i *filesystem* che non si utilizzano. In particolare può essere usato assieme a NFS per permettere ad un utente di utilizzare indifferentemente più macchine di una rete **NIS** senza dover tenere perennemente montate su ciascuna tutte le *home directory* degli utenti.

Per usufruire di tale funzionalità il *kernel* di *Linux* deve essere compilato con il supporto per `autofs` (in alternativa può essere caricato il modulo `autofs.o`). Inoltre sono necessari i programmi utente e il demone.

Tutte le richieste di accesso effettuate all'interno di una determinata *directory* (*master-directory*) vengono intercettate dal sistema e se la richiesta in questione riguarda una *subdirectory* presente nella configurazione del servizio *automount* ma attualmente non montata, questa viene montata automaticamente nel *filesystem*, in modo trasparente all'utente.

Anche l'operazione di rilascio viene gestita in automatico non appena la *directory* risulti inutilizzata per un certo periodo di tempo. Il *filesystem* da montare può essere un disco locale, un dispositivo rimovibile (cdrom, *floppy*) o un volume condiviso da un *server* di rete.

La configurazione di *automount* consta di un *file* di configurazione principale `/etc/auto.master` che contiene la lista delle *master-directory* nel formato

```
<master-dir> <file configurazione> <opzioni>
```

Per ogni *master-directory* viene definito il relativo *file* di configurazione ed eventuali opzioni, ad esempio `--timeout=xxx` con la quale si imposta il tempo di inutilizzo del *filesystem* prima che questo venga automaticamente smontato.

I *file* con configurazione di ogni *master-directory* ha il seguente formato:

```
<dir> <options> <filesystem>
```

in cui per ogni *subdirectory* si specificano le opzioni da passare al comando *mount*, ad esempio:

```
floppy -fstype=auto :/dev/floppy
cdrom -fstype=iso9660,ro :/dev/cdrom
```

Linux - Servizi Internet/Intranet

In questo capitolo viene descritto come configurare i principali servizi per Internet/Intranet. Gli esempi si riferiscono in modo specifico a *Linux Red Hat* 7.x.

I servizi descritti sono i seguenti:

- Telnet, **SSH**, RCP: servizi di **accesso al sistema**.
- **Bind**: *Domain Name Server*.
- **FTP**: trasferimento di *file*.
- **Apache**: *Web server*

Linux - Servizi di accesso al sistema

L'accesso al sistema in emulazione di terminale è possibile mediante il protocollo telnet. Il *client* telnet può essere utilizzato anche come strumento diagnostico, in quanto consente il collegamento diretto con qualunque porta TCP. Ad esempio, volendo testare il funzionamento del *server* POP3, è possibile eseguire il comando telnet *nameserver* 110.

Per fornire accesso al sistema mediante il protocollo telnet, in *Linux* si avvia il *server* `in.telnetd` con `inetd`, filtrando l'accesso mediante il TCP *wrapper*, conetenete una linea simile alla seguente in `/etc/inetd.conf`:

```
telnet stream tcp nowait root /usr/sbin/tcpd in.telnetd
```

Data la funzione del servizio telnet, è buona norma restringerne l'accesso solamente ai *client* fidati. Il *server* telnetd non necessita di particolari configurazioni, salvo la possibilità di personalizzare il messaggio che viene mostrato al *client* all'inizio del collegamento (`/etc/issue.net`).

In alternativa a telnet è possibile utilizzare i cosiddetti programmi r* (`rlogin`, `rsh`, `rcp`, ...) i quali oltre all'accesso in emulazione di terminale permettono l'esecuzione remota di comandi e la copia di *file* da e verso *server* remoti (`rcp`). È possibile utilizzare il *file* di configurazione `/etc/hosts.equiv` oppure un *file* `.rhosts` posto nella *directory* di un utente per permettere l'accesso a particolari utenti/indirizzi IP senza richiesta di *password*. Tale funzione è particolarmente utile all'interno di *shell script*, ma pericolosa dal punto di vista della sicurezza.

