

CONFIGURACIÓ D'UN SERVIDOR GNU/LINUX

Versió reduïda de l'original

Nom: Daniel Clemente Laboreo
Tutor: Joan Canudas
Curs: 2n BAT. C
Centre: IES Bruguers (Gavà)
Data: Gener 2003
<http://www.danielclemente.com/servidor/>

Índex

1 INTRODUCCIÓ	1
1.1 Context	1
1.2 Justificació.....	2
1.3 Objectius	3
2 CONFIGURACIÓ DEL SERVIDOR.....	4
2.1 Abans de començar	4
2.1.1 Mètode a seguir	4
2.1.2 Sobre l'ordinador	5
2.2 Instal·lació de Linux	6
2.2.1 Procés d'instal·lació.....	6
2.2.2 Configuracions bàsiques	7
2.3 Serveis bàsics	9
2.3.1 Integració amb la xarxa de Windows.....	9
2.3.2 Servidor web per a l'Intranet	13
2.3.3 Servidor d'FTP	16
2.3.4 Configuració del proxy.....	19
2.4 Serveis secundaris	27
2.4.1 Suport de PHP a la web.....	27
2.4.2 Suport de CGIs a la web.....	30
2.4.3 Estadístiques d'accés a la web.....	32
2.4.4 Connexions per Secure Shell.....	34
2.4.5 Web per a Internet	36
2.4.6 Servidor DNS	40
2.4.7 Firewall.....	45
3 CONCLUSIONS	49
4 ANNEXOS	50
4.1 Comandes importants	50
4.2 Seguretat	53
4.3 Manteniment	56
4.4 Altres utilitats	57
4.5 Glossari de termes	58
4.6 Llicència FDL de la GNU	63

1 INTRODUCCIÓ

1.1 Context

Sóc un alumne de l'institut IES Bruguers, de Gavà (Barcelona), usuari habitual de Linux, programador i webmaster. El meu Treball de Recerca consistirà en el reciclatge d'un ordinador vell d'una de les aules d'informàtica per convertir-ho en un ordinador central de l'institut que ofereixi diferents serveis a alumnes i professors; i només utilitzant només Linux i altre software lliure de codi obert.

L'ordinador posarà a l'abast dels alumnes i visitants una pàgina web amb continguts dinàmics, diferents formes de compartir arxius entre els departaments, servei de proxy per accelerar la navegació, i altres característiques avançades com els dominis DNS, les connexions SSH a l'ordinador, el filtrat de paquets, el PHP i els CGIs.

Tot això es farà sense haver de comprar programes professionals, ja que tot el software utilitzat és gratuït i legal. A més, aportarà a l'institut tecnologies molt més professionals en utilitzar un sistema operatiu del nivell de Linux.

El treball està orientat a lectors que ja tinguin una certa experiència en qualsevol distribució de Linux utilitzant la consola de comandes, o, al menys, que tinguin interès per aprendre (això és el més important). No s'explicaran les ordres bàsiques ja que hi ha molts tutorials més apropiats per tot Internet.

També suposarem que el lector entén una mica d'anglès (l'anglès relacionat amb la informàtica), perquè així els conceptes quedaran més clars.

1.2 Justificació

He escollit aquest tema per molts motius:

En primer lloc, vull donar a conèixer Linux (l'únic sistema operatiu que utilitzo) i treure la por a la gent que pensa que 'només és per a experts'. Aquest treball no servirà per demostrar que és molt fàcil d'usar, perquè precisament utilitzarem una de les distribucions més complicades, la Debian. Mandrake o Knoppix són més adequades per als principiants, però no per a la finalitat d'aquest treball.

També he fet el servidor perquè he vist que hi feia falta un ordinador central en una xarxa cada vegada més gran; feia falta un lloc comú que servís per compartir arxius entre els diferents departaments, i per simplificar les tasques de manteniment (instal·lació d'un nou programa a cada màquina, actualització dels antivirus, etc). I a més, algunes instal·lacions ho requerien (per exemple, per assolir la velocitat màxima de la línia ADSL s'havia de muntar un proxy).

