Administració de servidors

Remo Suppi Boldrito

PID_00174421

Universitat Oberta de Catalunya
www.uoc.edu

Índex

Introducció

La interconnexió entre màquines i les comunicacions d'alta velocitat han permès que els recursos que s'utilitzen no siguin al mateix lloc geogràfic de l'usuari. UNIX (i per descomptat GNU/Linux) és probablement el màxim exponent d'aquesta filosofia, ja que des del seu inici ha fomentat l'intercanvi de recursos i la independència de dispositius. Aquesta filosofia s'ha plasmat en una cosa comuna avui en dia com els serveis. Un servei és un recurs (que pot ser universal o no) i que permet, en certes condicions, obtenir informació, compartir dades o simplement processar la informació a distància. El nostre objectiu és analitzar els serveis que permeten el funcionament d'una xarxa. Generalment, en aquesta xarxa hi haurà una màquina (o diverses, segons les configuracions) que farà possible l'intercanvi d'informació entre les altres. Aquestes màquines s'anomenen *servidors* i contenen un conjunt de programes que permeten que la informació estigui centralitzada i sigui fàcilment accessible. Aquests serveis permeten la reducció de costos i amplien la disponibilitat de la informació, però s'ha de tenir en compte que un servei centralitzat presenta inconvenients, ja que pot quedar fora de servei i deixar sense atenció tots els usuaris. En aquest mòdul es veuran els principals serveis que permeten que una màquina GNU/Linux tingui un paper molt important en una infraestructura tecnològica, tant en centralitzar i distribuir dades com en ser punt d'informació, accés o comunicació.

Serveis replicats

Una arquitectura de servidors ha de tenir els serveis replicats (*mirrors*) per a resoldre els inconvenients que comporta.

Objectius

En els materials didàctics d'aquest mòdul trobareu els continguts i les eines procedimentals per a aconseguir els objectius següents:

1. Presentar els aspectes més rellevants dels conceptes involucrats, tant teòrics com pràctics, en l'estructura de servidors/serveis en un sistema GNU/Linux.
2. Analitzar els conceptes relatius a serveis i servidors específics d'un sistema GNU/Linux.
3. Experimentar amb la configuració i adaptar la instal·lació de serveis a un entorn determinat.
4. Analitzar i participar en discussions sobre les possibilitats actuals i futures de nous serveis i els obstacles que hi ha, bàsicament en aspectes de seguretat, en els diferents entorns de treball de GNU/Linux.

1. Administració de servidors

Els serveis es poden classificar en dos tipus: de vinculació ordinador-ordinador o de relació home-ordinador. En el primer cas, es tracta de serveis requerits per altres ordinadors, mentre que en el segon són serveis requerits pels usuaris (encara que hi ha serveis que poden actuar en ambdues categories). En el primer tipus hi ha serveis de noms, com el *domain name system* (DNS), el servei d'informació d'usuaris (NIS-YP), el directori d'informació LDAP o els serveis d'emmagatzematge intermedi (*proxies*). En la segona categoria hi ha serveis de connexió interactiva i execució remota (ssh, telnet), transferència de fitxers (ftp), intercanvi d'informació a escala d'usuari, com el correu electrònic (MTA, IMAP, POP), missatges (*news*), *World Wide Web*, *Wiki* i arxius (NFS). Per mostrar les possibilitats de GNU/Linux Debian-FC, descriurem cadascun d'aquests serveis amb una configuració mínima i operativa, però sense descuidar els aspectes de seguretat i estabilitat.

1.1. Sistema de noms de domini (*domain name system, DNS*)

La funció del servei de DNS és convertir noms de màquines (llegibles i fàcils de recordar pels usuaris) en adreces IP i viceversa.

A la consulta de quina és la IP de *pirulo.remix.com*, el servidor respondrà 192.168.0.1 (aquesta acció es coneix com a *mapping*); de la mateixa manera, quan se li proporcioni l'adreça IP, respondrà amb el nom de la màquina (això es coneix com a *reverse mapping*).

El sistema de noms de domini (DNS) és una arquitectura arborescent que evita la duplicació de la informació i facilita la cerca. Per això, un únic DNS només té sentit com a part de l'arbre. L'aplicació que presta aquest servei s'anomena *named* i s'inclou en la majoria de distribucions de GNU/Linux (`/usr/sbin/named`) i forma part d'un paquet anomenat `bind` (actualment versió 9.x), coordinat per l'Internet Software Consortium (ISC). El DNS és simplement una base de dades, per la qual cosa cal que les persones que la modifiquin en coneguin l'estructura, ja que, en cas contrari, el servei quedarà afectat. Com a precaució, s'ha de tenir una cura especial a guardar les còpies dels arxius per a evitar qualsevol interrupció en el servei. El paquet sobre Debian està com a `bind` i `bind.doc`. [5, 7, 11]. Les configuracions són similars a

les de FC, però caldrà instal·lar `bind, bind-utils` i `caching-nameserver`, que seran gestionades pel `yum`, per exemple.

1.1.1. Servidor de noms cau

En primer lloc, es configurarà un servidor de DNS per a resoldre consultes i que actuï com a memòria cau per a les consultes de noms (*resolver, caching only server*). És a dir, la primera vegada consultarà el servidor adequat perquè es parteix d'una base de dades sense informació, però la següent vegada respondrà el servidor de noms cau, amb la corresponent disminució del temps de resposta. Per a configurar el servidor de noms cau, es necessita l'arxiu `/etc/bind/named.conf` (a Debian), que té el format següent (s'han respectat els comentaris originals a l'arxiu, indicats per //):

```
options {
    directory "/var/cache/bind";
    // query-source address * port 53;
    // forwarders {
    // 0.0.0.0;
    //};
    auth-nxdomain no; # conform to RFC1035
};
// prime the server with knowledge of the root servers
zone "." {
     type hint;
     file "/etc/bind/db.root"; };
     // be authoritative for the localhost forward and reverse zones, and for
     // broadcast zones as per RFC 1912
zone "localhost" {
     type master;
     file "/etc/bind/db.local";
     };
zone "127.in-addr.arpa" {
     type master;
     file "/etc/bind/db.127";
     };
zone "0.in-addr.arpa" {
     type master;
     file "/etc/bind/db.0";
     };
zone "255.in-addr.arpa" {
     type master;
     file "/etc/bind/db.255";
     };
// add entries for other zones below here
```

La sentència `directory` indica on seran els arxius de configuració restants (`/var/cache/bind` en el nostre cas).

L'arxiu `/etc/bind/db.root` contindrà una cosa similar a (es mostren algunes línies, on els comentaris són les línies que comencen per ";"; s'ha de anar amb compte, a més, amb els punts (.) al començament d'algunes línies, ja que formen part del format de l'arxiu [aquest arxiu es pot obtenir actualitzat, directament d'Internet]):

```
...
; formerly NS.INTERNIC.NET
;
. 3600000 IN NS A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET. 3600000 A 198.41.0.4
;
; formerly NS1.ISI.EDU
;
. 3600000 NS B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET. 3600000 A 128.9.0.107
;
; formerly C.PSI.NET
;
. 3600000 NS C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET. 3600000 A 192.33.4.12
;
...
```

Aquest fitxer descriu els *root name servers* al món. Aquests servidors canvien, de manera que l'arxiu s'ha d'actualitzar periòdicament. Les següents seccions són les `zones`; les `zones localhost` i `127.in-addr.arpa`, que es vinculen als fitxers a l'arxiu `etc/bind/db.local` i `/etc/bind/db.127`, es refereixen a la resolució directa i inversa per a la interfície local. Les `zones` següents són per a les zones de difusió (segons RFC, 1912) i al final s'haurien afegit les pròpies. Per exemple, l'arxiu `db.local` podria ser (";" significa *comentari*):

```
; BIND reverse data file for local loopback interface
$TTL 604800
@ IN SOA ns.remix.bogus. root.remix.bogus. (
     1         ; Serial
     604800         ; Refresh
     86400         ; Retry
     2419200         ; Expire
     604800)         ; Negative Cache TTL
@ IN NS         ns.remix.bogus.
1.0.0 IN PTR         localhost.
```

Explicarem com s'utilitza més endavant. Ara s'ha de posar com a *name server* en el `/etc/resolv.conf`:

```
search subdominio.su-dominio.dominio
su-dominio.dominio
# por ejemplo, search remix.bogus bogus
nameserver 127.0.0.1
```

on caldrà reemplaçar els `subdominio.su-dominio.dominio` pels valors adequats. La línia `search` indica quins dominis se cercaran per a qualsevol amfitrió (*host*) que es vulgui connectar* i `nameserver` especifica l'adreça del seu *nameserver* (en aquest cas la seva màquina, que és on s'executarà el `named`). El `search` té aquest comportament: si un client cerca la màquina `pirulo`, primer cercarà `pirulo.subdominio.su-dominio.dominio`, a continuació `pirulo.su-dominio.dominio` i, finalment, `pirulo`. Això implica temps de

***És possible substituir `search` per `domain`, encara que tenen comportaments diferents.**

cerca; ara bé, no és necessari posar la resta si es té la certesa que `pirulo` està a `subdominio.su-dominio.dominio`.

El pas següent és posar en marxa el `named` i mirar els resultats de l'execució. Per posar en marxa el dimoni (*daemon*), podeu fer directament amb l'*script* d'inicialització `/etc/init.d/bind9 start` (verifiqueu el nom de l*script* al directori `/etc/init.d`. En cas que el `named` ja s'estigui executant, llavors feu `/etc/init.d/bind9 reload`) o, si no, feu també `/usr/sbin/named`. Mirant el registre del sistema a l'arxiu `/var/log/daemon.log` veurem una cosa així:

```
Sep 1 20:42:28 remolix named[165]: starting BIND 9.2.1
Sep 1 20:42:28 remolix named[165]: using 1 CPU
Sep 1 20:42:28 remolix named[167]: loading configuration from "/etc/bind/named.conf"
```

Aquí s'indica l'arrencada del servidor i els missatges d'errors (si n'hi ha), els quals s'hauran de corregir i s'haurà de tornar a començar. Ara es pot verificar la configuració amb ordres com `nslookup` (original i fàcil, però obsolet segons alguns autors), `host` o `dig` (recomanat). La sortida de `dig -x 127.0.0.1` serà alguna cosa com ara:

```
# dig -x 127.0.0.1
;; <<>> DiG 9.2.1 <<>> -x 127.0.0.1
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 31245
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0
;; QUESTION SECTION: ;1.0.0.127.in-addr.arpa. IN PTR
;; ANSWER SECTION: 1.0.0.127.in-addr.arpa. 604800 IN PTR localhost.
;; AUTHORITY SECTION: 127.in-addr.arpa. 604800 IN NS ns.remix.bogus.
;; Query time: 1 msec
;; SERVER: 127.0.0.1 \#53(127.0.0.1)
;; WHEN: Mon Sep 1 22:23:35 2010
;; MSG SIZE rcvd: 91
```

on es pot veure que la consulta ha trigat 1 milisegon. Si es disposa de connexió a Internet, es podria buscar alguna màquina dins del vostre domini i veure el comportament del vostre servidor. En *BIND9* hi ha el `lwresd` (*lightweight resolver daemon*), que és el dimoni que proveeix de serveis de noms clients que utilitzen la biblioteca de *BIND9 lightweight resolver*. És essencialment un servidor cau (com el que s'ha configurat) el que fa les consultes utilitzant el *BIND9 lightweight resolver protocol* en lloc del protocol DNS. Aquest servidor escolta per la interfície 127.0.0.1 (per la qual cosa només atén processos de la màquina local) en UDP i el port 921. Les consultes dels clients es descodifiquen i es resolen amb el protocol DNS. Quan s'obtenen les respostes, el `lwresd` les codifica en format *lightweight* i les retorna al client que les ha demanades.

Finalment, com ja s'ha esmentat, el nucli utiliza diverses fonts d'informació per a la xarxa, que s'obtenen des d'`/etc/nsswitch.conf`. Aquest arxiu indica des d'on s'obté la font d'informació i per als noms de màquines i IP hi ha una secció com:

```
hosts: files dns
```

Aquesta línia (si no hi és s'ha d'afegir) indica que qui necessiti un nom d'una màquina o una IP primer ho ha de consultar a `/etc/hosts` i després a DNS, d'acord amb els dominis indicats a `/etc/resolv.conf`.

1.1.2. *Forwarders*

En xarxes amb una càrrega considerable és possible equilibrar el trànsit amb la secció de *forwarders*. Si el vostre proveïdor de xarxa (ISP) té un o més *nameservers* estables, és recomanable utilitzar-los per a descongestionar les consultes sobre el seu servidor. Per a això, s'ha de treure el comentari (`//`) de cada línia de la secció `forwarders` de l'arxiu `/etc/bind/named.conf` i reemplaçar el 0.0.0.0 amb les IP dels *nameservers* del vostre ISP. Aquesta configuració és recomanable quan la connexió és lenta.

1.1.3. Configuració d'un domini propi

DNS té una estructura en arbre i l'origen es coneix com a "." (vegeu el fitxer `/etc/bind/db.root`). Sota el "." hi ha els TLD (dominis de primer nivell o *top level domains*) com **org**, **com**, **edu**, **net**, etc. Quan es busca en un servidor, si aquest no coneix la resposta, es buscarà recursivament en l'arbre fins a trobar-la. Cada "." en una adreça (per exemple, `pirulo.remix.com`) indica una branca de l'arbre de DNS diferent i un àmbit de consulta (o de responsabilitat) diferent que es recorrerà recursivament d'esquerra a dreta.

Un altre aspecte important, a més del domini, és l'`in-addr.arpa` (*inverse mapping*), el qual també està imbricat com els dominis i serveix per a obtenir noms quan es consulta per l'adreça IP. En aquest cas, les adreces s'escriuen a l'inrevés, en concordança amb el domini. Si `pirulo.remix.com` és la 192.168.0.1, llavors s'ha d'escriure com 1.0.168.192, en concordança amb `pirulo.remix.com`. Després configurarem el domini propi `remix.bogus` a l'arxiu `/etc/bind/db.127` [11]:

```
; BIND reverse data file for local loopback interface
$TTL 604800
@ IN SOA ns.remix.bogus. root.remix.bogus. (
    1         ; Serial
    604800        ; Refresh
    86400         ; Retry
    2419200         ; Expire
    604800 )        ; Negative Cache TTL
@ IN NS ns.remix.bogus.
1.0.0 IN PTR localhost.
```

S'ha de tenir en compte el "." al final dels noms de domini. L'origen de la jerarquia d'una zona està especificat per la identificació de la zona, en el nostre cas `127.in-addr.arpa`. Aquest arxiu (`db.127`) conté tres registres: `SOA`, `NS` i `PTR`. L'**SOA** (*Start of Authority*) ha d'estar en tots els arxius de zona a l'inici, després de `TTL`, i el símbol `@` significa l'origen del domini; **NS**, el servidor de noms per al domini, i **PTR** (*Domain Name Pointer*), que és l'amfitrió 1 a la subxarxa (127.0.0.1) i s'anomena *local host*. Aquest és l'arxiu sèrie 1 i el responsable d'aquest és `root@remix.bogus` (últim camp de la línia `SOA`). Ara es podria reiniciar el `named` de la manera indicada abans i, amb el `dig -x 127.0.0.1`, en veure el funcionament (que seria idèntic al que hem mostrat anteriorment). A continuació s'ha d'afegir una nova zona en el `named.conf`:

```
zone "remix.bogus" {
    type master;
    notify no;
    file "/etc/bind/remix.bogus";
};
```

Cal recordar que, en el `named.conf`, els dominis van sense el "." final. En l'arxiu `remix.bogus` s'han de posar els *hosts*, dels quals serem responsables:

```
; Zone file for remix.bogus
$TTL 604800
@ IN SOA ns.remix.bogus. root.remix.bogus. (
    199802151         ; serial, todays date + todays serial
    604800         ; Refresh
    86400         ; Retry
    2419200         ; Expire
    604800 )         ; Negative Cache TTL
@ NS ns         ; Inet Address of name server
    MX 10 mail.remix.bogus.         ; Primary Mail Exchanger
localhost         A         127.0.0.1
ns         A         192.168.1.2
mail         A         192.168.1.4
    TXT "Mail Server"
ftp         A         192.168.1.5
    MX 10 mail
www         CNAME         ftp
```

Aquí apareix un nou registre `MX`, que és el *Mail eXchanger*. És el lloc on s'enviaran els correus electrònics que hi arribin, `alguien@remix.bogus`, i serà a `mail.remix.bogus` (el número indica la prioritat si tenim més d'un `MX`). Recordeu que "." sempre és necessari en els arxius de zona al final del domini (si no s'hi posa, el sistema afegeix el domini `SOA` al final, el que transformaria, per exemple, `mail.remix.bogus` en `mail.remix.bogus.remix.bogus`, que és incorrecte). `CNAME` (*canonical name*) és la manera de donar un o diversos àlies a una màquina. A partir d'aquest moment estaríem en condicions (després de `/etc/init.d/bind9 reload`) de provar, per exemple, el següent: `dig www.remix.bogus`.