Linux - SSH

L'utilizzo di telnet e dei programmi `r-*` è sconsigliato, principalmente perché il traffico e le *password* vengono trasferite in chiaro. Al loro posto si dovrebbe utilizzare `ssh`, che offre funzioni analoghe e codifica il traffico mediante algoritmi crittografici. Esistono due versioni del *software*, `ssh` e `ssh2`, parzialmente incompatibili fra loro. In entrambi i casi si tratta di *software* non disponibile come *Open Source*.

Per ovviare a tale mancanza, è stata creata *Open*SSH, disponibile sotto licenza GPL. Questa viene fornita di serie con *Linux* e verrà utilizzata come riferimento in questo capitolo.

Le funzioni offerte da *Open*SSH sono molteplici e pertanto vedremo solamente le più semplici. Alcuni esempi di utilizzi sono i seguenti:

- connessione sicura ad un *server* in emulazione di terminale (`ssh nomeserver`);
- scambio sicuro di *file* (`scp utente@nomeserver:/disk2/documenti/file.doc file.doc`);
- *tunnelling* codificato per il protocollo X11;
- realizzazione di versioni codificate dei servizi;
- ridirezione di porte TCP su canali sicuri;

- realizzazione di tunnel VPN per collegamenti sicuri fra reti (PPP-over-SSH);
- utilizzo all'interno di *shell script*: è possibile l'autenticazione sia mediante il meccanismo `rhosts` tipico di rsh, sia utilizzando chiavi crittografiche;

*Open*SSH viene lanciato come un *server standalone*. La configurazione avviene mediante i *file* che si trovano nella *directory* `/etc/ssh`. Il controllo degli accessi può avvenire sia mediante le apposite direttive *AllowHosts* e *DenyHosts* in `/etc/ssh/sshd_conf`, sia utilizzando il TCP *wrapper*, in quanto sshd è compilato con il supporto della libreria `libwrap`.

Per il corretto funzionamento di SSH è necessario generare delle chiavi crittografiche per l'autenticazione. Tale operazione in genere viene svolta in modo automatico dagli *script* di installazione del pacchetto.

Per maggiori informazioni si può fare riferimento alla documentazione in linea oppure al sito http://www.openssh.org/

Linux - Name Server

Per la comprensione degli esempi che seguono è necessario disporre delle conoscenze di base relative al funzionamento del sistema DNS.

Il servizio di risoluzione dei nomi, ovvero la trasformazione di un indirizzo da nome simbolico ad indirizzo IP, viene fornito in molti sistemi *UNIX* attraverso il *server* DNS *bind*.

Il programma che gestisce le richieste (*named*) viene attivato come un demone. Il *file* di configurazione di *bind* è `/etc/named.conf` (nelle versioni più vecchie di *named*, al posto del *file* `/etc/named.conf`, si usava `/etc/named.boot`, con una diversa sintassi).

Per il controllo del *server* DNS si utilizza il programma `ndc`. In particolare i comandi `ndc start` e `ndc stop` permettono rispettivamente di attivare e disattivare il servizio.

Le zone, ovvero le tabelle contenenti le definizioni dei domini, si trovano nella *directory* definita nel *file* di configurazione:

```
options {
directory /var/named;
};
```

La configurazione del *server* DNS dipende dalla funzione che esso deve effettuare, in particolare dal fatto che esso debba essere utilizzato per fornire delle zone (DNS principale o secondario per un dominio Internet) oppure debba solamente occuparsi di risolvere i nomi per conto dei *client* di una rete locale (*cache* DNS).

Per ulteriori informazioni si può fare riferimento alla FAQ di *BIND* oppure consultare i seguenti documenti:

- *DNS and BIND, 4th edition, Cricket Liu and P. Albitz, O'Reilly & Associates*.
- *Linux DNS Server Administration, Craig Hunt, Sybex*.
- *Bind Homepage*.

Linux - Servizio di risoluzione di nomi per i client della rete locale

Il seguente esempio mostra come configurare *BIND* per fornire il servizio di nomi per i *client* di una rete locale:

```
options {
directory /var/named;
};

zone . {
type hint;
file named.root;
};
```

Poiché il meccanismo di risoluzione dei nomi opera in modo gerarchico e non sono definite altre zone, tutte le richieste di risoluzione dei nomi avvengono utilizzando la zona speciale ., mappata in un *file* che contiene la lista dei *root nameserver*.