A més, crec que posar un sistema operatiu com Linux, per al qual hi existeixen milers d'aplicacions professionals des de fa molts anys, elevarà la qualitat de les instal·lacions informàtiques i donarà moltes més possibilitats als alumnes i professors. L'ordinador serà del mateix nivell que els que es poden trobar a les universitats d'informàtica.

És clar que el treball també servirà per contribuir a la documentació de Linux en català, però aquesta vegada amb un cas pràctic fet per apartats, i amb les opinions, problemes, idees i errors que han aparegut durant la seva realització.

Finalment, aquest treball també l'he fet amb la intenció d'ajudar els futurs informàtics que estan aprenent i que volen conèixer coses noves. Ho faig perquè a mi

m'hauria agradat assabentar-me abans de l'existència de Linux.

1.3 Objectius

Què es vol aconseguir amb el treball? No és només muntar un ordinador que faci de tot, sinó que...:

- es vol millorar la qualitat de les instal·lacions del centre, utilitzant programes professionals.
- es vol promoure l'ús de Linux i el software lliure, seguint l'exemple de Debian (la distribució menys comercial de totes)
- es volen aprofitar els ordinadors vells que no funcionen amb altres sistemes operatius
- i, és clar, es vol muntar un ordinador amb les següents característiques:
- que sigui fàcil d'usar
- amb servidor web, un per a Internet i un altre per a la xarxa interna
- que el servidor web tingui suport per a scripts CGIs (comptadors, per exemple) i llenguatge PHP
- que també mostri gràficament les estadístiques d'accés a la pàgina
- amb servidor DNS per no haver de recordar IPs
- amb servidor d'FTP per a posar arxius
- que estigui integrat a la xarxa de Windows i que pugui compartir carpetes
- que accepti connexions remotes per administrar-ho des d'altre lloc
- que estigui protegit contra els alumnes entremaliats.

2 CONFIGURACIÓ DEL SERVIDOR

2.1 Abans de començar

2.1.1 Mètode a seguir

El treball pràctic està dividit en tres blocs: instal·lació, serveis bàsics, i serveis secundaris

- **Instal·lació** : instal·larem des de zero el Linux amb el mínim d'opcions.

Després començarem a canviar algun paràmetre i preparar el sistema per la part difícil.

- **Serveis bàsics** : intentarem solucionar les necessitats de la xarxa.

Integrarem l'ordinador a la xarxa de Windows, farem que sigui un servidor web (per a la xarxa interna) amb Apache, posarem un servidor d'FTP i farem que faci de proxy per poder controlar les connexions i a més accelerar la navegació.

- **Serveis secundaris** : ampliarem els de la primera part i n'afegirem d'altres. Per exemple, farem que el servidor web accepti connexions tant de la xarxa interna com d'Internet, i les tracti de forma diferent. També el millorarem donant-li suport per a PHP i per CGIs, i pujarem una pàgina on podrem veure estadístiques sobre les visites al servidor (pàgines més demanades, quantitat de visites per dia, etc). Farem que el servidor accepti connexions remotes amb la finalitat de poder administrar el PC des de qualsevol lloc tal com es faria si s'estigués davant. Altres punts molt importants d'aquesta part són la configuració del servei DNS (traductor de noms de domini a les IPs adequades) i el firewall (que

protegirà el servidor dels curiosos).

També pensarem en el manteniment de l'ordinador una vegada acabada tota la instal·lació, i en com actuar quan es comenci a quedar antiquat o surtin problemes.

2.1.2 Sobre l'ordinador

La xarxa de l'institut Bruguers consta de dues aules, una a cada pis, més els ordinadors dels diferents departaments que hi ha a ambdós pisos. El servidor es muntarà a una de les dues aules d'informàtica, junt amb un altre que té Windows NT 4.0 i que controla la xarxa.