El darrer pas és configurar la zona inversa, és a dir, perquè pugui convertir adreces IP en noms, per exemple, afegint-hi una ova zona:

```
zone "1.168.192.in-addr.arpa" {
    type master;
    notify no;
    file "/etc/bind/192.168.1";
};
```

I l'arxiu `/etc/bind/192.168.1` similar a l'anterior:

```
$TTL 604800
@ IN SOA ns.remix.bogus. root.remix.bogus. (
    199802151         ; serial, todays date + todays serial
    604800         ; Refresh
    86400         ; Retry
    2419200         ; Expire
    604800 )         ; Negative Cache TTL
@ NS         ns.remix.bogus.
2 PTR         ns.remix.bogus
4 PTR         mail.remix.bogus
5 PTR         ftp.remix.bogus
```

Aquest novament es podria provar amb `dig -x 192.168.1.4`. S'ha de tenir en compte que aquests exemples són d'IP privades, és a dir, no IP d'Internet. Una altra qüestió important és no oblidar el `notify no`, ja que en cas contrari, els nostres experiments amb DNS es propagaran als servidors de l'arbre de DNS (i modificaran fins i tot els DNS del nostre proveïdor o institució). Només s'ha de modificar quan estem segurs que funciona i volem propagar els canvis*. Una vegada creat un servidor mestre (*master server*), s'ha de crear un servidor esclau (*slave server*) per seguretat. Aquest és idèntic al mestre, excepte que la zona, en lloc de `type master`, ha de tenir `slave` i la IP del mestre. Per exemple:

***Per veure un exemple real, consulteu DNS-HOWTO a http://tldp.org/HOWTO/DNS-HOWTO-7.html.**

```
zone "remix.bogu" {
    type slave;
    notify no;
    masters {192.168.1.2; }
};
```

1.2. NIS (YP)

Per tal de facilitar l'administració i donar comoditat a l'usuari en xarxes de diferents mides que executen GNU/Linux (o Sun o qualsevol altre sistema operatiu amb suport per a aquest servei), s'executen serveis de *Network Information Service*, NIS (o *Yellow Pages*, YP, en la definició original de Sun). GNU/Linux pot donar suport com a client-servidor de NIS i pot actuar com a client (versió "beta") de NIS+, que és una versió més segura i optimitzada de NIS. La informació que es pot distribuir en NIS és: usuaris (*login names*), contrasenyes (*passwords*, `/etc/passwd`), directoris d'usuari (*home directories*) i informació de grups (*group information*, `/etc/group`), la qual cosa té l'avantatge que, des de qualsevol màquina client o des del mateix servidor, l'usuari es podrà connectar amb el mateix compte i la mateixa contrasenya i al mateix directori (encara que el directori s'haurà de muntar prèviament a totes les màquines client per NFS o mitjançant el servei d'`automount`). [17, 10]

L'arquitectura NIS és del tipus client-servidor, és a dir, hi ha un servidor que disposarà de totes les bases de dades i uns clients que consulten aquestes dades de manera transparent per a l'usuari. Per això, s'ha de pensar en la possibilitat de configurar servidors "de reforç"(anomenats *secundaris*) perquè els usuaris no quedin bloquejats davant la caiguda del servidor principal. És per això que l'arquitectura s'anomena realment *de múltiples servidors* (*master+mirrors-clients*).

1.2.1. Com s'ha d'iniciar un client local de NIS en Debian?

Un client local annexa l'ordinador a un domini NIS ja existent: primer s'ha de verificar que s'hi han instal·lat els paquets `netbase` (xarxa bàsica TCP/IP), `portmap` (servidor que converteix números RPC (crida a procediment remot o *remote procedure call*, en ports DARPA) i `nis` (específic). Cal verificar la instal·lació dels dos primers paquets per a tots els programes que executen RPC, incloent-hi NFS i NIS. Es recomana fer servir l'ordre `apt-get` (o també l'ordre `kpackage`) i es pot verificar si està instal·lat amb `apt-cache pkgnames` en mode text. El procediment d'instal·lació del paquet NIS demanarà un domini (`NIS domainname`). Aquest és un nom que descriu el conjunt de màquines que utilitzen el NIS (no és un nom d'amfitrió). Cal tenir en compte que `NISPirulo` és diferent de `Nispirulo` com a nom de domini. Per a configurar-lo es pot utilitzar l'ordre `nisdomainname`, domini que s'emmagatzemarà a `/proc/sys/kernel/domainname`. En primer lloc, el servei `portmap` s'ha d'iniciar amb:

```
/etc/init.d/portmap start
```

Es pot comprovar si aquests serveis estan actius amb `rpcinfo -p`. Si el servidor NIS no és local, s'haurà d'utilitzar l'ordre `ypbind`, que es fa servir per a trobar un servidor per al domini especificat, sigui mitjançant `broadcast` (no s'aconsella perquè no és segur) o buscant el servidor indicat a l'arxiu de configuració `/etc/yp.conf` (recomanable).

Sintaxi de l'arxiu `/etc/yp.conf`

`domain nisdomain server hostname`: indica que s'utilitza el *hostname* per al domini `nisdomain`. Es podria tenir més d'una entrada d'aquest tipus per a un únic domini.

`domain nisdomain broadcast`: indica que s'utilitza *broadcast* sobre la xarxa local per a descobrir un servidor de domini `nisdomain`.

`ypserver hostname`: indica que s'utilitza *hostname* com a servidor. És recomanable utilitzar aquesta línia (`ypserver`) en la qual s'ha d'introduir l'adreça IP del servidor NIS. Si s'indica el nom, assegureu-vos que es pot trobar la IP per DNS o que aquesta consta a l'arxiu `/etc/hosts` ja que, d'altra manera, el client es bloquejarà.

Per a iniciar el servei s'ha d'executar:

```
/etc/init.d/nis stop     per a aturar-lo
/etc/init.d/nis start     per a iniciar-lo
```

A partir d'aquest moment, el client NIS estarà funcionant. Això es pot confirmar amb `rpcinfo -u localhost ypbind`, que mostrarà les dues versions del protocol actiu, o es pot utilitzar l'ordre `ypcat mapname` (per exemple, `ypcat passwd`, que mostrarà els usuaris NIS definits en el servidor) on les relacions entre *mapnames* i les taules de la base de dades NIS estan definides a `/var/yp/nicknames`.

1.2.2. Quins recursos cal especificar per a utilitzar el NIS?

Considerarem que tenim instal·lada una de les últimes distribucions de Debian (qualsevol a partir de Sarge) que suporta la Libc6 (igualment per a FC4 o superior) i es vol que els usuaris d'una màquina client puguin accedir a la informació del servidor. En aquest cas, s'ha d'orientar la consulta de l'inici de sessió (*login*) a les bases de dades adequades amb els passos següents:

1) Verifiqueu el fitxer `/etc/nsswitch.conf` i assegureu-vos que les entrades `passwd`, `group`, `shadow` i `netgroup` són similars a:

```
passwd: compat
group: compat
shadow: compat ...
netgroup: nis
```

Consulteu la sintaxi d'aquest arxiu a `man nsswitch.conf`.

2) Afegiu la línia següent a les màquines clients NIS al final de `/etc/passwd` (indicarà que si l'usuari no és local, ho preguntarà al servidor de NIS):

`+::::::` (un "+" i sis ":")

Cal tenir en compte que a l'`/etc/passwd` es pot utilitzar el `+` i el `?` davant de cada nom d'usuari a l'`/ etc/passwd` per a incloure o excloure l'inici de sessió d'aquests usuaris (*override*). Si s'utilitzen contrasenyes amb *shadow* (l'opció més segura, ja que no permet que un usuari normal pugui veure la contrasenya encriptada d'altres usuaris), s'ha d'incloure la següent línia al final de l'arxiu `/etc/shadow`:

`+:::::::::` (un "+" i vuit ":")

3) S'ha d'afegir també la línia següent al final d'`/etc/group`:

`+:::` (un "+" i tres ":")

4) Les cerques de *hosts* (*hosts lookups*) es faran mitjançant DNS (i no per NIS), per la qual cosa, per a aplicacions Libc6 en el fitxer `/etc/nsswitch.conf` caldrà canviar l'entrada `hosts` per la següent línia: `hosts: files dns`. O, si es prefereix fer-ho per NIS, `hosts: files nis`. Per a aplicacions Libc5, caldrà modificar el fitxer `host.conf`, i posar `order hosts, dns` o `order hosts, nis`, segons desitgi.

Amb aquesta configuració es pot establir una connexió local (sobre el client NIS) a un usuari que no estigui definit al fitxer `/etc/passwd`, és a dir, un usuari definit en una altra màquina (*ypserver*). Per exemple, es podria fer `ssh -l user localhost`, on `user` és un usuari definit a *ypserver*.

1.2.3. Com s'ha d'executar un *master NIS server*?

Considerem que la màquina té instal·lat el paquet `nis` i el `portmap` (aquest últim en funcionament) i que s'han creat les bases de dades del NIS (vegeu el subapartat 1.2.4.). Caldrà assegurar-se que, a l'`/etc/hosts`, totes les màquines que formaran part del domini estiguin en el format FQDN (*fully qualified domain name*), que és on s'indica la IP, el nom amb domini i el nom sense domini de cada màquina (per exemple, `192.168.0.1 pirulo.remix.com pirulo`). Això és necessari només al servidor, ja que el NIS no utilitza DNS. A més, hi és a l'arxiu `/etc/defaultdomain` amb el nom del domini escollit. No utilitzeu el vostre domini DNS per a no posar en risc la seguretat, excepte si configureu adequadament els arxius `/etc/ypserv.securenets` (que indiquen amb una parella `netmask/network` des de quin lloc es podran connectar els clients) i `/etc/ypserv.conf` (que fa un control més detallat perquè indica quins *hosts* poden accedir a quins mapes; per exemple: `passwd.byname` o `shadow.byname`).

Verifiqueu que existeix `NISSERVER = master en /etc/default/nis`. Es pot afegir el número de xarxa local a l'arxiu `/etc/ypserv.securenets` per motius de seguretat. Inicieu el servidor i executeu primer `/etc/init.d/nis stop` i després `/etc/init.d/nis start`. Aquesta sentència iniciarà el servidor (`ypserv`) i el *password daemon* (`yppasswdd`), l'activació del qual es podrà consultar amb `ypwich -d domain`.

1.2.4. Com s'ha de configurar un servidor?

La configuració del servidor es fa amb l'ordre `/usr/lib/yp/ypinit -m`. Tanmateix, cal verificar que existeix l'arxiu `/etc/networks`, que és imprescindi-

ble per a aquest *script*. Si aquest fitxer no existeix, creeu-ne un de buit amb `touch/etc/networks`. També es pot executar el client `ypbind` sobre el servidor; així, tots els usuaris entren per NIS, com es va indicar anteriorment, modificant el fitxer `/etc/passwd`, on totes les entrades normals abans de la línia `+::::::` seran ignorades pel NIS (només podran accedir-hi localment), mentre que les posteriors podran accedir pel NIS des de qualsevol client [17]. Considereu que, a partir d'aquest moment, les ordres per a canviar la contrasenya o la informació dels usuaris, com `passwd`, `chfn` o `adduser`, no són vàlides. En lloc seu s'hauran d'utilitzar ordres com ara `yppasswd`, `ypchsh` i `ypchfn`. Si es canvien els usuaris o es modifiquen els arxius esmentats, caldrà reconstruir les taules de NIS amb l'ordre `make` al directori `/var/yp` per a actualitzar les taules. Tingueu en compte que Libc5 no suporta `shadow passwd` (contrasenyes a l'arxiu `/etc/shadow`), per la qual cosa no s'ha d'utilitzar *shadow* amb NIS si teniu aplicacions amb Libc5. No hi haurà cap problema si teniu Libc6, que accepta NIS amb suport *shadow*. La configuració d'un servidor esclau és similar a la del mestre, excepte si `NISSERVER = slave` a `/etc/default/nis`. Sobre el mestre s'ha d'indicar que distribueixi les taules automàticament als esclaus, posant `NOPUSH = "false"` a l'arxiu `/var/yp/Makefile`. Ara s'ha d'indicar al mestre qui és el seu esclau, amb l'execució de:

```
/usr/lib/yp/ypinit -m
```

i introduir els noms dels servidors esclaus. Això reconstruirà els mapes, però no enviarà els arxius als esclaus. Per a això, sobre l'esclau, executeu:

```
/etc/init.d/nis stop
/etc/init.d/nis start
/usr/lib/yp/ypinit -s nombre_master_server
```

Així, l'esclau carregarà les taules des del mestre. També es podria posar en el directori `/etc/cron.d` l'arxiu NIS amb un contingut similar a (recordeu fer un `chmod 755 /etc/cron.d/nis`):

```
20 *  * * *  root  /usr/lib/yp/ypxfr_1perhour >/dev/null 2>&1
40 6 * * *    root  /usr/lib/yp/ypxfr_1perday  >/dev/null 2>&1
55 6,18 * * *    root  /usr/lib/yp/ypxfr_2perday  >/dev/null 2>&1
```

Així es garantirà que tots els canvis del mestre es transfereixin al servidor NIS esclau.

Actualització de les taules NIS

És recomanable que després d'usar `adduser` per a afegir un nou usuari sobre el servidor, executeu `make -C/var/yp` per a actualitzar les taules NIS (i sempre que es canviï alguna

característica de l'usuari, per exemple la paraula clau amb l'ordre `passwd`, que només canviarà la contrasenya local i no la de NIS). Per a provar que el sistema funciona i que l'usuari donat d'alta és al NIS, podeu fer `ypmatch userid passwd`, on `userid` és l'usuari donat d'alta amb `adduser` abans i després de haver fet el `make`. Per a verificar el funcionament del sistema NIS podeu utilitzar l'*script* d'http://tldp.org/HOWTO/NIS-HOWTO/verification.html, que permet una verificació més detallada del NIS.

1.3. Serveis de connexió remota: telnet i ssh

1.3.1. Telnet i telnetd

Telnet és una ordre (client) utilitzada per a comunicar-se interactivament amb un altre amfitrió que executa el dimoni telnetd. L'ordre telnet es pot executar com `telnet host` o interactivament com `telnet`, la qual posarà l'indicador (*prompt*) "telnet>" i després, per exemple, `open host`. Un cop establerta la comunicació, s'haurà d'introduir l'usuari i la contrasenya amb la qual es vol connectar el sistema remot. Hi ha diverses ordres (en mode interactiu) com `open`, `logout` i `mode` (s'han de definir les característiques de visualització), `close`, `encrypt`, `quit`, `setembre` i `unset`, o es poden executar ordres externes amb "!". Es pot utilitzar un fitxer `/etc/telnetrc` per a definicions per defecte o `.telnetrc` per a definicions d'un usuari particular (haurà d'estar al directori *home* de l'usuari).

El dimoni `telnetd` és el servidor de protocol telnet per a la connexió interactiva. Generalment, és el dimoni `inetd` que posa en marxa `telned`; es recomana incloure un `wrapper tcpd` (que utilitza les regles d'accés a `host.allow` i `host.deny`) en la crida a `telnetd` dins de l'arxiu `/etc/inetd.conf`. Per a incrementar la seguretat del sistema s'hauria d'incloure una línia com:

La seguretat del sistema s'estudia en el mòdul " Administració de seguretat".

```
telnet stream tcp nowait telnetd.telenetd /usr/sbin/tcpd /usr/bin/in.telnetd)
```

En algunes distribucions (per exemple, Debian 3.0 o superiors), la funció de `inetd` es pot reemplaçar per la de `xinetd`, que requereix la configuració de l'arxiu `/etc/xinetd.conf`. Si es vol posar en marxa `inetd`, a manera de prova, es pot usar la sentència `/etc/init.d/inetd.real start`. Si l'arxiu `/etc/uissue.net` està present, el `telnetd` en mostrarà el contingut a l'inici de la sessió. També es pot usar `/etc/security/access.conf` per a habilitar i deshabilitar inicis de sessió d'usuari, amfitrions o grups d'usuaris, segons es connectin.