Ad esempio se un *client* locale richiede la risoluzione del nome *www.sottodominio.dominio.com*, *named* verifica se è definita localmente la zona sottodominio.dominio.com. Non essendo questa definita, la ricerca continua con le zone dominio.com, .com e infine ., corrispondente al livello gerarchico più elevato nello spazio di nomi del DNS.

La zona . è contenuta nel *file* `named.boot`, che dovrebbe essere tenuto aggiornato con quello fornito da Internic. Essa ovviamente non contiene il *database* di tutti i nomi di dominio definiti, bensì gli indirizzi dei *server* a cui richiedere tali informazioni (*root nameserver*).

Uno di essi viene contattato per ottenere informazioni su quali *server* DNS hanno autorità per la zona .com e la procedura si ripete fino ad ottenere l'indirizzo dei *server* DNS che contiene le informazioni sulla zona sottodominio.dominio.com, a cui viene chiesto l'indirizzo (IN A) corrispondente a www.sottodominio.dominio.com.

Ovviamente *named* non ripete ogni volta tutte queste operazioni, ma tiene in memoria una *cache* delle zone a cui ha già acceduto, in modo da velocizzare le operazioni ed evitare richieste inutili (il tempo di validità di una zona nella *cache* dipende dai valori definiti all'interno della zona stessa).

Volendo limitare l'utilizzo del *nameserver* ai soli *client* della rete locale si possono definire delle apposite *access list*:

```
options {
allow-query { 192.168.10.0/24; localhost; };
};
```

Un altro metodo per fornire la risoluzione dei nomi ad una rete locale è quello di non utilizzare la zona . ma di passare invece tutte le richieste ad un altro DNS. Il *file* `named.conf` in questo caso diventa:

```
options {
directory /var/named;
forwarders { 111.112.113.114; };
};
```

Linux - Risoluzione degli indirizzi di rete LAN

Nel caso si desideri utilizzare *BIND* per la risoluzione degli indirizzi delle macchine nella rete locale, si deve definire in *named.conf* la zona relativa al dominio utilizzato:

```
zone intranet.miodominio.com {
type master;
file intranet.miodominio.com.zone;
};
```

Inoltre è buona norma definire anche la zona per la risoluzione inversa da indirizzo IP a nome simbolico:

```
zone 1.168.192.in-addr.arpa {
type master;
file 192.168.1.rev;
};
```

Le relative tabelle devono essere create all'interno della *directory* definita in `named.con`, ad esempio `/var/named/intranet.miodominio.com.zone`:

```
@ IN SOA server.miodominio.com. root.miodominio.com. (
2002031800 ; Serial
28800 ; Refresh
7200 ; Retry
604800 ; Expire
86400 ) ; Minimum
NS server.miodominio.com.
pc001 A 192.168.1.1
pc002 A 192.168.1.2
```

La tabella di riversa `/var/named/192.168.1.rev` diventa:

```
@ IN SOA server.miodominio.com. root.miodominio.com. (
2002031800 ; Serial
28800 ; Refresh
7200 ; Retry
604800 ; Expire
86400 ) ; Minimum
NS server.miodominio.com.
1 PTR pc001.intranet.miodominio.com.
2 PTR intranet.miodominio.com.
```

Le due tabelle devono essere tenute sincronizzate fra di loro ed i valori del campo *Serial* incrementati ad ogni modifica dei *file* (non è importante che l'incremento sia unitario: nell'esempio è stata usata la convenzione diffusa di utilizzare la data di modifica seguita da due cifre usate per distinguere eventuali modifiche effettuate lo stesso giorno).

Per ulteriori dettagli sul formato delle zone si faccia riferimento ad un documento di informazioni specifico sull'argomento.

Linux - Utilizzo come nameserver per un dominio Internet

Volendo utilizzare *BIND* in un *server* accessibile su Internet per la risoluzione dei nomi di un dominio, la definizione della zona è del tutto analoga a quanto visto nel caso del *nameserver* locale. Si deve solamente aver cura di definire i record relativi ai *server* a cui deve essere consegnata la posta elettronica del dominio (MX - *Mail Exchangers*).