Aquesta xarxa és de classe C (255 ordinadors com a màxim), i la seva direcció IP és 192.168.0.0/24 (o sigui, que la màscara de subxarxa és 255.255.255.0). Apart d'ordinadors bastant diferents entre ells, també hi ha un JetDirect i una fotocopiadora connectats. L'accés a Internet es fa mitjançant una línia ADSL de 2 Mbits/s controlada per un router Cisco del qual no podem modificar la configuració ja que pertany a la xtec (Xarxa Telemàtica Educativa de Catalunya). Per tant, per fer un servidor web on es pugui entrar des d'Internet s'haurà de demanar permís a aquesta entitat.

No necessitarem comprar cap superordinador: a diferència dels de les grans empreses, el nostre servidor no sempre estarà utilitzant -se; i quan hagi de respondre alguna connexió, servir pàgines web o executar processos dels usuaris connectats difícilment arribarà al 100% d'ús. A més, Linux -a diferència d'altres sistemes operatius molt utilitzats- aprofita perfectament els recursos de l'ordinador, podent executar-se fins i tot en 386 amb 4 Mb de RAM i menys de 100 Mb. de disc dur.

Al nostre centre disposem d'un ordinador Pentium I a 100 Mhz, 48 Mb de RAM, 1'2 Gb de disc dur, CD-ROM, disquetera, pantalla a color, teclat, ratolí i targeta de xarxa Realtek 8139. Aquest ordinador té tot el que necessitem i més per a funcionar com a servidor d'un institut. No utilitzarem el CD-ROM ni el ratolí (encara que es pot fet), ja

que amb la xarxa, teclat i pantalla ja es pot fer tot.



L'ordinador assignat

2.2 Instal·lació de Linux

2.2.1 Procés d'instal·lació

Com que Linux és software lliure, s'ha desenvolupat seguint diverses tendències simultànies. A aquestes versions les anomenem distribucions, i n'hi ha milers, cadascuna dedicada a una necessitat en concret. Per al nostre servidor utilitzarem Debian, que és la més representativa del software lliure, la que menys errors dóna, i també la que té fama de ser més complicada d'usar (tot i que veurem que no ho és). Al mateix temps, Debian desenvolupa tres branques del seu sistema operatiu:

- **Estable (“stable”)**: la més recomanada. Està molt provada i teòricament no hauria de fallar res.
- **Inestable (“unstable”)**: la que van millorant els programadors cada dia. Quan passa un temps, la proven a fons (es converteix en “en proves”) fins que arriba a ser “stable”.
- **En proves (“testing”)**: la versió inestable bloquejada, que retoquen i perfeccionen. Quan, després d'uns mesos, veuen que pot sortir al públic, la converteixen en “stable”.

Por no arriscar-nos, utilitzarem la versió estable. La seva instal·lació no és difícil. No l'explicarem completament perquè hi ha millors documents a Internet, per exemple a <http://www.debian.org/releases/stable/i386/install.ca.html> (en català) o a http://www.osnews.com/story.php?news_id=2016 (en anglès). Al nostre cas hem fet una instal·lació per Internet: s'han de baixar una imatges de disquet, gravar-les, arrencar l'ordinador amb elles, crear particions (amb una swap de 50 mb.), afegir el mòdul de la

targeta de xarxa ([rtl8139](#)), posar [bruguers](#) com a nom de host, i 192.168.0.200 com a IP. També s'ha activat la opció de contrasenyes shadow, i s'han seleccionat les opcions mínimes per al sistema (no hi ha mode gràfic, ni jocs ni cap servidor; tot ho anirem posant nosaltres).

2.2.2 Configuracions bàsiques

Ara que tenim el sistema operatiu instal·lat hem de fer unes configuracions senzilles, que s'han de fer abans de començar a posar els programes i dimonis. Primer de tot ens identificarem com a root (amb la contrasenya que vam posar a la instal·lació).