L'arxiu `/etc/xinetd.conf` s'estudia en el mòdul "Administració de seguretat".

Cal recordar que, si bé la parella telnet-telnetd pot funcionar en mode *encrypt*, en les últimes versions (transferència de dades encriptades, que han d'estar compilades amb l'opció corresponent), és una ordre que ha quedat en l'oblit per la seva falta de seguretat (transmet el text en clar per la xarxa, la qual cosa permet la visualització del contingut de la comunicació des d'una altra màquina, per exemple, amb l'ordre `tcpdump`); no obstant això, es pot utilitzar en xarxes segures o situacions controlades.

Si no està instal·lat, es pot utilitzar (Debian) `apt-get install telnetd` i a continuació verificar que s'ha donat d'alta, o bé en `/etc/inetd.conf`, o bé en `/etc/xinetd.conf` (o en el directori en què estiguin definits els arxius; per exemple, `/etc/xinetd.d` segons s'indiqui en l'arxiu anterior amb la sentència `include/etc/xinetd.d`). L'arxiu `xinetd.conf` o també l'arxiu `/etc/xinetd.d/telentd` han d'incloure una secció com*:

***Qualsevol modificació a `xinetd.conf` haurà d'arrencar novament el servei amb `service xinetd restart`.**

```
service telnet {
disable = no
flags = REUSE
socket_type = stream
wait = no
user = root
server = /usr/sbin/in.telnetd
log_on_failure += USERID
}
```

SSL telnet(d)

Es recomana que, en comptes d'utilitzar `telnetd`, s'utilitzi `SSLtelnet(d)`, que substitueix `telnet(d)` i utilitza encriptació i autenticació per SSL, o que s'utilitzi SSH. El `SSLtelnet(d)` pot funcionar amb el `telnet(d)` normal en ambdues direccions, ja que a l'inici de la comunicació verifica si l'altre costat (*peer*) suporta SSL i, en cas contrari, continua amb el protocol telnet normal. Els avantatges respecte al `telnet(d)` són que les seves contrasenyes i dades no circulen per la xarxa en mode de text pla i ningú que utilitzi l'ordre abans esmentada (`tcpdump`) no podrà veure el contingut de la comunicació. També `SSLtelnet` es pot utilitzar per a connectar-se, per exemple, a un servidor web segur (per exemple `https://servidor.web.org`) simplement fent `telnet servidor.web.org 443`.

1.3.2. SSH, *Secure shell*

Un canvi aconsellable avui en dia és utilitzar `ssh` en lloc de `telnet`, `rlogin` o `rsh`. Aquestes tres últimes ordres no són segures (excepte `SSLtelnet`) per diverses raons. La més important és que tot el que es transmet per la xarxa, inclosos els noms d'usuaris i les contrasenyes, és en text pla (encara que hi ha versions de telnet-telnetd encriptats, ha de coincidir que tots dos ho siguin), de manera que qualsevol que tingui accés a aquesta xarxa, o a algun segment d'aquesta, pot obtenir tota aquesta informació i després suplantar la identitat de l'usuari. La segona raó és que aquests ports (telnet, rsh, etc.) són el primer lloc on un pirata (*cracker*) intentarà connectar-se. El protocol `ssh` (en la seva versió OpenSSH) proporciona una connexió encriptada i comprimida molt més segura que, per exemple, `telnet` (és recomanable utilitzar la versió 2.0 o versions superiors del protocol). Totes les distribucions actuals incorporen el client `ssh` i el servidor `sshd` per defecte.

ssh

Per executar l'ordre heu de fer:

```
ssh -l login name host o ssh user@hostname
```

Amb SSH es poden encapsular altres connexions com X11 o qualsevol altra TCP/IP. Si s'omet el paràmetre `-l`, l'usuari es connectarà amb el mateix usuari local i en ambdós casos el servidor demanarà la contrasenya per validar la identitat de l'usuari. SSH suporta diferents maneres d'autenticació (vegeu `man ssh`) basades en l'algorisme RSA i clau pública.

Amb l'ordre `ssh-keygen -t rsa|dsa` es poden crear les claus d'identificació d'usuari. L'ordre crea en el directori del `.shh` de l'usuari els fitxers* `id_rsa` i `id_rsa.pub`, les claus privada i pública, respectivament. L'usuari podria copiar la pública (`id_rsa.pub`) en l'arxiu `$HOME/.ssh/authorized_keys` del directori `.ssh` de l'usuari de la màquina remota. Aquest arxiu podrà contenir tantes claus públiques com llocs des d'on es vulgui connectar remotament a aquesta màquina. La sintaxi és d'una clau per línia i el seu funcionament és equivalent a l'arxiu `.rhosts` (encara que les línees tindran una mida considerable). Després d'haver introduït les claus públiques de l'usuari-màquina en aquest arxiu, aquest usuari es podrà connectar sense contrasenya des d'aquesta màquina.

***Per exemple, per a l'algorisme d'encriptació RSA.**

Normalment, si no s'han creat les claus, es demanarà a l'usuari una contrasenya, però com que la comunicació serà sempre encriptada, mai no serà accessible a altres usuaris que puguin escoltar a la xarxa. Per a més informació, consulteu `man ssh`. Per executar remotament una ordre, simplement feu:

```
ssh -l login name host_ordre_remota
Per exemple: ssh -l user localhost ls -al
```

sshd

L'`sshd` és el servidor (dimoni) per al `ssh` (si no estan instal·lats, es pot fer amb `apt-get install ssh`, que instal·l al servidor i al client). Junts reemplacen `rlogin`, `telnet`, `rsh` i proporcionen una comunicació segura i encriptada en dos *hosts* insegurs de la xarxa. Aquest s'arrenca generalment per mitjà dels arxius d'inicialització (`/etc/init.d` o `/etc/rc`) i espera connexions dels clients. El `sshd` de la majoria de les distribucions actuals suporta les versions 1 i 2 (o 3) del protocol SSH. Quan s'instal·la el paquet, es crea una clau RSA específica de l'amfitrió i quan el dimoni s'inicia, en crea una altra, l'RSA per a la sessió, que no s'emmagatzema en el disc i que canvia cada hora. Quan un client inicia la comunicació genera un número aleatori de 256 bits que està encriptat amb les dues claus del servidor i enviat. Aquest número s'utilitzarà durant la comunicació com a clau de sessió per a encriptar la comunicació que es transmetrà a través d'un algorisme d'encriptació estàndard. L'usuari pot seleccionar qualsevol dels algorismes disponibles oferts pel servidor. Hi ha algunes diferències (més segur) quan s'utilitza la versió 2 (o 3) del protocol. A partir d'aquest moment s'inicien alguns dels mètodes d'autenticació d'u-

suari descrits en el client o se li demana la contrasenya, però sempre amb la comunicació encriptada. Per a més informació, consulteu `man sshd`.

Túnel sobre SSH

Moltes vegades tenim accés a un servidor `sshd`, però per qüestions de seguretat no podem accedir a altres serveis que no estan encriptats (per exemple, un servei de consulta de correu POP3 o un servidor de finestres X11) o simplement volem connectar-nos a un servei al qual només es té accés des de l'entorn de l'empresa. Per això, és possible establir un túnel encriptat entre la màquina client (per exemple amb Windows i un client ssh anomenat `putty` de programari lliure) i el servidor amb `sshd`. En aquest cas, en vincular el túnel amb el servei, el servei veurà la petició com si vingués de la mateixa màquina. Per exemple, si volem establir una connexió per POP3 sobre el port 110 de la màquina remota, i que també té un servidor sshd, farem:

```
ssh -C -L 1100:localhost:110 usuari-id@host
```

Aquesta ordre demanarà la contrasenya per a l'usuari-id sobre l'amfitrió i un cop connectat s'haurà creat el túnel. Cada paquet que s'enviï a la màquina local sobre el port 1100 s'enviarà a la màquina remota *localhost* sobre el port 110, que és on escolta el servei POP3 (l'opció `-C` comprimeix el trànsit pel túnel).

Fer túnels sobre altres ports és molt fàcil. Per exemple, suposem que *només* tenim accés a un *remote proxy server* des d'una màquina remota (*remote login*), no des de la màquina local. En aquest cas, es pot fer un túnel per connectar el navegador a la màquina local. Considerem que tenim inici de sessió sobre una màquina passarel·la (*gateway*), que pot accedir a la màquina anomenada *proxy*, la qual executa l'*squid proxy server* sobre el port 3128. Executem:

```
ssh -C -L 8080:proxy:3128 user@gateway
```

Després de connectar-nos tindrem un túnel que escolta sobre el port local 8080 i que reconduirà el trànsit des de *gateway* cap a *proxy* al 3128. Per a navegar de manera segura només s'ha de fer `http://localhost:8080/`.

1.4. Serveis de transferència de fitxers: FTP

L'FTP (*File Transfer Protocol*) és un protocol client-servidor (sota TCP) que permet transferir arxius des de i cap a un sistema remot. Un servidor FTP és un ordinador que executa el dimoni `ftpd`.

Alguns llocs que permeten la connexió anònima fent servir l'usuari *anonymous* són generalment dipòsits de programes. En un lloc privat, caldrà un usuari i una contrasenya per accedir-hi. També és possible accedir a un servidor FTP amb un navegador i, generalment, avui dia els dipòsits de programes se substitueixen per servidors web (p. ex. Apache) o altres tecnologies com Bittorrent, que utilitza xarxes d'igual a igual (*peer to peer*, P2P) o servidors web amb mòduls de WebDav. No obstant això, aquest protocol es continua utilitzant en alguns casos i Debian, per exemple, ofereix accés amb usuari o contrasenya o la possibilitat de pujar arxius al servidor (si bé amb serveis web també és possible fer-ho). Per definició, el protocol (i els servidors/clients que l'implementen) d'FTP no és encriptat; les dades, els usuaris i les contrasenyes es transmeten en text clar per la xarxa, amb el risc que això implica. No obstant això, hi ha una sèrie de servidors/clients que suporten SSL i, per tant, encriptació.

1.4.1. Client FTP (convencional)

Un client FTP permet accedir a servidors FTP i hi ha una gran quantitat de clients disponibles. L'ús de l'FTP és summament simple. Des de l'indicador d'ordres, executeu:

```
ftp nom -servidor
```

O també `ftp` i després, de manera interactiva,: `open nom -servidor`

El servidor demanarà un nom d'usuari i una contrasenya (si accepta usuaris anònims, s'introduirà *anonymous* com a usuari i la nostra adreça de correu electrònic com a contrasenya), i a partir de l'indicador de l'ordre (després d'alguns missatges), podrem començar a transferir fitxers.

El protocol permet la transferència en mode ASCII o binari. És important decidir el tipus de fitxer que cal transferir perquè una transferència d'un binari en mode ASCII inutilitzarà el fitxer. Per a canviar d'un mode a un altre s'han d'executar les ordres `ascii` o `binary`. Les ordres útils del client FTP són l'`ls` (navegació en el directori remot), `get nom_del_fitxer` (per a descarregar fitxers) o `mget` (que admet `*`), `put nom_del_fitxer` (per a enviar fitxers al servidor) o `mput` (que admet `*`); en aquests dos últims s'ha de tenir permís d'escriptura sobre el directori del servidor. Es poden executar ordres locals si abans de l'ordre s'insereix un "`!`". Per exemple, `!cd/tmp` significa que els arxius que baixin a la màquina local es descarregaran en `/tmp`. Per a veure l'estat i el funcionament de la transferència, el client pot imprimir marques (*ticks*) que s'activen amb les ordres`hash` i `tick`. Hi ha altres ordres que es poden consultar en el full del manual (`man ftp`) o fent `help` dins del client.

Hi ha moltes alternatives per als clients, per exemple en mode text (`ncftp`, `lukemftp`, `lftp`, `cftp`, `yafc Yafc`) o en mode gràfic (`gFTP`, `WXftp`, `LLNL XFTP`, `guiftp`).

Enllaç d'interès

A la Wikipedia hi ha una comparativa de diversos clients FTP: http://en.wikipedia.org/wiki/Comparison_of_FTP_client_software

1.4.2. Servidors FTP

El servidor tradicional d'UNIX s'executa a través del port 21 i es posa en marxa pel dimoni `inetd` (o `xinetd`, segons es tingui instal·lat un o altre). En `inetd.conf` convé incloure el *wrapper* `tcpd` amb les regles d'accés en `host.allow` i el `host.deny` en la crida a l'`ftpd` per l'`inetd` per incrementar la seguretat del sistema. Quan rep una connexió, verifica l'usuari i la contrasenya i el deixa entrar si l'autenticació és correcta. Un FTP anònim treballa de manera diferent, ja que l'usuari només podrà accedir a un directori definit a l'arxiu de configuració i a l'arbre subjacent, però no cap a dalt, per motius de seguretat. Aquest directori generalment conté directoris `pub/, bin/, etc/` i `lib/` perquè el dimoni d'FTP pugui executar ordres externes per a peticions d'`ls`. El dimoni `ftpd` suporta els següents fitxers per a la seva configuració:

El tema de la seguretat del sistema s'estudia en el mòdul "Administració de seguretat".

- **/etc/ftpusers**: llista d'usuaris que no són acceptats al sistema. Un usuari per línia.
- **/etc/ftpchroot**: llista d'usuaris a qui es canviarà el directori base `chroot` quan es connectin. És necessari quan volem configurar un servidor anònim.
- **/etc/ftpwelcome**: anunci de benvinguda.
- **/etc/motd**: notícies després e l'inici de sessió.
- **/etc/nologin**: missatge que es mostra després de negar la connexió.
- **/var/log/ftpd**: *log* de les transferències.

Si en algun moment volem inhibir la connexió a l'FTP, es pot incloure l'arxiu `/etc/nologin`. L'`ftpd` mostra el seu contingut i acaba. Si hi ha un fitxer `.message` en un directori, l'`ftpd` el mostrarà quan s'hi accedeixi. La connexió d'un usuari passa per cinc nivells diferents:

1) tenir una contrasenya vàlida;

2) no aparèixer a la llista d'`/etc/ftpusers`;

3) tenir un intèrpret d'ordres (*shell*) estàndard vàlid;

4) si apareix a `/etc/ftpchroot`, se'l canviarà al directori `home` (també si és anònim o FTP);

5) si l'usuari és anònim o FTP, haurà de tenir una entrada en l'`/etc/passwd` amb usuari FTP, però s'hi podrà connectar especificant qualsevol contrasenya (per convenció s'utilitza l'adreça de correu electrònic).

És important prestar atenció al fet que un usuari habilitat únicament per a utilitzar el servei FTP no disposi d'un intèrpret d'ordres a l'entrada correspo-

nent d'aquest usuari a `/etc/passwd` per a impedir que tingui connexió, per exemple, per `ssh` o `telnet`. Per això, quan es creï l'usuari, caldrà indicar, per exemple:

```
useradd -d/home/nteum -s /bin/false nteum
```
i després: `passwd nteum`,

la qual cosa indicarà que l'usuari nteum no tindrà intèrpret d'ordres per a una connexió interactiva (si l'usuari ja existeix, es pot editar el fitxer `/etc/passwd` i canviar l'últim camp per `/bin/false`). A continuació, s'ha d'afegir com a última línia `/bin/false` a `/etc/shells`. Mourani descriu pas a pas com s'ha de crear un servidor FTP segur amb usuaris registrats i també un servidor FTP anònim per a usuaris no registrats [15]. Dos dels servidors no estàndards més comuns són el WUFTPD i el ProFTPD [4, 15].