Generalmente per un dominio sono richiesti più *server* DNS, possibilmente posti su reti diverse, in modo da assicurare un certo grado di ridondanza. I *nameserver* aggiuntivi vengono anche chiamati DNS secondari. Essi prelevano copia delle zone da un *nameserver* primario, che viene definito mediante una linea simile alla seguente in `named.conf`:

```
zone test.com {
type slave;
masters { 213.222.212.195; };
file test.com.zon;
```

```
};
```

Nel *server* primario è consigliabile restringere la possibilità di trasferimento di una zona solo ai *nameserver* secondari, in modo da evitare che la lista completa delle macchine presenti nella propria rete possa cadere in mano di estranei:

```
zone test.it {
type master;
file test.it.hosts;
allow-transfer { 193.6.122.33; };
};
```

Linux - Sicurezza del server DNS

Esistono diversi tipi di attacchi che possono essere portati a *named*. Pertanto è bene curare con attenzione il fattore sicurezza. Nelle versioni di *BIND* recenti esistono parecchie direttive che possono essere inserite nel *file* di configurazione allo scopo di rendere più difficili eventuali attacchi.

In particolare, *BIND* dovrebbe essere sempre fatto funzionare con i permessi di un utente non privilegiato. Per far questo è necessario modificare lo *script* di avvio `/etc/rc.d/init.d/named` aggiungendo nella linea di comando di *named* le opzioni:

- -u utente;
- -g gruppo;

Eventualmente è anche possibile *restringere la visibilità ad una porzione limitata del filesystem* mediante il meccanismo di `chroot` utilizzando l'opzione -t *directory*. In questo caso è necessario spostare i *file* di configurazione in una posizione tale che essi siano accessibili a *named* con il percorso corretto anche dopo il `chroot`.

Ad esempio, volendo restringere la visibilità al solo sottoalbero `/home/named` occorrerà creare il *file* di configurazione in `/home/named/etc/named.conf` e le zone in `/home/named/var/named`. Nel caso si utilizzi *BIND* come DNS secondario per un dominio, sarà necessario rendere disponibili con i percorsi corretti anche i programmi che si occupano della replica delle zone dal DNS principale (*named-xfer*) e le eventuali librerie dinamiche da essi utilizzate. I dettagli sono spiegati nel documento Chroot-*BIND*8-*HOWTO* oppure sul documento chrootdns.

Per altri dettagli sulle funzioni relative alla sicurezza si faccia riferimento alla documentazione fornita col *software* oppure al DNS-*HOWTO*.

Linux - FTP

Il *server* per il protocollo FTP (*File Transfer Protocol*) fornito di serie con *Linux* è `wu-ftp`, della *Washington University*. Esso viene fornito preconfigurato per essere lanciato mediante `inetd`, ma può essere utilizzato anche in modalità *standalone* richiamandolo da uno *script* di avvio mediante l'opzione -s.

I *file* di configurazione di `wu-ftpd` sono i seguenti:

- `/etc/ftpusers`: contiene la lista degli *account* con cui non possibile effettuare una connessione al *server* mediante FTP. Per motivi di sicurezza di solito l'accesso viene vietato agli utenti di sistema (*root*, *bin*, *mail*, ...).
- `/etc/ftpaccess`: se `ftpd` viene eseguito con l'opzione -a, questo *file* viene utilizzato per la gestione degli accessi. È possibile definire diverse classi di utenti a cui associare determinati

permessi sulle *directory* (`chmod`, *`delete`*, *`overwrite`*, *`rename`*, ...). Per ogni classe di utenti è inoltre possibile scegliere il numero massimo di connessioni accettabili contemporaneamente.
- `/etc/ftpconversions`: viene usato per determinare le modalità di conversione dei *file* compressi. Ad esempio è possibile fare in modo che richiedendo il trasferimento di un *file* esistente aggiungendo al nome l'estensione `.gz`, esso venga compresso al volo con `gzip`. Analogamente si può fare in modo che, dato il nome di una *directory* con aggiunta l'estensione `.tar.gz`, `ftpd` crei automaticamente il corrispondente archivio compresso.
- `/etc/ftphosts`: permette di filtrare l'accesso al servizio FTP da determinati utenti o indirizzi IP. Se il *server* viene attivato mediante `inetd`, è possibile creare delle *access list* anche utilizzando il TCP *wrapper*.