El què farem serà:

- **Configuració d'apt**: apt és el sistema de control de paquets exclusiu de Debian. Executarem `apt-setup` per configurar-ho, i després serà molt fàcil instal·lar els programes.
- **Actualització de paquets**: una vegada definits els servidors d'apt, farem `apt-get update` per actualitzar la informació sobre els nous paquets amb el servidor. No s'instal·larà res encara; només es sincronitzen.
Per actualitzar els paquets instal·lats amb les noves versions disponibles hem de fer `apt-get upgrade -u` (el -u és per mostrar-ne els noms). Després de mostrar la llista de paquets a baixar, responem (Y/n) i només caldrà esperar a que acabi. Potser mentre els instal·la apareix alguna pregunta, però normalment dóna suficient informació per saber què respondre.
Si volem algun paquet addicional només hem de fer `apt-get install nom`.
Podem instal·lar programes en qualsevol moment, a mesura que els necessitem. A l'annex hi ha una llista de programes recomanats. També és recomanable instal·lar ara `gpm` per poder utilitzar el ratolí a la consola (es configura amb `gpmconfig`).

- **Locals** : podem dir-li a Debian que preferim els programes i l'ajuda en espanyol fent un `dpkg-reconfigure locales` i seleccionant `es_ES` i `es_ES@euro`. Així podrem usar accents i altres símbols especials com el de l'euro, a més de veure la majoria de missatges i programes en espanyol i no en anglès.
- **Hora del sistema** : com que hi ha accions que depenen del temps, posarem la data i hora correctes amb comandes com `date -s "25/12/2003"` o `date -s "13:15:42"`
- **Tasques programades** : hi ha accions que s'executen cada dia, cada setmana o cada mes; el problema és que potser l'ordinador no estarà engegat a les hores que hi ha posades per defecte (6:25, 6:47 i 6:52). Només caldrà canviar a `/etc/crontab` el segon número de cada línia -que representa l'hora d'una acció programada- per un de més adient que les 6 hores del matí; per exemple, les 10 del matí.
- **Protecció d'alguns arxius importants** : hi ha arxius que un usuari normal no ha de tenir la necessitat de veure . Per això traurem els permisos de lectura, escriptura o execució a cadascun d'ells, utilitzant el comand `chmod`.
`chmod o-rwx /etc/passwd /etc/exports /etc/*netd.conf`
(`/etc/passwd` té informació sobre els usuaris del sistema i els seus privilegis, `/etc/exports` diu quins directoris es poden accedir remotament, i `/etc/inetd.conf` o `xinetd.conf` diu quins serveis es carreguen cada

vegada que s'encén el PC).

- **Instal·lació d'un nou kernel, si n'hi ha**: és recomanable; farà falta la versió 2.4 per posar el tallafocs. És fàcil d'instal·lar posant `apt-get install kernel-image-2.4.18-686` (per exemple) i seguint les instruccions.

2.3 Serveis bàsics

2.3.1 Integració amb la xarxa de Windows

Com que a l'institut els ordinadors estan connectats mitjançant una xarxa i un servidor Windows NT, hi inclourem també el nostre per poder compartir carpetes o impressores, o accedir a les dels altres ordinadors; tot utilitzant el paquet Samba.

Com sempre, fem un `apt-get install samba` i diem que sí que volem que faci unes preguntes per adaptar el fitxer de configuració `smb.conf`, encara que després el revisarem. Preguntarà el grup de treball (al nostre cas és `99PIENT22.DOM`), i si volem utilitzar contrasenyes xifrades (imprescindible per evitar errors). En acabat ja tenim els dos processos de Samba funcionant: `smbd` i `nmbd` (fan referència als dimonis de SMB i Netbios, respectivament). Si volem que Debian ens torni a preguntar tot l'anterior podem fer un `dpkg-reconfigure samba`

Amb Samba podem fer quatre operacions concretes: compartir unitats o impressores amb màquines Windows o Linux. Hem de tenir molt clar què volem que faci el servidor, i actuar en conseqüència. Si posem configuracions per defecte o canviem coses sense saber què son, no funcionarà exactament com volem, i arreglar-ho pot costar molt.