Enllaços d'interès

Per a saber més sobre WUFTPD i ProFTPD podeu visitar les pàgines web: http://en.wikipedia.org/wiki/WU-FTPD i http://www.proftpd.org

Per a instal·lar el Proftpd sobre Debian, executeu l'ordre: `apt-get install proftpd`. Un cop descarregat `debconf`, us preguntarà si s'ha d'executar per `inetd` o manualment (és recomanable triar l'última opció). Si s'ha d'aturar el servei (per a canviar la configuració, per exemple): `/etc/init.d/proftpd stop` i per a modificar el fitxer: `/etc/proftpd.conf`. Un servidor (Debian) molt interessant és el PureFtpd (`pure-ftpd`), que és molt segur, permet usuaris virtuals, quotes, SSL/TSL i un conjunt de característiques interessants.

Enllaços d'interès

Per a configurar un servidor FTP en mode encriptat (TSL) o per a tenir accés anònim podeu consultar: http://www.debian-administration.org/articles/228.
D'altra banda, per a saber més sobre la instal·lació i la configuració de PureFTPd podeu consultar: http://www.debian-administration.org/articles/383.

1.5. Serveis d'intercanvi d'informació a nivell d'usuari

1.5.1. *mail transport agent* (MTA)

Un MTA s'encarrega d'enviar i rebre els correus des d'un servidor de correu electrònic cap a i des d'Internet, que implementa el protocol SMTP (*simple mail transfer protocol*). Debian utilitza per defecte `exim`, ja que és més fàcil de configurar que altres paquets MTA, com són `smail` o `sendmail` (aquest últim és un dels precursors). `Exim` presenta característiques avançades com ara rebutjar connexions de llocs d'*spam* coneguts; té defenses contra correu brossa (*junk mail*) o bombardeig de correu (*mail bombing*) i és extremadament eficient en el processament de grans quantitats de correus. S'executa amb `inetd` en una línia a l'arxiu de configuració `/etc/inetd.conf` (o també amb `xinetd`). `Exim` utilitza un fitxer de configuració a `/etc/exim/exim.conf`, que es pot modificar manualment, si bé és recomanable fer-ho amb un *shell script* anomenat `eximconfig`, per a configurar `exim` interactivament. Els valors de la configuració dependran de la situació de la màquina; tanmateix, la seva connexió és molt fàcil, ja que el mateix *script* suggereix valors per defecte. No obstant això, a `/usr/doc/exim` es poden trobar exemples de configuracions típiques.

Es pot provar si la configuració és vàlida amb `exim -bV` i, si hi ha errors a l'arxiu de configuració, el programa els mostrarà en pantalla o, si tot és correcte, només posarà la versió i la data. Per a provar si pot reconèixer una bústia (*mailbox*) local, es pot utilitzar:

```
exim -v -bt usuari_local
```

Amb aquesta ordre es mostraran les capes de transport utilitzades i l'adreça local de l'usuari. També es pot provar, amb un usuari remot, de reemplaçar l'usuari local per una adreça remota i veure'n el comportament. A continuació, s'ha d'intentar enviar un correu local i un altre de remot, i passar els missatges directament a `exim` (sense utilitzar un agent, per exemple `mailx`) i teclejar, per exemple (tot junt):

```
exim postmaster@SuDominio
From: user@dominio
To: postmaster@SuDominio
Subject: Test Exim
Missatge de prova
Crtl D
```

A continuació es poden analitzar els arxius de traça `mainlog` i `paniclog` a `/var/log/exim/`, per a veure'n el comportament i quins són els missatges d'error generats. Òbviament, també es pot connectar al sistema com a usuari *postmaster* (o a qui s'hagi enviat el missatge) i llegir els correus per a veure si tot és correcte. Una altra manera és executar-lo en mode *debug* i utilitzar com a paràmetre `-dNro`, on `Nro` és el nivell de *debug* (1-9). El paràmetre normal amb el qual s'ha de posar en marxa és `exim-bs`, sigui per `inetd` o per `xinetd`. També és possible executar-lo com a dimoni mitjançant `/etc/init.d/exim start` en sistemes que necessitin prestacions elevades per al tractament dels correus. Consulteu la documentació (inclosa en Debian al paquet *exim-doc-html*) per a configurar filtres, verificar *hosts*, *senders*, etc. També és interessant instal·lar el paquet `eximon`, que és un monitor de l'`exim` i permet que l'administrador vegi la cua de correus i registres i faci diferents accions amb els missatges en cua per a distribuir-los (*freezing*, *Bouncing*, *thawing*, etc.).

L'última versió d'Exim és **Exim4**. Es pot instal·lar amb `apt-get install exim4-daemon-heavy` (instal·leu també `exim4-config` que servirà per a configurar `exim4`); cal tenir en compte que, si bé hi ha diferents paquets amb diferents possibilitats, `exim4-daemon-heavy` és la més completa. Una petita diferència a tenir en compte en la seva configuració és que, en lloc de tenir una única configuració `exim.conf` (és el que tindrà si s'instal·la `exim` des de les fonts), el paquet `exim4-config` (és convenient instal·lar-lo per a configurar `exim4`) utilitza petits arxius de configuració en lloc d'un únic arxiu. Aquests arxius estaran a `/etc/exim4/conf.d/*` i es concatenaran en un únic arxiu (`/var/lib/exim4/config.autogenerated` per defecte) per `update-exim4.conf`.

Lectura recomanada

Quant a Exim4, és aconsellable que llegiu `/usr/share/doc/exim/README.Debian.gz` i `update-exim4.conf(8)`. Per a més informació, podeu consultar el *HowTo* disponible a: http://www.exim.org/docs.html.

1.6. *Internet Message Access Protocol* (IMAP)

Aquest servei permet accedir als correus allotjats en un servidor mitjançant un client de correu com ara Thunderbird o el client de correu de Seamonkey (ambdós a mozilla.org). Aquest servei suportat pel dimoni `imapd` (els actuals suporten el protocol IMAP4rev1) permet accedir a un arxiu de correu electrònic (*mail file*) que està en una màquina remota. El servei `imapd` es presta a través dels ports 143 (imap2) o 993 (imaps) quan suporta encriptació per SSL. Si s'utilitza `inetd`, aquest servidor es posa en marxa amb una línia a `/etc/inetd.conf` com:

```
imap2 stream tcp nowait root /usr/sbin/tcpd /usr/sbin/imapd
imap3 stream tcp nowait root /usr/sbin/tcpd /usr/sbin/imapd
```

En aquest exemple es crida el *wrapper* `tcpd`, que funciona amb `hosts.allow` i `hosts.deny` per a incrementar la seguretat. Les aplicacions més populars són `uw-imapd` (Universitat de Washington i instal·lat per defecte a Debian) o la seva versió segura `uw-imapd-ssl, cyrus-imap` o `courier-imap`. Per a provar que el servidor `imap` funciona, es podria utilitzar un client, per exemple Thunderbird, crear un compte d'usuari local, configurar-lo adequadament perquè es connecti sobre la màquina local i verificar el funcionament d'`imap`.

Sobre Debian, la versió d'`imap` s'ha compilat per a suportar MD5 com a mètode d'autenticació dels usuaris remots, per a xifrar les contrasenyes de connexió i evitar la suplantació d'identitat per *sniffing* a la xarxa (el client utilitzat per connectar-se al servidor `imap` també ha de suportar el mètode d'autenticació per MD5). El mètode és molt simple i segur, però el servidor ha de conèixer les contrasenyes en text pla dels usuaris de correu i per això es recomana utilitzar la versió d'`imapd` sobre SSL, que funciona sobre el port 993. El protocol `imaps`, igual que `ssh`, es basa en xifrar la comunicació per mitjà d'un certificat de l'amfitrió (el client utilitzat per connectar-se al servidor també ha de suportar aquest mètode de connexió, per exemple, Thunderbird). Per a configurar el servidor `imaps`, instal·leu el paquet `uw-imap-dssl` de Debian, que és el servidor `imap` amb suport SSL.

La instal·lació genera un certificat autofirmat vàlid per un any i l'emmagatzema a `/etc/ssl/certs/imapd.pem`. Aquest certificat es pot reemplaçar per un altre signat per una companyia certificadora o es pot generar un de propi amb OpenSSL (o aconseguir-ne un de gratuït, per exemple a la web d'StartCom: https://www.startssl.com/?app=1). És convenient deixar només l'entrada `imaps` a l'arxiu `/etc/inetd.conf` i treure les entrades imap2 i imap3, si únicament es vol que l'accés a `imap` sigui per SSL. Un altre protocol de característiques semblants que en el passat va ser molt popular, però que avui s'ha vist superat per IMAP, és POP (*post office protocol*), versions 2 i 3. La seva instal·lació i posada en marxa és anàloga a la d'IMAP. Hi ha multitud de servidors POP, però els més comuns són `courier-pop, cyrus-pop3d` i `ipopd` (Universitat de Washington), `qpopper, solid-pop3d`.

1.6.1. Aspectes complementaris

Suposem que, com a usuaris, tenim quatre comptes de correu en servidors diferents i volem que tots els missatges que arriben a aquests comptes es recullin en un compte únic, al qual puguem accedir externament, i que hi hagi també un filtre de correu brossa (*antispam*).

Primer s'ha d'instal·lar `Exim + Imap` i comprovar que funcionen. S'ha de tenir en compte que si s'instal·la `courier-imap` (que segons alguns autors és millor que `uw-imapd`), aquest funciona sobre un format de correu anomenat `maildir` i que s'hauria de configurar `Exim` perquè també funcioni sobre `maildir` amb la següent configuració en `/etc/exim/exim.conf` (o en la que correspongui, si es té exim4), canviant l'opció `mail_dir format = true` (els correus es desaran en el compte de l'usuari local en un directori anomenat `Maildir`). A continuació, s'ha de reiniciar el servidor `exim` amb `/etc/init.d/exim restart`, repetir la prova de funcionament, enviar-nos un missatge i verificar que es pot llegir amb un client que suporti `maildir`*.

***Per exemple, `mutt`; `mailx` no ho suporta. Consulteu http://www.mutt.org.**

Per a recollir els missatges de diferents comptes s'ha d'utilitzar `Fetchmail`, (que s'instal·la com `apt-get install fetchmail`). Després cal crear el fitxer `.fetchmailrc` al nostre `$HOME` (també es pot usar l'eina `fetchmailconf`), que haurà de ser més o menys així:

```
set postmaster "pirulo"
set bouncemail
set no spambounce
set flush
poll pop.domain.com proto pop3
    user 'user1' there with password 'secret' is pirulo here
poll mail.domain2.com
    user 'user5' there with password 'secret2' is 'pirulo' here
    user 'user7' there with password 'secret3' is 'pirulo' here
```

L'acció `set` indica a `Fetchmail` que aquesta línia conté una opció global (enviament d'errors, eliminació dels missatges dels servidors, etc.). A continuació s'especifiquen dos servidors de correu: un perquè comprovi si hi ha correu amb el protocol POP3 i un altre perquè provi d'usar diversos protocols per tal de trobar un que funcioni. Es comprova el correu de dos usuaris amb la segona opció de servidor, però tot el correu que es trobi s'envia a la gestió de cues (*spool*) de correu de `pirulo`. Això permet comprovar diverses bústies de diferents servidors com si es tractés d'una única bústia MUA. La informació específica de cada usuari comença amb l'acció `user`. El `Fetchmail` es pot posar en el `cron` (per exemple, en `/var/spool/cron/crontabs/pirulo` i afegir `1 * * * * /usr/bin/fetchmail -s`) perquè s'executi automàticament o executar-lo en mode dimoni (poseu `set daemon 60` a `.fetchmailrc` i executeu-lo una vegada, per exemple, en `autostart` de GNOME/KDE o al `.bashrc`, on s'executarà cada 60 segons).

Per a treure el correu brossa heu d'utilitzar SpamAssassin i podeu configurar Kmail o Evolution (consulteu la bibliografia per veure com configurar-lo) per-

Podeu instal·lar SpamAssassin mitjançant `apt-get install spamassassin`.

què l'executin. Per a configurar-lo heu de fer servir Procmail, que és una eina molt potent (permet repartir el correu, filtrar-lo, reenviar-lo automàticament, etc.). Un cop instal·lat (`apt-get install procmail`), a continuació s'ha de crear un fitxer anomenat `.procmailrc` en el `home` de cada usuari, que cridarà l'SpamAssassin:

```
# Poseu yes per als missatges de funcionament o depuració
VERBOSE=no
# Considerem que els missatges són a "~/.Maildir"), canvieu-ho si és un altre
PATH=/usr/bin:/bin:/usr/local/bin:.
MAILDIR=$HOME/Maildir
DEFAULT=$MAILDIR/

# Directori per emmgatzemar els fitxers
PMDIR=$HOME/.procmail
# Comenteu si no volem log de Procmail
LOGFILE=$PMDIR/log
# filtre d'Smap
INCLUDERC=$PMDIR/spam.rc
```

L'arxiu `~/.procmail/spam.rc` conté:

```
# si l'SpamAssassin no està al PATH, afegiu a la variable PATH el directori
:0fw: spamassassin.lock
| spamassassin -a

# Les tres línies següents mouran el correu Spam a un directori anomenat
# "spam-folder". Si es vol desar el correu a la safata d'entrada,
# per a filtrar-lo després amb el client, comenteu les tres línies.


:0:
* ^X-Spam-Status: Yes
spam-folder
```

L'arxiu `~/.spamassassin/user_prefs` conté algunes configuracions útils per a SpamAssassin (consulteu la bibliografia).

```
# user preferences file. Vegeu man Mail::SpamAssassin::Conf

# Llindar per reconèixer un Spam.
# Default 5, però amb 4 funciona una mica millor
required_hits 4

# Llocs dels quals considerarem que mai no arribarà Spam
whitelist_from root@debian.org
whitelist_from *@uoc.edu

# Llocs dels quals sempre arriba SPAM (separats per comes)
blacklist_from viagra@domini.com

# les adreces a Whitelist i Blacklist són patrons globals com:
# "amic@lloc.com", "*@isp.net", o "*.domain.com".

# Inserteu la paraula SPAM en l'assumpte (facilita l'aplicació de filtres).
# Si no desitgeu comentar la línea.
subject_tag [SPAM]
```

Enllaç d'interès

Per a més informació sobre `procmail` i el filtratge de missatges, consulteu: http://www.debian-administration.org/articles/242.

Això generarà una etiqueta `X-Spam-Status: Yes` en la capçalera del missatge si es creu que el missatge és *spam*. Després s'haurà de filtrar i posar-lo en una altra carpeta o esborrar-lo directament. Es pot usar el `procmail` per a filtrar missatges de dominis, usuaris, etc. Finalment, es pot instal·lar un cli-

ent de correu i configurar els filtres perquè seleccionin tots els correus amb `X-Spam-Status: Yes` i els esborri o els enviï a un directori. Després verificarem els falsos positius (correus identificats com a brossa, però que no ho són). Un aspecte complementari d'aquesta instal·lació és que si es vol tenir un servidor de correu per correu web (*webmail*, és a dir, poder consultar els correus del servidor amb un navegador sense haver d'instal·lar un client ni configurar-lo, com ara consultar un compte de Gmail o Hotmail) és possible instal·lar Squirrelmail (`apt-get install squirrelmail`) per a oferir aquest servei.

Enllaç d'interès

Sobre Squirrelmail en Debian, consulteu: http://www.debian-administration.org/articles/200.

Enllaç d'interès

Hi ha altres possibilitats, com instal·lar MailDrop en comptes de Procmail, Postfix en comptes d'Exim, o incloure Clamav/Amavisd com antivirus (Amavisd permet vincular Postfix amb SpamAssassin i Clamav). Per saber més coses sobre aquest tema podeu visitar la pàgina web http://www.debian-administration.org/articles/364.

1.7. Grups de discussió

Els missatges (*news*) o grups de discussió se suporten amb el protocol NNTP. Cal instal·lar un servidor de grups de discussió quan es vol llegir els missatges fora de línia, quan es vol tenir un repetidor dels servidors centrals o es vol un servidor mestre de missatges propi. Els servidors més comuns són INN o CNEWS, però són paquets complexos i destinats a grans servidors. Leafnode és un paquet USENET que implementa el servidor TNP, especialment indicat per a llocs amb grups reduïts d'usuaris, però on es vol accedir a gran quantitat de grups de notícies. Aquest servidor s'instal·la en la configuració bàsica de Debian i amb `dpkg-reconfigure leafnode` es poden reconfigurar tots els paràmetres, com els servidors centrals, el tipus de connexió, etc. Aquest dimoni es posa en marxa des d'`inetd` de manera semblant a l'`imap` (o amb `xinetd`). Leafnode suporta filtres mitjançant expressions regulars indicades (del tipus ^Newsgroups:. `*` [,] alt.flame$) a `/etc/news/leafnode/filters`, on per a cada missatge es compara la capçalera amb l'expressió regular i, si coincideixen, es rebutja el missatge.