Lo stato del servizio FTP può essere tenuto sotto controllo mediante i comandi `ftpcount` (visualizza il numero di utenti connessi ed il numero massimo di connessioni ammissibili per ogni classe di utenza) e `ftpwho` (mostra la lista e le operazioni che stanno compiendo gli utenti attualmente collegati al sistema). Il log completo delle operazioni viene tenuto nel *file* `/var/log/xferlog`.

Linux - FTP anonimo e modalità chroot

Se si vuole permettere l'accesso al *server* FTP in modo anonimo, è necessario creare in `/etc/passwd` l'utente speciale ftp. La *password* ad esso associata non è importante, in quanto non viene controllata (è comunque necessario assegnarla o sostituirla col valore * nel *file* delle *password* onde evitare accessi indesiderati attraverso altri servizi).

Per utilizzare l'FTP anonimo si accede al sistema identificandosi come ftp, oppure come *anonymous*. Al posto della *password* di norma si utilizza il proprio indirizzo di posta elettronica.

L'accesso anonimo si differenzia da quello autenticato per il fatto che viene eseguito un `chroot` nella *home directory* dell'utente ftp (ad esempio `/home/ftp`). In tale *directory* pertanto deve essere ricreata una struttura di *filesystem* che contenga i seguenti *file*, necessari al demone `ftpd` per listare il contenuto delle *directory*:

```
/home/ftp/etc/passwd
/home/ftp/etc/group
/home/ftp/etc/ld.so.cache
/home/ftp/bin/ls
/home/ftp/bin/gzip
/home/ftp/lib/libc.so.6
/home/ftp/lib/ld-linux.so.2
```

Per motivi di sicurezza sarebbe preferibile che tali *file* fossero di proprietà dell'utente *root* e non scrivibili ad altri utenti.

Il contenuto dei *file* `passwd` e *`group`* può essere fittizio, in quanto vengono utilizzati solamente dal comando `ls` per visualizzare lo *username* e il nome del gruppo proprietario dei *file*. In caso contrario nell'*output* di `ls` verrebbero indicati i valori numerici di UID e GID.

Mediante la direttiva `guestuser nome` di `/etc/ftpaccess` è possibile definire degli utenti per i quali l'accesso deve essere eseguito in modalità `chroot`. Così facendo, l'utente non può risalire sopra la propria *home directory* e vedere il contenuto dell'intero *filesystem*. Per ogni utente per cui si attiva tale opzione deve essere ricreata la stessa struttura di *directory* vista per l'utente ftp anonimo. Inoltre il campo contenente la *home directory* nel *file* `/etc/passwd` deve essere modificato come nell'esempio che segue:

```
mrossi:x:306:100:Mario Rossi:/home/mrossi/./:/bin/bash
```

Volendo creare molti *account* che utilizzino il meccanismo di `chroot`, conviene sostituire il *server* `wu-ftpd` con il programma `Pureftpd`, il quale non necessita di dover ricreare la struttura di *directory* per ogni utente.

Linux - Apache

Il *server* HTTP più utilizzato in ambiente *UNIX* è *Apache*. In *Linux* esso viene avviato come un *server* a se stante mediante lo *script* `/etc/rc.d/rcX.d/S85httpd`, tuttavia è possibile anche utilizzarlo attraverso `inetd`, avendo cura di inserire la direttiva `ServerType inetd` nel *file* di configurazione `httpd.conf`. Per utilizzi meno impegnativi esistono dei *server* HTTP più leggeri e semplici da configurare, ad esempio BOA.

Apache ha una struttura modulare, in cui è possibile aggiungere nuove funzionalità mediante la ricompilazione del codice oppure, nelle versioni recenti, semplicemente aggiungendo un nuovo modulo, fornito sotto forma di libreria dinamica, nella *directory modules* ed adattando opportunamente il *file* di configurazione.