En el nostre cas, les decisions que hem pres són: compartir només una carpeta del servidor (amb els seus directoris) a tots els usuaris de Windows i deixar accedir als recursos compartits de qualsevol ordinador des del servidor. No tenim cap impressora connectada al servidor, per tant, Linux no n'ha de compartir cap. El que sí

que podem fer és que accedeixi a una impressora de Windows; però s'hauria de configurar amb els drivers apropiats.

El primer de tot serà crear un usuari al servidor que sigui l'únic que pugui tenir accés als recursos compartits. És l'usuari que utilitzaran tots els ordinadors que hi accedeixin, i, tal com hem dit abans, cal que aquest usuari estigui creat a tots els Windows i tingui la mateixa contrasenya. Podem fer una excepció amb els Windows NT ja que, com hem dit abans, permeten escriure un nom i contrasenya en accedir-hi, independentment del nom de l'usuari connectat. Nota: no cal que anem ordinador per ordinador creant aquest usuari, ja que es crearà automàticament quan algú posi el seu nom i contrasenya a la pantalla d'inici de sessió.

Al nostre cas aquest usuari l'anomenarem `bruguers` i li donarem una contrasenya pública i fàcil de memoritzar (però no d'endevinar!). La contrasenya és opcional, però el no posar-ne representa un forat de seguretat molt gran (sobretot quan estem parlant de sistemes Microsoft),

Afegim l'usuari al servidor amb `adduser bruguers` (no és necessari escriure cap dada més), i l'afegim a la llista d'usuaris de Samba -que es troba a `/etc/samba/smbpasswd-` amb el comand `smbpasswd -a bruguers`, i posant la mateixa contrasenya.

Ara hem d'editar el fitxer de configuració `/etc/samba/smb.conf`. A més de comprovar la configuració actual, hem d'afegir les següents línies per evitar un dels mètodes d'intrusió més utilitzats des de fa molts anys: el dels recursos compartits.

```
hosts allow = 192.168.0. localhost
valid users = bruguers
```

A la secció titulada `Share Definitions` posarem aquest fragment:

```
[homes]
comment = Home Directories
browseable = yes
writable = yes
create mask = 0777
directory mask = 0777
```

Això fa que es comparteixin les carpetes personals dels usuaris (`[homes]`). En el nostre cas, només n'hi ha un, d'usuari, que s'anomena `bruguers`. Aquest té permís per a escriure al seu directori (`writable=yes`) i tot el que crei podrà ser llegit, modificat o executat per qualsevol usuari del sistema (per això posem els permisos a `777`, o, el que és el mateix, `u=rwx g=rwx o=rwx`). El `browseable=yes` fa que el recurs es pugui veure en navegar pels recursos compartits del servidor.

El següent paràgraf, que comença amb `[printers]`, el comentarem sencer amb un caràcter `#` a cada línia, perquè no tenim impressores per compartir connectades al servidor.

Reiniciem el servei amb `/etc/init.d/samba restart` i provem d'accedir-hi des d'un ordinador Windows explorant la xarxa (a '`Entorno de red`' o '`Mis sitios de red`'). Veurem un ordinador anomenat '`Bruguers`', i a dins, la carpeta compartida, on podem entrar i desar-hi coses. També veurem la icona per gestionar les impressores.