La configuració d'aquest servidor és simple i tots els arxius han de ser propietat d'un usuari de missatges amb permís d'escriptura (s'ha de verificar que aquest propietari existeix a `/etc/passwd`). Tots els arxius de control, missatges i la configuració són a `/var/spool/news`, excepte la configuració del servidor mateix, que és al fitxer `/etc/news/leafnode/config`. Durant la configuració hi ha alguns paràmetres que s'han de configurar obligatòriament(per exemple, perquè el servidor pugui connectar amb els servidors mestres), com són `server` (servidor de missatges des d'on aquests s'obtenen i s'envien) i `expire` (nombre de dies després dels quals s'esborrarà un fil o sessió que ja ha estat llegit). A més, hi ha un conjunt de paràmetres d'àmbit general o específic del servidor que es podrien configurar opcionalment. Per a més informació, consulteu la documentació (`/usr/doc/leafnode/README.Debian` o `man leafnode`). Per a verificar el funcionament del servidor, es pot fer

`telnet localhost nntp` i, si tot funciona correctament, sortirà la identificació del servidor i es quedarà esperant una ordre. Com a prova, es pot introduir `help` (per a avortar, feu `Ctrl+` i després `Quit`).

1.8. *World Wide Web* (httpd)

Apache és un dels servidors més populars i amb més prestacions d'HTTP (protocol de transferència d'hipertext o *hypertext transfer protocol*). Apache té un disseny modular i suporta extensions dinàmiques de mòduls durant la seva execució. És molt configurable pel que fa al nombre de servidors i mòduls disponibles i suporta diversos mecanismes d'autenticació, control d'accés, metafitxers, *proxy caching*, servidors virtuals, etc. Amb mòduls (inclosos a Debian) és possible tenir PHP3, Perl, Java Servlets, SSL i altres extensions*.

***Podeu consultar la documentació a http://www.apache.org.**

Apache està dissenyat per a executar-se com un procés dimoni autònom. Així, crea un conjunt de processos fill que gestionen les peticions d'entrada. També es pot executar com un dimoni d'Internet per mitjà d'`inetd`, de manera que es posarà en marxa cada vegada que es rebi una petició. La configuració del servidor pot ser extremadament complexa, segons les necessitats (consulteu la documentació); tanmateix, aquí veurem una configuració mínima acceptable. Els arxius de configuració són a `/etc/apache` i són `httpd.conf` (arxiu principal de configuració), `srm.conf`, `access.conf` (aquests dos últims es mantenen per compatibilitat i la seva funcionalitat està en l'anterior), `mime.conf` (formats MIME) i `magic` (número d'identificació d'arxius). Els arxius de registre són a `/var/log/apache` i són `error.log` (registra els errors en les peticions del servidor), `access.log` (registra qui ha accedit i a què) i `apache.pid` (identificador del procés). Apache es posa en marxa des de l'*script* d'inici `/etc/init.d/apache` i els `/etc/rcX.d`, però es pot controlar manualment amb l'ordre `apachectl`. També es pot utilitzar l'ordre `apacheconfig` per a configurar el servidor. A Debian, els directoris per defecte són:

1) **/var/www**: directori de documents HTML

2) **/usr/lib/cgibin**: directori d'executables (cgi) pel servidor

3) **http://server.dominio/~user**: pàgines personals dels usuaris

4) **/home/user/public.html**: directori de pàgines personals

L'arxiu que es llegeix per defecte de cada directori és `index.html`. Un cop instal·lats els paquets `apache` i `apache-common`, Debian configura bàsicament el servidor i el posa en marxa. Es pot comprovar que funciona obrint un navegador (per exemple, el Konqueror) i posant `http://localhost` en la barra d'URL, i es carregarà la pàgina `/var/www/index.html`.

1.8.1. Configuració manual (mínima) d'`httpd.conf`

Veurem ara alguns dels paràmetres més importants en la configuració d'Apache (l'exemple es pren de la versió 1.X d'Apache i hi ha alguns canvis menors si s'utilitza la versió 2).

ServerType standalone
Recomanat, més eficaç
ServerRoot /etc/apache
On hi ha els arxius de configuració
Port 80
On el servidor escoltarà les peticions
User www-data
Els user *i* group *amb què s'executarà el servidor (important per seguretat) han de ser usuaris vàlids (poden estar* locked*)*
Group www-data
ServerAdmin webmaster@pirulo.remix.com
Adreça d'usuari que atendrà els errors
ServerName pirulo.remix.com
Nom del servidor enviat als usuaris (ha de ser un nom vàlid a /etc/host o DNS)
DocumentRoot /var/www
Directori on hi haurà els documents
Àlies /icons/ /usr/share/apache/icons/
On hi ha les icones
ScriptAlias /cgibin/ /usr/lib/cgibin/
On hi ha els script *CGI*

1.8.2. Apache 2.2 + SSL + PHP + MySQL

Una qüestió important per als servidors web dinàmics és aprofitar els avantatges d'Apache en mode segur (SSL), PHP (un llenguatge de programació usat generalment per a crear contingut per a llocs web) i MySQL+PHPAdmin, tot això funcionant conjuntament. Partirem de la instal·lació sobre un Debian, però no per mitjà de paquets `deb`, sinó des del programari baixat dels llocs respectius; així es pot repetir l'experiència sobre altres distribucions. Òbviament, després aquests paquets no es podran controlar per `apt` o un altre gestor de paquets. Cal anar amb compte amb les versions que poden canviar i no superposar la instal·lació a paquets ja instal·lats.

1) Descàrrega dels fitxers necessaris (per exemple, en el directori `/root -> cd /root`):

Apache: des d'http://httpd.apache.org/download.cgi: *httpd-2.2.4.tar.bz2*
PHP: des d'http://www.php.net/downloads.php PHP 5.2.1 (*tar.bz2*)
MySQL des d'http://mysql.org/get/Downloads/MySQL-4.1/mysql-standard-4.1.21-pc-linux-gnu-i686.tar.gz/from/pick
PHPAdmin des d'http://prdownloads.sourceforge.net/phpmyadmin/phpMyAdmin-2.9.1-all-languages.tar.bz2?download

2) Utilitats: `bzip2 libssl-dev openssl gcc g++ cpp make`. Verifiqueu que no estan instal·lades; en cas contrari, feu:

```
apt-get install bzip2 libssl-dev openssl gcc g++ cpp make
```

3) Apache:

```
cd /root
tar jxvf httpd-2.2.4.tar.bz2
cd httpd-2.2.4
```

Amb `prefix` indiquem que s'instal·larà, per exemple, */usr/local/apache2*

```
./configure --prefix=/usr/local/apache2 --with-ssl=/usr/include/openssl --enable-ssl
make
make install
```

Modifiquem el fitxer de configuració */usr/local/apache2/conf/httpd.conf* i canviem l'usuari i el grup de treball per `www-data`

```
User www-data
Group www-data
```

Canviem l'amo i el grup del directori de dades a `www-data`:

```
chown -R www-data:www-data /usr/local/apache2/htdocs
```

Modifiquem l'usuari www-data per canviar-ne el directori *home* a `/etc/passwd`:

```
www-data:x:33:33:www-data:/usr/local/apache2/htdocs:/bin/sh
```

Servidor Apache instal·lat. Per iniciar-lo (per aturar-lo, canvieu `start` per `stop`):

```
/usr/local/apache2/bin/apachectl start
```

Es pot col·locar un *script* per a arrencar el servidor Apache en l'arrencada.

```
ln -s /usr/local/apache2/bin/apachectl /etc/rcS.d/S99apache
chmod 755 /etc/rcS.d/S99apache
```

4) SSL:

A `/usr/local/apache2/conf/httpd.conf`, traiem el comentari de la línia:

```
Include conf/extra/httpd-ssl.conf
```

Es generen els fitxers amb les claus per al servidor segur. Cal adequar les versions a les que s'hagin descarregat. La primera ordre `openssl` és una línia sencera i acaba amb `1024`). A `/root` fem:

```
openssl genrsa -rand ../httpd-2.2.4.tar.bz2:../php-5.2.1.tar.bz2:../phpMyAdmin-2.9.1-all-languages.tar.bz2 -out server.key 1024
openssl rsa -in server.key -out server.pem
openssl req -new -key server.key -out server.csr
openssl x509 -req -days 720 -in server.csr -signkey server.key -out server.crt
```

Es copien els fitxers...

```
cp server.crt /usr/local/apache2/conf/
cp server.key /usr/local/apache2/conf/
```

Reiniciem el servidor...

```
/usr/local/apache2/bin/apachectl restart
```

Enllaç d'interès

Per saber com heu d'afegir el mòdul SSL a un servidor que no el tingui instal·lat, consulteu http://www.debian-administration.org/articles/349.

5) MySQL:

Creem un grup i un usuari per a MySQL, si no n'hi ha cap:

```
groupadd mysql
useradd -g mysql mysql
```

En el directori en què s'instal·larà MySQL (`/usr/local/`), fem:

```
cd /usr/local/
gunzip < /root/mysql-standard-4.1.21-pc-linux-gnu-i686.tar.gz | tar xvf -
ln -s mysql-standard-4.1.21-pc-linux-gnu-i686 mysql
cd mysql
```

Creem una base de dades i canviem els permisos:

```
scripts/mysql_install_db –user=mysql
chown -R root .
chown -R mysql data
chgrp -R mysql .
```

Es pot col·locar un *script* per a iniciar el servidor MySQL.

```
ln -s /usr/local/mysql/support-files/mysql.server /etc/rcS.d/S99mysql.server
chmod 755 /etc/rcS.d/S99mysql.server
```

Iniciem el servidor:

```
/etc/rcS.d/S99mysql.server start
```

Es pot entrar en la base de dades i canviar la contrasenya del *root* per seguretat*.

```
/usr/local/mysql/bin/mysql
```

Un cop dins, fem:

```
USE mysql
```

Co·loquem la contrasenya `pirulo` a l'usuari *root*

```
UPDATE user SET Password=PASSWORD('pirulo') WHERE User='root';
FLUSH privileges;
```

Per entrar a MySQL, farem:

```
/usr/local/mysql/bin/mysql -u root -ppirulo
```

***Consulteu http://dev.mysql.com/doc/refman/5.0/en/index.html per a la sintaxi**

6) PHP (reemplaceu amb les versions adequades):

Utilitats necessàries:

```
apt-get install libxml2-dev curl libcurl3-dev libjpeg-mmx-dev zlib1g-dev  libpng12-dev
```

Amb el servidor Apache aturat, fem:

```
cd /root
tar jxvf php-5.2.0.tar.bz2
cd php-5.2.0
```

Amb `prefix` podeu indicar on voleu instal·lar-lo (tota una línia):

```
./configure - -prefix=/usr/local/php5 - -enable-mbstring - -with-apxs2=/usr/local/apache2/bin/apxs
- -with-mysql=/usr/local/mysql - -with-curl=/usr/include/curl
- -with-jpeg-dir=/usr/include - -with-zlib-dir=/usr/include - -with-gd - -with-xml - -enable-
ftp - -enable-bcmath
make
make install
cp php.ini-dist /usr/local/php5/lib/php.ini
```

Modifiquem Apache (`/usr/local/apache2/conf/httpd.conf`) a la part indicada:

```
<IfModule mime_module>
    AddType application/x-httpd-php .php .phtml
    AddType application/x-httpd-php-source .phps
```

I també:

```
DirectoryIndex index.php index.html
```

Reiniciem el servidor:

7) PHPAdmin

```
cd /usr/local/apache2/
```

Es descomprimeix `phpmyadmin` en el directori d'`apache2` (aneu amb compte amb les versions):

```
tar jxvf /root/phpMyAdmin-2.9.1-all-languages.tar.bz2
mv phpMyAdmin-2.9.1-all-languages phpmyadmin
cd phpmyadmin
cp config.sample.inc.php config.inc.php
```

Cal modificar el fitxer de configuració (`config.inc.php`):

```
$cfg['blowfish_secret'] = 'pirulo';
```

Traiem l'usuari i la contrasenya de l'usuari per defecte, amb dues (') seguides:

```
$cfg['Servers'][$i]['controluser'] = '';
$cfg['Servers'][$i]['controlpass'] = '';
```

Canviem Apache (`/usr/local/apache2/conf/httpd.conf`) afegint a *<IfModule alias_module>*:

```
<IfModule alias_module>
```

```
    Alias /phpmyadmin "/usr/local/apache2/phpmyadmin/"
<Directory "/usr/local/apache2/phpmyadmin/">
    Order allow,deny
    Allow from all
   </Directory>
```

Reiniciem el servidor, que es pot cridar amb `http://localhost/phpadmin`.

Enllaç d'interès

Podeu obtenir-ne més informació als llocs respectius de cada aplicació i a LWP. La adreça és: http://www.lawebdelprogramador.com/temas/tema_stablephpapachemysql.php.

1.9. Servidor de WebDav

WebDav és la sigla de *Web Based Distributed Authoring and Versioning* (també es refereix al grup de treball d'Internet Engineering Task Force). Aquest protocol permet que la web es transformi en un mitjà llegible i editable i proporciona funcionalitats per a crear, canviar i moure documents en un servidor remot (típicament un servidor web). Això s'utilitza sobretot per a permetre l'edició dels documents que envia un servidor web, però també es pot aplicar a sistemes d'emmagatzematge generals basats en la web i als quals es pot accedir des de qualsevol lloc. En aquest subapartat instal·larem un servidor WebDav sobre Apache. El procés és el següent:

Enllaç d'interès

Sobre la integració de WebDav amb Apache, podeu consultar l'article "WebDAV on Apache2" disponible a: http://www.debian-administration.org/articles/285

1) Instal·leu el servidor Apache, feu les configuracions mínimes i verifiqueu que funciona (instal·leu-lo, per exemple, amb `apt-get install apache2`).

2) Habiliteu els mòduls d'Apache que són necessaris per a WebDav: `a2enmod dav_fs` i `a2enmod dav`.

3) Creeu el directori per al directori virtual (podeu fer, per exemple, `mkdir -p /var/www/test`) i permeteu que Apache sigui el propietari del directori `chown www-data /var/www/test/`.

4) En l'arxiu `/etc/apache2/sites-available/default` creeu un directori virtual,

```
<VirtualHost *>
   ServerAdmin root@localhost
   DocumentRoot /var/www/test/
   <Directory /var/www/test>
      Options Indexes MultiViews
      AllowOverride None
      Order allow,deny
      allow from all
   </Directory>
   Alias /webdav /var/www/test
   <Location /webdav>
      DAV On
      AuthType Digest
      AuthName "webdav"
      AuthDigestProvider file
```

```
        AuthUserFile /var/www/test/digest-password
        Require valid-user
    </Location>
</VirtualHost>
```

en el qual es pot veure que s'ha decidit utilitzar una autenticació segura amb MD5 Digest. Podeu comprovar que la configuració és correcta amb l'ordre `apache2ctl configtest`.

5) Creeu el sistema d'autenticació en habilitar el mòdul `digest` d'Apache `a2enmod auth_digest` i creeu la contrasenya per a un usuari: `htdigest -c /var/www/test/digest-password webdav admin`, que ens demanarà la contrasenya per a l'usuari *admin* (dues vegades per a verificar-lo) i el desarà en el fitxer encriptat.

6) Es reinicia Apache perquè llegeixi la configuració `/etc/init.d/apache2 reload` i ja ens podem connectar a `http://localhost/webdav`, després d'have-nos autenticat.

7) Una altra manera de provar-ho és amb un client WebDav, per exemple Cadaver*, amb `apt-get install cadaver`. A continuació, ens connectem al servidor amb `cadaver http://localhost/webdav` i després d'autenticar-nos, podem crear un directori (`mkdir`), editar un fitxer, llistar un directori (`ls`), canviar de directori (`cd`), canviar els permisos d'execució (`chexec`), esborrar (`rm`), etc.