In *Red Hat Linux Apache* viene fornito precompilato sotto forma di pacchetti RPM contenenti il *server* base (http) e i moduli per le funzionalità aggiuntive più utilizzate (`mod_ssl`, `mod_perl`, ...).

Nel sito del progetto *Apache*, è presente una lista di tutti i moduli disponibili per *Apache* (*Module Registry*), comprendente sia quelli inclusi nella distribuzione del *server* che quelli realizzati da terzi. Esistono moduli adatti per diverse funzioni (*parsing* di linguaggi, metodi di autenticazione, *logging* ...).

Per maggiori informazioni si invita a leggere la documentazione fornita in linea col prodotto oppure a visitare il sito, che contiene una lista di siti e libri adatti ad approfondire l'argomento. Ulteriori informazioni sono presenti anche nei seguenti *HOWTO* di *Linux*: *Apache*-Introduzione-*HOWTO*, *Apache-Compile-HOWTO*, *ISP-Setup-RedHat-HOWTO*, *SSL-RedHat-HOWTO*, WWW-*HOWTO*.

Linux - Configurazione di Apache

La configurazione di *Apache* avviene mediante i *file* contenuti nella *directory*

`/etc/httpd/conf`

(è possibile specificare un altro percorso utilizzando l'opzione -d nella linea di comando di `httpd`), il più importante dei quali è `httpd.conf`, che permette di definire tutti gli aspetti della configurazione del *server*. In alcune installazioni la configurazione è divisa fra i *file* `httpd.conf`, `srm.conf` e `access.conf`. Una volta modificato un *file* di configurazione è necessario riavviare il servizio con `/etc/rc.d/init.d/httpd restart`.

Volendo sono disponibili delle interfacce grafiche o via *Web* che consentono una configurazione semplificata del *server* (si faccia riferimento al sito per una lista del *software* disponibile).

Particolare importanza ha la definizione delle due *directory*:

- *ServerRoot* (esempio: `/etc/httpd/conf`), contenente i *file* di configurazione;
- *DocumentRoot* (esempio: `/var/www/html`), a partire dalla quale si inseriscono i documenti che si desidera rendere disponibili attraverso il protocollo HTTP.

Mediante la direttiva *UserDir* è anche possibile permettere agli utenti del sistema di creare una propria *directory* personale, ad esempio `/home/nomeutente/public_html`, che sarà visibile come

`http://nomeserver/~nomeutente`.

Mediante la direttiva *Directory* di `httpd.conf` si possono definire i permessi di accesso e le *Options* di *default* per la diverse *directory*, che vengono applicati in modo ricorsivo anche alle sotto*directory*.

Se si specifica l'opzione *AllowOverride* è possibile permettere la modifica di permessi e *Options* da parte degli utenti mediante un *file* di nome `.htaccess` posto nella *directory* desiderata, che vale in modo ricorsivo anche per tutte le sotto*directory*.

```
<Directory /var/www/html>
Options Indexes Includes ExecCGI
AllowOverride None
order allow,deny
allow from all
</Directory>
```

Linux - Restrizione di accesso ad una directory

I metodi base di autenticazione forniti da *Apache* filtrano gli accessi in base all'indirizzo del *client* oppure richiedendo uno *username* ed una *password* che vengono confrontati con quelli contenuti in un *file* creato mediante il programma `htpasswd`. Esistono moduli aggiuntivi (`mod_auth_*`) che consentono altri tipi di autenticazione, ad esempio utilizzando il metodo standard PAM, un *server* di autenticazione (*Radius*, *Windows* NT, ...) oppure una tabella di *database* (`mod_auth_mysql`, ...).