Probablement no funcioni a la primera degut a les diferències entre cada Windows. Els problemes que hi ha entre Samba (Linux) i Windows són els mateixos que els que Windows 98 i Windows NT s'ocasionen mútuament. Bàsicament, es resumeixen en: “es pot accedir de NT a 98 però de 98 a NT no”. A continuació en presentem els més típics, junt amb algunes possibles solucions:

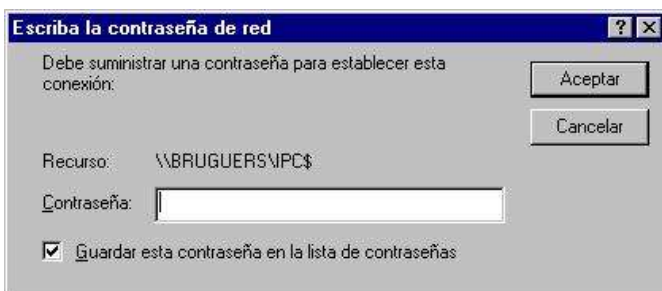
- A Windows NT/2000, no deixa entrar al servidor ('Acceso denegado').

Probablement estigui desactivada la opció d'utilitzar contrasenyes encriptades. Cal afegir `encrypt passwords = yes` al fitxer de configuració. No hauríem de tenir cap més problema amb Windows NT/2000, ja que és el sistema operatiu que millor ha desenvolupat el protocol SMB.

- A Windows 98, no es veu cap ordinador de la xarxa, i entrant a “Toda la red” dona un missatge d'error

La causa és que en entrar a Windows, quan demanava nom d'usuari i contrasenya, hem escollit 'Cancelar', i per tant no hem iniciat sessió dins la xarxa Microsoft. Cal entrar com a un usuari amb nom (i contrasenya opcional). Només entrant ja s'ha creat l'usuari, però els podem gestionar des de 'Mi PC' -> 'Panel de control' -> 'Usuarios'.

- A Windows 98, es veu el servidor, però en intentar entrar demana la contrasenya del recurs `IPC$`



El famós quadre IPC\$

No hem d'escriure cap contrasenya especial; aquesta és la manera que Windows té per dir-nos que estem intentant entrar com a un usuari no permès al servidor. La

solució és crear a la màquina Windows un usuari amb el mateix nom i contrasenya que el que es permet al servidor, o fer a la inversa: afegir al servidor l'usuari que es connectarà des de Windows.

Una altra solució és activar l'usuari invitat, anomenat 'guest' o 'Invitado' i amb contrasenya en blanc; però hem d'evitar fer això a tota costa, ja que posaria en perill les unitats i carpetes compartides.

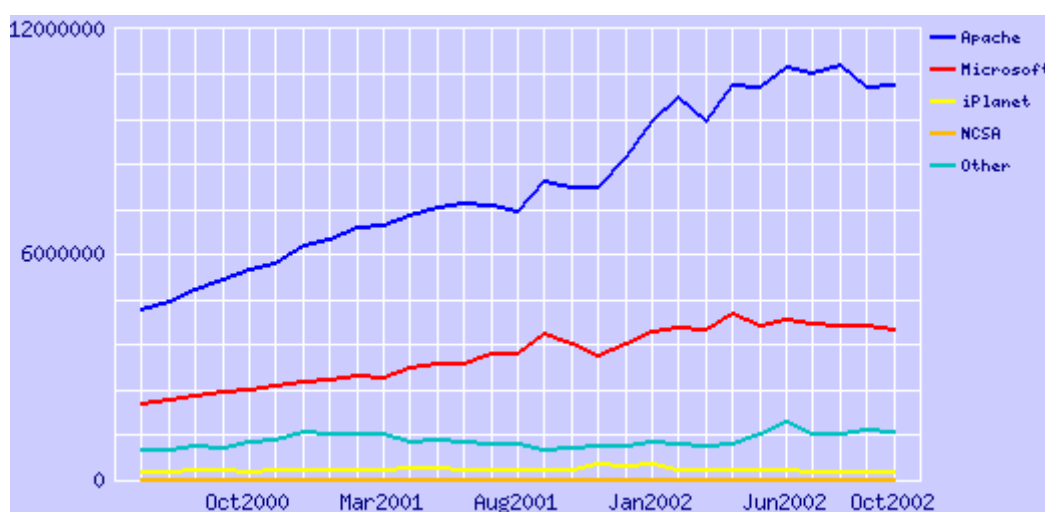
Sens dubte és més fàcil crear l'usuari a Windows 98: només cal escriure `bruguers` en comptes de qualsevol cosa al quadre d'inici de sessió, i posar la contrasenya correcta per crear aquest nou usuari. Per tant, ho deixem de forma que només els usuaris autenticats puguin accedir als recursos compartits del servidor; si algú no posa nom d'usuari en iniciar sessió o se n'inventa un altre és que no té permís per entrar-hi.