*http://www.webdav.org/cadaver

1.10. Servei de *proxy*: Squid

Un servidor intermediari (*proxy server*, PS) s'utilitza per a estalviar amplada de banda de la connexió de xarxa, millorar la seguretat i incrementar la velocitat en navegar per la xarxa (*web-surfing*).

Squid és un dels principals PS, ja que és *OpenSource* i accepta ICP (característiques que li permeten intercanviar *hints* amb altres PS), SSL (per a connexions segures entre intermediaris) i suporta objectes FTP, Gopher, HTTP i HTTPS (segur). El seu funcionament és simple: emmagatzema els objectes més demanats en memòria RAM i els menys en una base de dades en el disc. Els servidors Squid, a més, es poden configurar de manera jeràrquica per a formar un arbre d'intermediaris, depenent de les necessitats. Hi ha dues configuracions possibles:

1) com a accelerador d'`httpd` per a aconseguir més prestacions al servei de la web;

2) com a servidor cau (*proxy-caching server*) per a permetre que els usuaris d'una corporació utilitzin el PS per a sortir a Internet.

En la primera, actua com a intermediari invers, és a dir, accepta una petició del client i serveix l'objecte, si el té. Si no el té, el demana i el passa

al client quan el té i l'emmagatzema per a la pròxima vegada. En la segona opció, es pot utilitzar com a control i per a restringir els llocs on es pot connectar a Internet o autorizar l'accés a determinades hores del dia. Un cop instal·lat (paquet `squid` a Debian, també es pot instal·lar `squid-cgi`, `squidguard` o `squidtaild`), es generen tres arxius: `/etc/squid.conf` (configuració), `/etc/init.d/squid` (inicialització) i `/etc/logrotate.d/squid` (de control dels registres).

1.10.1. Squid com a accelerador d'http

En aquest mode, si el servidor de web està en la mateixa màquina que el PS, aleshores s'haurà de reconfigurar perquè atengui peticions del port 81 (en Apache, canvieu Port 80 per Port 81 a `httpd.conf`). L'arxiu de configuració (`/etc/squid.conf`) conté una gran quantitat d'entrades, però aquí només veurem les indispensables [15]:

```
http_port 80     On escolta httpd
icp_port 0     On escolta ICP
hierarchy_stoplist cgi-bin \?
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY
cache_mem 100 MB     Memòria per a objectes en curs
redirect_rewrites_host_header off
cache_replacement_policy lru
memory_replacement_policy lru
cache_dir ufs /var/spool/squid 100 16 256
    Tipus i lloc on està la base de dades cau de disc
emulate_httpd_log on
acl all src 0.0.0.0/0.0.0.0     Accés per a tothom
http_access allow all     I a tot
cache_mgr root     Mail responsable
cache_effective_user proxy     UID
cache_effective_group proxy     GID
httpd_accel_host 192.168.1.1     Servidor real de web
httpd_accel_port 81     Port
logfile_rotate 0
log_icp_queries off
buffered_logs on
```

D'aquesta manera, l'opció `httpd_accel_host` desactiva la possibilitat que s'executi com a servidor cau*.

*Per a més informació, consulteu http://www.squid-cache.org.

1.10.2. Squid com a servidor cau

D'aquesta manera s'habilita l'Squid perquè controli l'accés a Internet, quan i a què accediran els usuaris. En aquest cas, l'arxiu de configuració ha d'incloure les modificacions o afegits següents a `/etc/squid.conf`:

```
acl localnet src 192.168.1.0/255.255.255.0
acl localhost src 127.0.0.1/255.255.255.255
acl Safe_ports port 80 443 210 70 21 102565535
acl CONNECT method CONNECT
acl all src 0.0.0.0/0.0.0.0
http_access allow localnet
```

```
http_access allow localhost
http_access deny Port segur
http_access deny CONNECT
http_access deny all
cache_emulate_httpd_log on
```

La gran diferència amb l'altra manera són les línies `acl`, que, en aquest cas, permetran que els clients de la classe C 192.168.1.0 accedeixin al PS; també el *localhost* IP i altres ports que podran accedir a Internet 80(http), 443(https), 210(wais), 70(gopher) i 21(ftp). A més, es nega el mètode `connect` per a evitar que des de fora es puguin connectar al PS i després es neguen tots els IP i ports sobre el PS [15].

Enllaços d'interès

Més informació a http://www.squid-cache.org/ i per a un intermediari transparent, a http://tldp.org/HOWTO/TransparentProxy-1.html.

1.11. OpenLdap (Ldap)

LDAP significa 'protocol d'accés a directoris lleugers' (*lightweight directory access protocol*) i és un protocol per a accedir a dades basat en un servei X.500. Aquest s'executa sobre TCP/IP i el directori és similar a una base de dades que conté informació basada en atributs. El sistema permet organitzar aquesta informació de manera segura i utilitza rèpliques per a mantenir la seva disponibilitat, cosa que assegura la coherència i la verificació de les dades a les quals s'ha accedit o modificat.

El servei es basa en el model client-servidor, on hi ha un o més servidors que contenen les dades; quan un client es connecta i demana informació, el servidor respon amb les dades o amb un punter a un altre servidor, d'on podrà extreure més informació. Tanmateix, el client només veurà un directori d'informació global [15, 18]. Per a importar i exportar informació entre servidors `ldap` o per a descriure una sèrie de canvis que s'aplicaran al directori, el format utilitzat és LDIF (LDAP *data interchange format*). LDIF emmagatzema la informació en jerarquies orientades a objectes que després es transformaran al format intern de la base de dades. Un arxiu LDIF té un format similar a:

```
dn: o = UOC, c = SP o: UOC
objectclass: organization
dn: cn = Pirulo Nteum, o = UOC, c = SP
cn: Pirulo Nteum
sn: Nteum
mail: nteumuoc.edu
objectclass: person
```

Cada entrada s'identifica amb un nom indicat com a `DN` (*distinguished name*). El DN consisteix en el nom de l'entrada més una sèrie de noms que el relacionen amb la jerarquia del directori i on hi ha una classe d'objectes (`objectclass`) que defineix els atributs que es poden utilitzar en aquesta entrada. LDAP proporciona un conjunt bàsic de classes d'objectes: **grups** (inclou llistes desordenades d'objectes individuals o grups d'objectes), **localitzacions** (com ara països i la seva descripció), **organitzacions i persones**. Una entrada

pot, a més, pertànyer a més d'una classe d'objecte, per exemple, un individu es defineix per la classe persona, però també es pot definir per atributs de les classes `inetOrgPerson`, `groupOfNames` i `organization`.

L'estructura d'objectes del servidor (anomenat *schema*) determina quins són els atributs permesos per a un objecte d'una classe (que es defineixen en el fitxer `/etc/ldap/schema` com `inetorgperson.schema, opeldap.schema, corba.schema, nis.schema`, etc.). Totes les dades es representen com una parella atribut = valor, en què l'atribut és descriptiu de la informació que conté; per exemple, l'atribut utilitzat per a emmagatzemar el nom d'una persona és `commonName`, o cn, és a dir, una persona anomenada Pirulo nteum es representarà per cn: Pirulo nteum i tindrà associats altres atributs de la classe persona com `givenname: Pirulo Surname: nteum mail: pirulo@uoc.edu`. En les classes hi ha atributs obligatoris i optatius, i cada atribut té una sintaxi associada que indica quin tipus d'informació conté l'atribut, per exemple, `bin` (*binary*), `ces` (*case exact string*, s'ha de buscar igual), `cis` (*case ignore string*, poden ignorar majúscules i minúscules durant la cerca), `tel` (*telephone number string*, s'ignoren espais i '-') i `dn` (*distinguished name*). Un exemple d'un arxiu en format LDIF podria ser:

```
dn: dc = UOC, dc = com
objectclass: top
objectclass: organizationalUnit


dn: ou = groups, dc = UOC, dc = com
objectclass: top
objectclass: organizationalUnit
ou: groups


dn: ou = people, dc = UOC, dc = com
objectclass: top
objectclass: organizationalUnit
ou: people


dn: cn = Pirulo Nteum, ou = people, dc = UOC, dc = com
cn: Pirulo Nteum
sn: Nteum
objectclass: top
objectclass: person
objectclass: posixAccount
objectclass: shadowAccount
   uid:pirulo userpassword:{crypt}p1pss2ii(0pgbs*do& = )eksd uidnumber:104
   gidnumber:100
   gecos:Pirulo Nteum
   loginShell:/bin/bash
   homeDirectory: /home/pirulo
   shadowLastChange:10877
   shadowMin: 0
   shadowMax: 999999
   shadowWarning: 7
   shadowInactive: -1
   shadowExpire: -1
   shadowFlag: 0


dn:
cn = unixgroup, ou = groups, dc = UOC, dc = com
objectclass: top
```

```
objectclass: posixGroup
cn: unixgroup
   gidnumber: 200
   memberuid: pirulo
   memberuid: altre usuari
```

Les línies llargues poden continuar avall i començar amb un espai o un tabulador (format LDIF). En aquest cas, s'ha definit la base DN per a la institució `dc = UOC, dc = com`, la qual conté dues subunitats: `people` i `groups`. A continuació, s'ha descrit un usuari que pertany a `people` i a `group`. Una vegada preparat l'arxiu amb les dades, aquest s'ha d'importar al servidor perquè estigui disponible per als clients LDAP. Hi ha eines per a transferir dades de diferents bases de dades a format LDIF [18]. Sobre Debian, s'ha d'instal·lar el paquet `slapd`, que és el servidor d'OpenLdap. Durant la instal·lació farà diverses preguntes com ara el mètode d'instal·lació del directori: auto; les extensions al directori (*domain-host*, lloc, institució): amfitrió, domini, contrasenya de l'administrador; replicació dels canvis locals a altres servidors: no. Aquesta instal·lació generarà un arxiu de configuració a `/etc/ldap/slapd.conf` i la base de dades sobre `/var/lib/ldap`. Per altra banda, també hi ha un altre arxiu `/etc/ldap/ldap.conf` (o hi pot haver el `$HOME/.ldaprc`), que és l'arxiu de configuració usat per a inicialitzar valors per defecte quan s'executen clients ldap. En aquest s'indica quina és la base de dades, quin és el servidor ldap, els paràmetres de seguretat, les dimensions de la cerca, etc.

L'arxiu de configuració del servidor `/etc/ldap/slapd.conf` (vegeu la pàgina de manual `man slap.conf`) està compost per diferents seccions, cadascuna indicada per una de les directives següents: *global*, *backend specific* i *database specific*, en aquest ordre. La directiva global és de caràcter general, s'aplica a tots els *backends* (bases de dades) i defineix qüestions generals com els permisos d'accés, els atributs, els temps d'espera, els *schema*, etc. La directiva *backend specific* defineix els atributs al processador de fons (*backend*) específic que defineix (`bdb`, `dnssrv`, `ldbm`, etc.) i el *database specific*, els atributs específics per a aquesta base de dades que defineix. Per a posar en marxa el servidor, s'ha d'executar `/etc/init.d/slapd start` (o `/etc/init.d/slapd stop` per a aturar-lo). Durant la instal·lació, el sistema haurà creat els enllaços adequats per a executar-lo després de l'inici.

1.11.1. Creació i manteniment de la base de dades

Hi ha dos mètodes per a introduir dades a la base de dades de LDAP. El primer és fàcil i adequat per a un nombre baix de dades, és interactiu i, per a afegir-hi noves entrades, s'han d'utilitzar eines com ldapadd, gq, phpldapadmin, JXplorer, etc. Amb el segon s'ha de treballar fora de línia, és l'adequat per a BD grans i s'utilitza l'ordre `slapadd` inclosa amb slapd. Com que és més general, en aquest subapartat descriurem sintèticament el segon mètode (deixarem el primer per a un cas d'ús que explicarem a continuació), en què primer s'ha de verificar que contingui els següents atributs a `slapd.conf`: `suffix` (*top* del

directori, per exemple, sufix "o = nteum, c = org"); directori `/var/lib/ldap` (directori en què es crearan els índexs i que pugui escriure slapd). Cal verificar, a més, que la base de dades contingui les definicions dels índexs que es volen:

```
index cn,sn,uid
index objectClass pres,eq
```

Una vegada definit l'`slapd.conf`, s'ha d'executar l'ordre:

```
slapadd -l entrada -f configuració [-d nivell] [-n
sencer | -b sufix]
```

Els arguments són:

-l: arxiu en format LDFI.
-f: arxiu de configuració del servidor, on s'indica com s'han de crear els índexs.
-d: nivell de depuració.
-n: núm. de base de dades, si n'hi ha més d'una.
-b: especifica quina base de dades cal modificar.

Hi ha altres ordres amb slapd, com ara `slapindex`, que permet regenerar els índexs, i `slapcat`, que permet transferir la BD a un arxiu en format LDIF.

1.11.2. Instal·lació (bàsica) del servidor

En primer lloc, hem de tenir ben configurat l'amfitrió pel que fa al nom (hostname o FQDN, Fully Qualified Domain Name); per exemple, en la màquina que s'ha utilitzat en les proves (debian.nteum.org), s'ha configurat `hostname debian` i a `/etc/hosts` s'ha afegit l'adreça IP de la màquina `10.0.2.15 debian.nteum.org debian`. Per a no tenir problemes posteriors durant la configuració, també cal definir la variable `ServerName debian.nteum.org` en el servidor d'Apache (arxiu /etc/apache2/httpd.conf).

Lectura recomanada

Sobre la instal·lació bàsica del servidor podeu llegir el document *Redes de área local: Aplicaciones y Servicios Linux*, disponible a: http://www.isftic.mepsyd.es/formacion/materiales/85/cd/REDES_LINUX/index.htm

El servidor OpenLDAP està disponible en el paquet slapd i per a instal·lar-lo s'ha de fer `apt-get install slapd dbx.x-util ldap-utils` (això també instal·la `dbx.x-util`, que són les utilitats per a la base de dades dbd i s'han de reemplaçar les `x.x` pel número de versió, per exemple `db4.2-util`). Durant la instal·lació ens demanarà que hi introduïm la contrasenya d'administrador del servidor ldap (per exemple 'ldapadmin'). Com ja hem comentat, la configuració del servidor LDAP s'emmagatzema en l'arxiu */etc/ldap/slapd.conf* i es pot editar manualment o utilitzar l'assistent de configuració de slapd: `dpkg-reconfigure slapd`. El directori LDAP ha de tenir una base, a partir de la qual pengen la resta d'elements. Com a nom de la base habitualment s'utilitza el nom del domini; així, si el nostre domini és nteum.org, la base per

al nostre directori LDAP serà dc=nteum, dc=org. L'*script* d'inicialització ens demanarà:

1) ometre la configuració (resposta, NO),

2) el domini (i l'utilitzarà per a crear el nom distingit, DN),

3) el nom de la nostra organització (nteum),

4) la contrasenya de l'usuari admin (administrador) del servidor LDAP,

5) els possibles gestors de dades per a emmagatzemar el directori (és recomanable utilitzar el sistema BDB),

6) si volem que s'elimini la base de dades quan s'elimini slapd (recomanable, NO),

7) si hi ha una base de dades LDAP prèvia, en cas que la vulguem moure (recomanat, SÍ) i

8) si es vol utilitzar LDAP versió 2 (recomanat, NO) i el servidor LDAP ja estarà configurat.

Per a arrencar el servidor LDAP es pot usar `/etc/init.d/slapd restart` i `/etc/init.d/slapd stop` per a aturar-lo (l'arrencada automàtica es po fer amb `update-rc.d slapd defaults`).