Il seguente esempio di *file* `.htaccess` mostra come limitare l'accesso *GET* e *POST* ad una *directory* ai soli utenti definiti nel *file* `passwd_sito`, che per ragioni di sicurezza viene conservato in una posizione del *filesystem* non accessibile mediante HTTP.

```
AuthUserFile /home/pippo/SECURE/passwd_sito
AuthGroupFile /home/pippo/SECURE/group_sito
AuthName Password_Approvazione
AuthType Basic
<Limit GET POST>
require valid-user
</Limit>
```

Un esempio di restrizione in base all'indirizzo del *client* è invece il seguente:

```
<Limit GET POST>
order deny,allow
deny from all
allow from .test.it
allow from 192.168.22.
</Limit>
```

Mediante la direttiva *Options* è possibile abilitare o restringere l'utilizzo di alcune funzioni problematiche per la sicurezza, ad esempio inibire l'utilizzo di *script* CGI in particolari *directory* oppure fare in modo che il *server* non segua i link simbolici.

```
Options ExecCGI FollowSymLinks
```

Linux - Sicurezza

Oltre alle restrizioni appena viste, esistono altre funzioni di *Apache* che permettono di gestire in modo sicuro il servizio HTTP, come la possibilità di far funzionare il *server* con i permessi di un utente non privilegiato, mediante le direttive *User* e *Group*, oppure di fare girare i programmi CGI

con i permessi dell'utente proprietario. Nelle versioni recenti di *Apache* non è possibile avviare il *server* con i permessi di *root*, a meno di ricompilare il programma attivando una particolare opzione.

Linux - Supporto per SSL

Esistono due metodi per fornire ad *Apache* il supporto per la codifica crittografica dei dati mediante SSL:

- *Apache*-SSL: si tratta di una versione del *server* modificata in modo da contenere il supporto per SSL;
- `mod_ssl`: è un modulo aggiuntivo che può essere utilizzato nella versione standard di *Apache*.

In entrambi i casi il funzionamento si basa sulla libreria *Open*SSL.

Linux - Log degli accessi e server virtuali

Il *logging* degli accessi avviene nel *file*

- `/var/log/httpd/access_log`,

mentre eventuali errori vengono mandati in

- `/var/log/httpd/error_log`.

È possibile variare il percorso di tali *file* utilizzando le direttive `TransferLog` e `ErrorLog`. Il formato standard per questi *file* è il *Common Logfile Format*, compatibile con molti programmi per generare statistiche di accesso. È comunque possibile utilizzare anche altri formati standard (esempio: *Combined*) oppure definire un proprio formato.

È possibile utilizzare *Apache* per creare *server* virtuali, ovvero per gestire i siti *Web* di più domini diversi, sia utilizzando per ogni sito un indirizzo IP diverso, sia condividendo uno stesso indirizzo IP fra più siti.

Mediante la direttiva di configurazione *VirtualHost* è possibile ridefire per ogni *server* virtuale molte opzioni di *Apache*, fra cui quelle relative ai *file* di log, in modo da avere un report separato per ogni dominio.

Linux - Pagine dinamiche e scripting

Oltre che per fornire pagine statiche, è possibile utilizzare *Apache* anche per creare siti dinamici, utilizzando dei programmi CGI oppure uno dei molti linguaggi di *scripting* disponibili (PHP, Perl, TCL, SSI, ...). Una piattaforma di sviluppo molto diffusa in ambiente *Linux* è composta del *server Apache* utilizzato assieme al linguaggio PHP ed al *database MySQL*.

Maggiori dettagli sul linguaggio PHP possono essere reperiti, assieme ad una ampia bibliografia, sul sito, nel manuale fornito col linguaggio e nel PHP-*HOWTO*.

Sicurezza in Linux - Proxy

Per *Linux* esistono diversi *software* adatti ad essere utilizzati come *proxy* per permettere l'accesso all'esterno da parte degli utenti di una rete interna con indirizzi privati. Utilizzando assieme diversi prodotti è possibile utilizzare i protocolli più diffusi (HTTP, FTP, telnet, POP3, H.323, ...).

Una soluzione di *firewall/proxy* molto completa e compatibile con quasi tutti i protocolli è *SOCKS*. Tuttavia esso non è molto utilizzato in quanto ha lo svantaggio di richiedere delle modifiche ai programmi preesistenti.

TIS Firewall Toolkit (FWTK), è una raccolta di *firewall* basati su *proxy*, compatibile con i protocolli FTP, HTTP, NNTP, RLOGIN, TELNET e SSL.