Com s'ha indicat en el punt anterior, hi ha diferents formes i eines per a accedir al directori LDAP i poder crear i modificar elements en aquest directori, entre les quals gq, phpldapadmin i JXplorer. En aquesta instal·lació pràctica utilitzarem un explorador de directoris LDAP (LDAP *browser*) com el phpldapadmin. Per a això, executem `apt-get install phpldapadmin`, reiniciem el servidor Apache `/etc/init.d/apache2 restart` i ens connectem per mitjà d'un navegador a `http://localhost/phpldapadmin/`. Hi apareixerà la pàgina de benvinguda i a la banda esquerra podrem fer l'inici de sessió (amb la contrasenya de l'LDAP). Una vegada connectats al servidor, seleccionem el root (dc=nteum, dc=org) i creem, entre les plantilles (*templates*) que hi apareixen, una *Organizational Unit (ou)*, a la qual anomenarem `users`. Repetim els passos (opció `create new child` des de el root) i creem una altra *ou* anomenada `groups`. Ara només ens queda crear els usuaris i els grups i assignar els usuaris als seus grups. Dins la unitat organitzativa `groups`, crearem els grups `vendes` (gid=1001) i `compres` (gid=1002) i en la unitat organitzativa `users` crearem els usuaris `joan pirulo` (uid=1001, vendes, id=jpirulo) i `anna pirulo` (uid=1002, compres, id=apirulo). Per a això, dins de `groups` fem `create new child` i seleccionem `Posix Group`. Creem els dos grups indicats, però haurem de modificar el `gid`, ja que els assigna per defecte a partir de 1.000 i nosaltres els volem diferents. Repetim l'operació en `users`, seleccionant `User Account`. Aquí ens demanarà les dades dels usuaris (nom, cognom, grup, contrasenyes, directori *home*, intèrpret d'ordres, etc.), i després haurem de canviar l'`uid`, ja que els assigna per defecte.

Ara veurem l'autenticació en un sistema LDAP (OpenLDAP), ja que una de les utilitats més importants d'un servidor LDAP és com a servidor d'autenticació, i veurem les modificacions que cal fer en un sistema Linux per a autenticar els usuaris en lloc d'utilitzar els arxius `/etc/passwd`, `/etc/group` i `/etc/shadow`. Per a això és necessari instal·lar i configurar els paquets libpam-ldap i libnss-ldap; el primer (pam-ldap) permet utilitzar el LDAP en les aplicacions que utilitzen PAM, com en el cas de Linux (l'arxiu de configuració és `/etc/pam_ldap.conf`). Per a especificar el mode d'autenticació de cada servei, és necessari configurar els arxius d'`/etc/pam.d/`. La segona biblioteca (nss-ldap) permet al servidor LDAP suplantar els arxius `/etc/passwd`, `/etc/group` i `/etc/shadow` com a bases de dades del sistema*. Finalment, haurem de configurar l'arxiu `/etc/nsswitch.conf` perquè s'utilitzi LDAP com a base de dades del sistema, en lloc dels arxius `passwd`, `group` i `shadow`. Per a instal·lar pam-ldap s'ha d'executar `apt-get install libpam-ldap` i contestar totes les preguntes (amfitrió i root de l'ldap, usuari admin, contrasenya, etc.) i es pot reconfigurar amb `dpkg-reconfigure libpam-ldap`. Les opcions es desen a `/etc/pam_ldap.conf`. En aquest arxiu hi ha alguns paràmetres que ja estan configurats, però hi hem d'afegir la resta i verificar què tenim (en funció del que ja hem posat en l'assistent de la instal·lació):

***L'arxiu de configuració és `/etc/libnss-ldap.conf` o `/etc/ldap.conf`, segons la versió.**

```
host 192.168.1.239 //nom o IP del servidor
base dc=nteum,dc=org
ldap_version 3
rootbinddn cn=admin,dc=nteum,dc=org
nss_base_passwd ou=users,dc=nteum,dc=org?one
nss_base_shadow ou=users,dc=nteum,dc=org?one
nss_base_group ou=groups,dc=nteum,dc=org?one
```

Per instal·lar la biblioteca libnss-ldap, executem l'ordre `apt-get install libnss-ldap` i s'iniciarà l'assistent de configuració d'aquesta llibreria (per a reconfigurar-la, `dpkg-reconfigure libnss-ldap`). L'assistent modificarà l'arxiu `/etc/libnss-ldap.conf`, que és on s'emmagatzema la configuració de la biblioteca i que posteriorment haurem d'editar manualment per a introduir algun canvi que no fa l'assistent. Finalment, haurem afegir a aquest arxiu (`/etc/libnss-ldap.conf`) les dues línies següents per indicar-li les unitats organitzatives:

```
nss_base_passwd ou=users,dc=nteum,dc=org?one
nss_base_group ou=groups,dc=nteum,dc=org?one
```

Perquè el servidor LDAP actuï com si es tractés dels arxius `passwd`, `group` i `shadow`, cal afegir a les línies que fan referència a `passwd`, `group` i `shadow` de l'arxiu `/etc/nsswitch.conf` la paraula '`ldap`' després de la paraula '`compat`' (o després de la paraula '`files`', depenent de com estigui configurat l'arxiu `/etc/nsswitch.conf`).

Per acabar, ens queda editar els arxius que hi ha a `/etc/pam.d` per a configurar l'autenticació de cadascun dels serveis. Per a no haver de configurar-los

independentment, tenim els arxius que comencen per `common`, que només els generals i els específics els inclouen (línia `@include`). Els arxius són:

- `/etc/pam.d/common-auth` (per a autenticar),
- `/etc/pam.d/common-account` (per a disposar d'un compte),
- `/etc/pam.d/common-session` (per a iniciar una sessió) i
- `/etc/pam.d/common-password` (per a canviar la contrasenya).

Tots aquests arxius contenen una línia que fa referència a `pam_unix.so` –la biblioteca que correspon a l'autenticació contra els arxius UNIX– i hem d'afegir les biblioteques `pam_ldap.so` per autenticar l'usuari (les afegirem a sobre, així autenticarà primer contra el servidor LDAP i després, si l'autenticació falla, provarà amb els arxius UNIX). Per a això, editeu els arxius següents tot afegint sobre la llibreria `pam_unix.so`:

- /etc/pam.d/common-auth: **auth sufficient pam_ldap.so**
- /etc/pam.d/common-account: **account sufficient pam_ldap.so**
- /etc/pam.d/common-session: **session sufficient pam_ldap.so**
- /etc/pam.d/common-password: **password sufficient pam_ldap.so**

Si volem que algun servei s'autentiqui de manera diferent, aleshores podem editar el fitxer del servei (per exemple `/etc/pam.d/su`, `/etc/pam.d/ssh`, `/etc/pam.d/ftp`, etc.), eliminar la línia que comença per `@include` i introduir la configuració específica que vulguem.

El servidor LDAP ja hauria d'autenticar correctament, cosa que es pot provar amb l'ordre `pamtest` del paquet `libpam-dotfile`. Per a instal·lar-la s'ha de fer `apt-get install libpam-dotfile` i podrem provar que el servei `passwd` (canviar contrasenya) funciona sobre un usuari del directori LDAP (per exemple, `anna`), executant `pamtest passwd apirulo`. Ens ha de permetre canviar la contrasenya. Executeu també l'ordre `finger` sobre usuaris que estiguin únicament en el directori LDAP, (per exemple, `joan`) `finger jpirulo`, i ens mostrarà tota la informació d'aquest usuari. Per exemple:

```
debian:# finger apirulo
Login: apirulo              Name: anna pirulo
Directory: /home/users/apirulo          Shell: /bin/sh
Last login Wed Nov 17 04:19 (EST) on tty6
No mail.
No Plan.

debian:# su - apirulo
apirulo@debian:~$
```

1.12. Serveis d'arxius (NFS, *Network File System*)

El sistema NFS permet que un servidor exporti un sistema d'arxiu perquè es pugui utilitzar interactivament des d'un client. El servei es compon d'un servidor `nfsd` i un client (`mountd`) que permeten compartir un sistema d'arxiu

(o part d'aquest) a través de la xarxa. A Debian, instal·leu `apt-get install nfs-common portmap` per al client, mentre que el servidor necessita `apt-get install nfs-kernel-server nfs-common portmap`. A Debian, el servidor es posa en marxa per mitjà dels *scripts* `nfscommon` i `nfs-kernel-server` a `/etc/init.d` (i els enllaços adequats a `/etc/rcX.d`). El servidor utilitza un arxiu (`/etc/exports`) per a gestionar l'accés remot als sistemes d'arxiu i el control d'aquests. Sobre el client (o un altre usuari per mitjà de `sudo`), el root pot muntar el sistema remot amb l'ordre:

```
mount Ipserver:directori-remot directori_local
```

i a partir d'aquest moment, el directori-remot es veurà dins de directori local (aquest ha d'existir abans d'executar el `mount`). Aquesta tasca en el client es pot automatitzar utilitzant l'arxiu de *mount* automàtic (`/etc/fstab`), incloent-hi una línia; per exemple:

```
pirulo.remix.com:/usr/local /pub nfs rsize=8192,wzise=8192,timeo=14.
```

Aquesta sentència ens indica que es muntarà el directori `/usr/local` de l'amfitrió `pirulo.remix.com` en el directori local `/pub`. A més, els dos paràmetres `rsize` i `wzise` són les mides de blocs de lectura i escriptura, `timeo` és el *timeout* d'RPC (si no s'especifiquen aquests tres valors, es prenen els valors per defecte). L'arxiu `/etc/exports` serveix d'ACL (llista de control d'accés) dels sistemes d'arxiu que es poden exportar als clients. Cada línia conté un sistema de fitxers (*filesystem*) per a exportar, seguit dels clients que el poden muntar, separats per espais en blanc. A cada client es pot associar un conjunt d'opcions per a modificar-ne el comportament (consulteu *man exports* per veure una llista detallada de les opcions). Un exemple d'això podria ser:

```
# Exemple d'/etc/exports
/           /master(rw) trusty(rw,no_root_squash)
/projects   proj*.local.domain(rw)
/usr        .local.domain(ro) @trusted(rw)
/pub        (ro,insecure,all_squash)
/home       195.12.32.2(rw,no_root_squash) www.first.com(ro)
/user       195.12.32.2/24(ro,insecure)
```

La primera línia exporta el sistema d'arxius sencer (`/`) a `master` i `trusty` en mode lectura/escriptura. A més, per a `trusty` no hi ha *uid squashing* (el root del client accedirà com a root als arxius root del servidor, és a dir, els dos root són equivalents malgrat ser de màquines diferents. S'indica per a màquines sense disc). La segona i tercera línies mostren exemples de '`*`' i de *netgroups* (indicats per `@`). La quarta línia exporta el directori `/pub` a qualsevol màquina del món, només de lectura, permet l'accés de clients NFS que no utilitzen un

port reservat per a l'NFS (opció `insecure`) i tot s'executa sota l'usuari nobody (opció `all squash`). La cinquena línia especifica un client per la seva IP i en la sisena s'especifica el mateix però amb màscara de xarxa (/24) i amb opcions entre parèntesis sense espai de separació. Només hi pot haver espais entre els clients habilitats. És important tenir en compte que hi ha tres versions d'NFS: V2, V3 i, recentment, V4). Les més comunes són V3 i, en algunes instal·lacions, V2. Si des d'un client V3 es connecta a un servidor V2, aquesta situació s'ha d'indicar amb un paràmetre.

1.13. Servidor de wiki

Un (o una) **wiki** (del hawaià *wiki wiki*, 'ràpid') és un lloc web col·laboratiu que poden editar diversos usuaris, els quals poden crear, editar, esborrar o modificar el contingut d'una pàgina web, de manera interactiva, fàcil i ràpida; aquestes facilitats fan d'un wiki una eina eficaç per a l'escriptura col·laborativa. La tecnologia wiki permet escriure de manera col·aborativa, i mitjançant un navegador, pàgines web allotjades en un servidor públic (les pàgines wiki); utilitzar una notació senzilla per a donar format, crear enllaços, etc., i conservar un historial de canvis que permet recuperar de manera senzilla qualsevol estat anterior de la pàgina. Quan algú edita una pàgina wiki, els seus canvis apareixen immediatament en el web, sense passar per cap tipus de revisió prèvia. Wiki també es pot referir a una col·lecció de pàgines hipertext, que qualsevol persona pot visitar i editar (definició de Wikipedia). Debian té la seva wiki a http://wiki.debian.org/ i Ubuntu a https://help.ubuntu.com/community/. Ambdues estan basades en **MoinMoin**, una *Python WikiClone* que permet inicialitzar ràpidament la seva pròpia wiki i només es necessiten un servidor de web i el llenguatge Python instal·lat. Al web de MoinMoin hi ha les instruccions detallades per a instal·lar MoinMoin, però hi ha dues maneres principals de fer-ho: instal·lació ràpida i instal·lació de servidor.

Enllaç d'interès

Per saber més coses sobre MoinMoin podeu visitar la seva pàgina web a: http://moinmo.in.
En concret, hi trobareu les ordres detallades per a install·llar MoinMoin a: http://master19.moinmo.in/InstallDocs.

1.13.1. Instal·lació ràpida

1) Descarregueu el paquet des d'http://moinmo.in/MoinMoinDownload, el qual serà, per exemple, per a la versió 1.9 `moin-1.9.0.tar.gz`. Si voleu verificar-ne la integritat, podeu fer `md5sum moin-x.x.x.tar.gz` i comprovar que coincideixin el *hash* generat i el que hi ha en la pàgina de descàrrega.

2) Desempaqueteu MoinMoin `tar xvzf moin-x.x.x.tar.gz`. Això crearà un directori `moin-x.x.x` en el directori actual, amb els arxius al seu interior.

3) Atès que MoinMoin està escrita en Python, cal utilitzar l'intèrpret de Python:

```
cd moin-x.x.x; python wikiserver.py
```

Aquesta ordre mostrarà en pantalla els missatges d'execució del servidor. Entre aquesta informació hi haurà l'adreça IP sobre la qual està corrent el servidor,

que pot ser del tipus `http://127.0.0.1:8080`. Aquesta opció utilitza un servidor web intern, és accessible des d'`http://localhost:8080/` i funcionarà fins que es pressioni `Ctrl-C` en el terminal.

1.13.2. Instal·lació de servidor

MoinMoin és una aplicació WSGI (*web server gateway interface*) i, per tant, el millor entorn per a executar Moin Moin n'és un que permeti WSGI com, per exemple, Apache amb mod_wsgi. Per això es recomana utilitzar aquest mètode en sistemes amb molta càrrega. Si tot s'ha configurat adequadament, es pot utilitzar test.wsgi per a provar que tot funciona correctament (és recomanable utilitzar el *mod_wsgi daemon model* i no l'*embedded model*, ja que és més segur).

Enllaç d'interès

Les instruccions d'instal·lació de WSGI per a Apache i de configuració de MoinMoin en aquest cas es poden trobar en l'adreça: http://moinmo.in/HowTo/ApacheWithModWSGI.

Instal·lació de MoinMoin

Per a instal·lar MoinMoin en un directori específic (que no és imprescindible, ja que es pot posar en un directori anomenat `MoinMoin/`), es pot utilitzar l'*script* `setup.py` o simplement descomprimir el paquet i després copiar els arxius. Si s'escull utilitzar `setup.py`, caldrà executar:

```
python setup.py install - -force - -record=install.log - -prefix='/usr/local' - -install-data=/srv
```

O, si es prefereix utilitzar el directori per defecte:

```
python setup.py install - -force - -record=install.log
```

Amb `- -install-data=/path` es pot canviar el directori a `/path` i l'opció `- -force` és important en la segona configuració per a eliminar qualsevol altra configuració.

Instal·lació d'una única wiki

Per a configurar MoinMoin, assumirem que el directori on l'hem instal·lat és `/moin/code` (reemplaceu-lo per l'adequat) i que el de la configuració és `/moin/config`.

1) El primer punt és configurar l'arxiu `moin.cgi` (o `moin.wsgi` si utiliza l'entorn WSGI) per a indicar a Python on són els arxius de MoinMoin (reemplaceu el `/path` indicat) o traieu el comentari en el codi: `sys.path.insert(0, '/moin/code')` `sys.path.insert(0, '/moin/config')`. Aquest arxiu és el que després haurà d'accedir al servidor web.

2) L'acció següent és configurar la wiki des del directori `/moin/config/`, cosa que es pot fer per a una sola wiki o per a un conjunt (*farmer*).

3) Copieu l'arxiu `wiki/config/wikiconfig.py` de la distribució en el directori `/moin/config/`.

4) Editeu aquest fitxer que, com veureu, està comentat per a facilitar-ne la configuració. És important que configureu bé els `/ paths` com a rutes absolutes i és essencial que configureu les directives següents: `data_dir` (on hi ha els arxius de la instal·lació), `data_underlay_dir` (MoinMoin ve amb tot un sistema de pàgines d'ajuda i aquesta directiva apunta on són aquestes pàgines), `interwikiname` (identificador per a la wiki), `sitename` (el nom de la wiki). Aneu amb compte, ja que `data_dir` conté informació sensible sobre la vostra wiki que ningú no ha llegir, excepte el codi de la wiki. Per això no heu de permetre que aquest directori sigui accessible a través del servidor web i tampoc no heu de copiar-lo al directori root del vostre servidor web.