Per ogni protocollo che si intende utilizzare è necessario lanciare mediante `inetd` o `xinetd` un apposito demone, che riceve le richieste di connessione su una porta TCP (ad esempio 2323 per telnet-gw), le filtra in base all'indirizzo IP sorgente, il nome dell'utente o altri parametri, e le passa al servizio reale.

Nonostante FWTK sia utilizzabile gratuitamente, non si tratta di *software Open Source* e per ottenerlo è necessaria una procedura di registrazione sul sito del produttore.

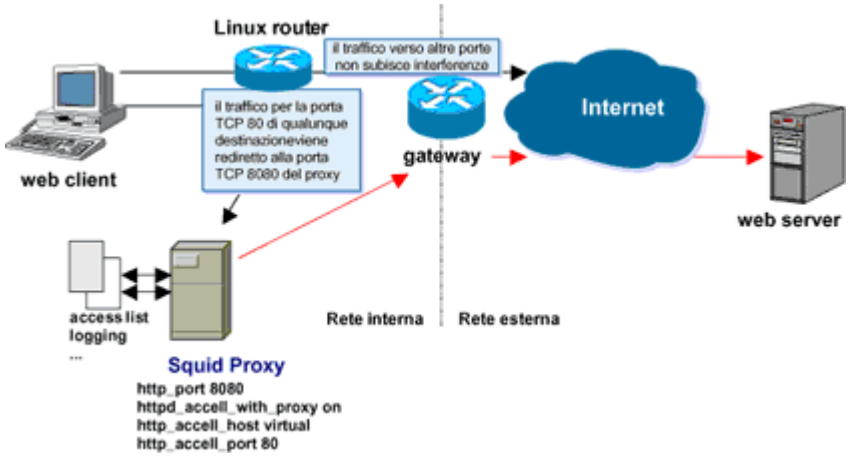
Sicurezza in Linux - Squid

Fra i molti *proxy* per il protocollo HTTP, il più diffuso è Squid. Si tratta di un programma molto flessibile e completo, dotato di una vasta serie di programmi aggiuntivi. Squid offre fra le altre cose la possibilità di creare delle gerarchie di *proxy*.

Squid offre anche funzionalità di *cache* delle pagine a cui si è acceduto. Questo permette di ridurre il traffico di rete, in quanto le pagine già in memoria non vengono ricaricate, ma bensì ne viene solamente verificato lo stato di aggiornamento. Nel caso di *file* di dimensioni significative, come le immagini, questo comporta notevoli vantaggi sia in termine di velocità di accesso che di *bytes* trasferiti.

La configurazione di Squid avviene di solito mediante la modifica diretta del *file* di testo `/etc/squid/squid.conf`, ma viene anche fornito un semplice *script* CGI che permette la configurazione di alcuni aspetti attraverso una interfaccia *Web*. Altri programmi di configurazione più complessi possono essere reperiti nel sito SQUID (ad esempio esistono dei moduli per `webmin` e `linuxconf`).

La gestione delle *access list* di Squid può essere estesa utilizzando un *authenticator* esterno, che consente di implementare nuovi metodi di autenticazione, ad esempio basati su *username*/*password* o sull'orario di accesso. È inoltre possibile utilizzare un *redirector* esterno per riscrivere gli indirizzi URL inseriti dagli utenti. Questo consente la creazione di *black list*.



È possibile utilizzare Squid assieme alle funzionalità di filtraggio del traffico presenti in *Linux* per costruire un *transparent proxy*. Esso consiste nel redirigere in modo automatico verso il *proxy* tutto il

traffico generato dalle macchine della rete interna e destinato alla porta TCP 80 di un indirizzo esterno. Per far ciò è ovviamente necessario predisporre la topologia della rete in modo che tutto il traffico della rete passi per una singola macchina su cui sia possibile intercettarlo e ridirigerlo.

Essendo la funzione di transparent *proxy* svolta a livello di pacchetti IP, non sarà necessario configurare il supporto per il *proxy* sui singoli *browser*, che, anzi, non si accorgeranno neppure di passare attraverso un *proxy*. Tale soluzione permette di effettuare il *caching* automatico del traffico *Web* della propria rete, oppure di applicare funzioni di *logging*, *access list* o *black list*.