5) Si no l'heu instal·lat, haureu de copiar el contingut de `wiki/data/` de la distribució en el directori especificat a `data_dir` i haureu de fer el mateix amb `wiki/underlay/` en el directori indicat a `data_underlay_dir`.

Instal·lació de múltiples wikis

Primer heu de copiar `wiki/config/wikifarm/*` de la distribució al directori `/moin/config/`. Després s'han de seguir les instruccions anteriors per a cadascuna de les wikis de la col·lecció (*farm*), tenint en compte que:

1) és necessari tenir `data_dir` i `data_underlay_dir` separats per a cada wiki;

2) si voleu que comparteixin alguna configuració, llavors aquesta ha d'estar a `farmconfig.py` i les específiques han d'estar a `mywiki.py`.

Instal·lació de MoinMoin sobre Ubuntu

En aquest subapartat instal·larem MoinMoin (versió 1.9) sobre Ubuntu, utilitzant el servidor Apache amb el mòdul WSGI, que és molt més eficient que el CGI. Per a això, comencem instal·lant `sudo apt-get install apache2 libapache2-mod-wsgi`. Per utilitzar l'última versió de MoinMoin, no installarem el paquet des del dipòsit d'Ubuntu, sinó que explicarem com s'ha de fer manualment. Baixem des de l'adreça http://moinmo.in/ l'última versió (en aquest cas, `moin-1.9.0.tar.gz`) i fem `tar xvzf moin-1.9.0.tar.gz` des d'un terminal. Després passem al directori `cd moin-1.9.0` i executem:

```
sudo python setup.py install - force --prefix /usr/local --record= install.log
```

Podem executar simplement l'ordre `cd /usr/local/share/moin/server` per fer una prova senzilla, i fer `sudo python test.wsgi`, que ens donarà un missatge com aquest: `Running test application - point your browser at http://localhost:8000/ ...` A continuació, podem escriure al navegador `http://localhost:8000/` i tindrem la pàgina de prova WSCGI amb informació del sistema (`Crtl-C` per a acabar l'execució del servidor).

A continuació hem de configurar la wiki:

1) Per copiar els fitxers de configuració, executem primer la següent ordre: `cd /usr/local/share/moin`; després `sudo cp server/moin.wsgi` i finalment `sudo cp config/wikiconfig.py`.

2) A continuació, hem de configurar el servidor Apache (executem `sudo gedit /etc/apache2/apache2.conf`) i afegir -hi al final (i desat l'arxiu):

```
# Configuració MoinMoin WSGI
# Invoqueu la moin wiki a l'url http://servername/FrontPage:
WSGIScriptAlias / /usr/local/share/moin/moin.wsgi

# Creeu els dimonis wsgi daemons i utilitzeu user/group igual que el data_dir:
WSGIDaemonProcess moin user=www-data group=www-data processes=5 threads=10 maximum-
requests=1000 umask=0007

# Utilizeu els dimonis definits en la línia anterior
WSGIProcessGroup moin
```

3) Per configurar WSGI, primer hem d'executar la següent ordre: `sudo gedit /usr/local/share/moin/moin.wsgi`. Després afegim al final de la secció *a2* la línia `sys.path.insert(0, '/usr/local/share/ moin')`.

4) Afegim seguretat a l'entorn amb les quatre ordres següents:

```
cd /usr/local/share;
sudo chown -R www-data:www-data moin;
sudo chmod -R ug+rwX moin;
sudo chmod -R o-rwx moin,
```

la qual cosa permetrà que només el *Web server service user* (www-data) pugui modificar-les.

5) Després, apliquem els canvis i reiniciem Apache amb la següent ordre: `sudo /etc/init.d/apache2 restart`. Tindrem la wiki activa a l'adreça `http://localhost/` amb la pàgina inicial.

6) Finalment, hi apliquem unes configuracions addicionals mínimes: `sudo gedit /usr/local/share/moin/wikiconfig.py`; traiem el comentari de `page_front_page = u"FrontPage"` i indiquem el nom de l'administrador, per exemple, `superuser = [u"WikiAdmin", ]`. Finalment, executem l'ordre `sudo /etc/init.d/apache2 restart`. Per configurar el llenguatge, entrem com a administrador (WikiAdmin)* i després anem a l'adreça següent: `http://localhost/LanguageSetup?action=language_setup`.

*** Si no tenim un usuari, seleccionem inici de sessió i el creem.**

1.14. Gestió de còpies de seguretat (*backups*)

Hi ha diverses opcions per a fer còpies de seguretat d'un conjunt d'ordinadors. Una de les més potents és **Bacula***, que és una col·lecció d'eines per a fer còpies de seguretat en una xarxa. Bacula es basa en una arquitectura client-servidor que resulta molt eficaç i fàcil d'utilitzar, ja que presenta un conjunt molt ampli de característiques i és eficient per a un conjunt d'ordinadors personals i també per a grans instal·lacions. El paquet està format per diferents components, i alguns dels més importants són:

***http://www.bacula.org**

- **Bacula-director**, dimoni que gestiona la lògica dels procediments de seguretat;
- **Bacula-storage**, dimoni que s'encarrega de gestionar els dispositius d'emmagatzematge;
- **Bacula-file**, dimoni per mitjà del qual Bacula obté els fitxers que necessita per a fer la còpia de seguretat i que caldrà instal·lar en les màquines font dels fitxers que s'han de protegir, i
- **Bacula-console**, que permet interactuar amb el servei de seguretat.

Bacula suporta discos durs, cintes, DVD, USB i també diferents bases de dades (MySQL, PostgreSQL i SQLite). Com a contrapartida, però, cal instal·lar tots els paquets, i la instal·lació i la posada a punt poden ser complexes.

Un altre paquet interessant és **BackupPC***, que permet fer còpies de seguretat de disc a disc amb una interfície basada en la web. El servidor s'executa en qualsevol sistema Linux i admet diferents protocols perquè els clients puguin escollir la manera de connectar-se al servidor. Aquest programa no és adequat com a sistema de còpia de seguretat d'imatges de disc o particions, ja que no suporta còpies de seguretat pel que fa al bloc de disc, però és molt simple de configurar i la possible intrusió en la xarxa d'ordinadors en què es vol fer la còpia és mínima. Aquest servidor incorpora un client *server message block* (SMB) que es pot utilitzar per a fer la còpia de seguretat de recursos compartits de xarxa d'equips que executen Windows.

***http://backuppc.sourceforge.net/info.html**

Instal·lació del servidor

Per a instal·lar el servidor [3], primer executeu `apt-get install backuppc smbfs libfile-rsyncp-perl`, que també hi instal·larà altres paquets necessaris de Perl i preguntarà quin servidor d'Apache hi teniu instal·lat i si voleu configurar-lo, a més del grup d'SMB. La instal·lació generarà una contrasenya per a l'accés web*. Si voleu canviar la contrasenya, podeu fer-ho amb `htpasswd /etc/backuppc/htpasswd backuppc`. A continuació, afegiu a l'arxiu `/etc/samba/smb.conf` l'opció `unix charset = ISO8859-1` i traieu el comentari del paràmetre `'- -checksum-seed=32761'`, de les op-

***Recordeu anotar bé l'usuari i la contrasenya, que en el nostre cas és `backupPC` i `Bg0BRd90` com a contrasenya per a la instal·lació de prova.**

cions `RsyncArgs` i `RsyncRestoreArgs` del fitxer principal de configuració, `/etc/backuppc/config.pl`. Amb això ja podeu accedir a la pàgina web del servidor (`http://localhost/backuppc`) i us demanarà l'usuari i la contrasenya indicats anteriorment.

Client en les estacions de treball

Les còpies de seguretat de les estacions Windows es faran (d'acord amb les indicacions de la documentació de Backuppc) amb les `rsyncd`. Per a això cal instal·lar el paquet `rsyncd`, que es pot descarregar de la mateixa pàgina de projecte, i seguir les indicacions de `README.txt`. Es recomana utilitzar els mateixos nom d'usuari i contrasenya per als `rsyncd`, a més de configurar el fitxer `c:\rsyncd\rsyncd.secrets` per a cada estació de treball i en les opcions `RsyncdPasswd` i `RsyncdUserName`, en el fitxer de configuració del servidor. A més, per a aplicar les còpies de seguretat caldrà configurar el fitxer `c:\rsyncd\rsyncd.conf` per a cada estació, i instal·lar el servidor rsync. Per a les estacions de treball Linux també s'utilitzarà `rsyncd` (una altra opció és utilitzar `rsync` sobre `ssh`).

Configuració addicional del servidor

En el servidor s'ha d'afegir una línia a `/etc/backuppc/hosts` i un fitxer de configuració per a cada estació de treball en el directori `/etc/backuppc`). Per exemple, per a l'anterior configuració (el nostre amfitrió de prova es diu **nteum**), els fitxers resultants són:

```
# File: /etc/backuppc/hosts
# ...
#
host dhcp user moreUsers # <— do not edit this line
#farside 0 craig jill,jeff # <— example static IP host entry
#larson 1 bill # <— example DHCP host entry
localhost 0 backuppc
nteum 0 nteum_user

# File: nteum.pl
#
$ConfXferMethod = 'rsyncd';
$ConfRsyncShareName = ['docs','soft','tmp','data'];
```

Estacions Linux

Per a les estacions Linux només cal instal·lar el paquet `rsync` en el client i crear els fitxers `/etc/rsyncd.conf` i `/etc/rsyncd.secrets`, com per exemple:

```
# File: rsyncd.conf
# rsyncd per al host debianSYS
#


pid file=/var/run/rsyncd.pid
lock file = /var/lock/rsyncd
read only = yes
list = false
auth users = bkpuser
```

```
strict modes = true
secrets file = /etc/rsyncd.secrets
hosts allow = 192.168.1.1


[root]
comment = home de root
path = /root


[home]
comment = homes
path = /home


[etc]
comment = configuracions
path = /etc


[var]
comment = dades de programes
path = /var



# File: rsyncd.secrets
# The format of this file is user:password. You can have as many entries
# as you wish. These accounts are sepecifc to the rsync daemon and share
# no relation to Windows local/domain accounts, nor Cywin entries in the
# passwd file.
#
# SECURITY WARNING: Don't use these defaults of UUU for the user name
# and PPP for the password! Change them!!
#
bkpuser:xxxxxxxxx
```

La configuració addicional per al servidor consistiria, per una banda, a afegir una línia al fitxer `/etc/backuppc/hosts` i, per l'altra, a crear el fitxer `/etc/backuppc/debianSYS.pl`:

```
# File debianSYS.pl
$ConfXferMethod = 'rsyncd';
$ConfRsyncShareName = ['etc','home','var','root'];
```

És important que el servidor pugui trobar el nom de la màquina i és per això que es recomana posar una entrada `ip host` a `/etc/hosts`. Finalment, s'ha d'executar el dimoni `rsync` en el client i per a això hi ha dues opcions: autònomament o mitjançant `inetd`. En aquest cas escollim la primera opció i cal canviar la configuració d'`/etc/default/rsync` perquè el dimoni estigui actiu per defecte en executar `dpkg-reconfigure rsync` (cal reconfigurar el servei).

Creació d'arxius

Si bé aquest servei crearà còpies de seguretat, només és un servidor i en entorns crítics és aconsellable fer còpies de les còpies externes al servidor, en suport DVD, USB, etc. BackupPC proporciona un mecanisme per a generar arxius a la carta i el seu mecanisme és similar a configurar un nou amfitrió, però indicant

com a mètode la paraula `archive`. Per exemple, en el nostre cas tenim el fitxer `/etc/backuppc/archive.pl` per a preparar còpies de seguretat externes en DVD:

```
# File: etc/backuppc/archive.pl
#
# arxiu de còpies de seguretat extern (en DVD)
#
$ConfXferMethod = 'archive';
```

Com podeu observar, és simple, ja que només s'han de modificar les opcions generals de l'arxiu que hi ha a `/etc/backuppc/config.pl`, secció `Archive` (solament es mostren les parts més interessants):

```
# ...
# Archive Destination. The Destination of the archive
# e.g. /tmp for file archive or /dev/nst0 for device archive
$ConfArchiveDest = '/var/lib/backuppc/archives';


# Archive Compression type
$ConfArchiveComp = 'gzip';


# Archive Parity Files
$ConfArchivePar = 0;


# Archive Size Split
$ConfArchiveSplit = 4500;


# Archive Command
$ConfArchiveClientCmd = '$Installdir/bin/BackupPC_archiveHost'
. ' $tarCreatePath $splitpath $parpath $host $backupnumber'
. ' $compression $compext $splitsize $archiveloc $parfile *';
```

Activitats

1. Configureu un servidor DNS com a memòria cau i amb un domini propi.

2. Configureu un servidor/client NIS amb dues màquines i exporteu els directoris d'usuari del servidor per NFS.

3. Configureu un servidor SSH per a accedir des d'una altra màquina sense contrasenya.

4. Configureu un servidor Apache+ SSL+ PHP+ MySQL+ PHPAdmin per a visualitzar els fulls personals dels usuaris.

5. Creeu i configureu un sistema de correu electrònic amb Exim, Fetchmail, SpamAssassin i un servidor IMAP per a rebre correus des de l'exterior i poder llegir-los des d'una màquina remota amb el client Mozilla (Thunderbird).

6. Instal·leu la Wiki MoinMoin i creeu un conjunt de pàgines per a verificar-ne el funcionament.

7. Instal·leu el servidor de còpies de seguretat BackupPC i genereu una còpia de seguretat des d'una màquina Windows i una altra des d'una màquina Linux. Trieu el mètode de comunicació amb els clients i justifiqueu la resposta.

Bibliografia

[1] *Apache2 + SSL*.
<http://www.debian-administration.org/articles/349>

[2] *Apache2 + WebDav*.
<http://www.debian-administration.org/articles/285>

[3] **Baila, S.** (2005). *Instalación de BackupPC*.
<http://fitxers.sargue.net/fitxers/diarioBackupPC.html>

[4] *Comparison of FTP client software*
<http://en.wikipedia.org/wiki/Comparison_of_FTP_client_software>

[5] **Debian.org**. *Debian*.
<http://www.debian.org>

[6] *Exim*.
<http://www.exim.org/docs.html>

[7] **IETF**. Dipòsit de Request For Comment desenvolupats per Internet Engineering Task Force (IETF) al Network Information Center (NIC). <http://www.ietf.org/rfc.html>

[8] **Instituto de Tecnologías Educativas**. *Redes de área local: Aplicaciones y Servicios Linux*.
<http://www.isftic.mepsyd.es/formacion/materiales/85/cd/REDES_LINUX/index.htm>

[9] **Kiracofe, D.** *Transparent Proxy with Linux and Squid mini-HOWTO*.
<http://tldp.org/HOWTO/TransparentProxy-1.html>

[10] **Kukuk, T.** *The Linux NIS(YP)/NYS/NIS+ HOWTO*.
<http://tldp.org/HOWTO/NIS-HOWTO/verification.html>

[11] **Langfeldt, N.** *DNS HOWTO*.
<http://tldp.org/HOWTO/DNS-HOWTO-7.html>

[12] *LWP*.
<http://www.lawebdelprogramador.com/temas/tema_stablephpapachemysql.php>

[13] *MoinMoin*.
<http://moinmo.in/>

[14] *MoinMoin + Ubuntu*.
<http://moinmo.in/HowTo/UbuntuQuick>

[15] **Mourani, G.** (2001). *Securing and Optimizing Linux: The Ultimate Solution*. Open Network Architecture, Inc.

[16] *Mutt*.
<http://www.mutt.org>

[17] *NIS HOWTO*.
<http://www.linux-nis.org/doc/nis.debian.howto>

[18] **Pinheiro Malère, L. E.** (2007). *Ldap*. *The Linux Documentation Project*.
<http://tldp.org/HOWTO/LDAP-HOWTO/>

[19] *ProcMail*.
<http://www.debian-administration.org/articles/242>

[20] *Proftpd*.
<http://www.debian-administration.org/articles/228>

[21] *PureFtpd*.
<http://www.debian-administration.org/articles/383>

[22] *Squid proxy server, Proxy Cache*.
<http://www.squid-cache.org/>