Per accedir a la xarxa des de Linux podem usar comandes com `smbclient -L host` per llistar els recursos compartits, `smbmount //host/nomdelrecurs /mnt/samba` per muntar un recurs, `nmblookup host` per conèixer la IP d'un ordinador, o `nbtscan 192.168.0.0/24` per escanejar tota la xarxa. També podem utilitzar programes gràfics com `komba` i `xf samba` des de qualsevol lloc de la xarxa. Altres utilitats que ens poden anar bé són `testparm` per a comprovar la configuració i `smbstatus` per a veure qui hi ha connectat.

2.3.2 Servidor web per a l'Intranet

El servidor web és un dimoni que escolta al port 80 i respon les peticions de documents HTML (o d'altres formats). D'entre les desenes que n'hi ha al mercat (comercials i gratuïts), mirarem quins són els més usats segons la pàgina

<http://www.netcraft.com/survey>



Número de servidors (tots els dominis) Juny 2000 - Octubre 2002

Hem d'afegir que “Microsoft” inclou tots els servidors web d'aquesta marca (el més conegut és IIS), i que iPlanet és el conjunt de tots els servidors Netscape i iPlanet. Això deixa bastant clar que Apache és el millor servidor. A més, és gratuït, de codi obert, i funciona en qualsevol plataforma.

Integrar Apache a Debian és tan senzill com fer un `apt-get install apache` (es suposa que la base de dades d'apt-get ja està actualitzada mitjançant `apt-get update`) i automàticament es baixaran tots els paquets necessaris. Podem fer una petita comprovació escrivint la IP del servidor en el navegador de qualsevol ordinador de

la xarxa. Si ho hem fet bé, apareixerà una pàgina de prova d'Apache.

No és bon hàbit deixar la configuració per defecte, ja que podria no fer tot el que volem, o, pitjor encara, fer més coses del que suposem. Per tant, anem a revisar l'arxiu de configuració del dimoni HTTP a `/etc/apache/httpd.conf`. Gairebé està configurat; només hem de canviar el `ServerAdmin` i posar (o esborrar) l'e-mail de l'administrador, que sortirà quan hi hagi errors; i el `ServerName`, on posarem el nom del host o la IP de l'equip (`bruguers` al nostre cas).

No s'han de fer més canvis. Abans, però, de reiniciar el servidor, esborrarem el contingut del directori on hi ha la web (`/var/www`) i posarem una senzilla creada per nosaltres, amb enllaços i imatges. A la pàgina principal li anomenarem `index.htm` per veure si també funciona (l'original era `index.html`). Per exemple, podem fer un `index.htm` molt senzill així:

```
<BR><BR>
<CENTER>Has entrat al <B>nou servidor web</B> de
l'institut.<BR><BR>
<IMG SRC="obriers_treballant.jpg" ALT="En construcció">
</CENTER>
```

Aquesta pàgina ja ens serveix per provar si accepta noms llargs de fitxers, si accepta extensions que no siguin HTM/HTML, si es veuen les imatges, si el text arriba com a HTML (i no com a text pla) i si un `.HTM` funciona bé com a pàgina principal. Seria interessant donar-se compte de que també és 'Case sensitive', o sigui, que `index.HTM` és diferent de `index.htm`. Lògicament, la raó és que el sistema d'arxius de Linux ja és així.

Després de col·locar a `/var/www` el fitxer `index.htm` i alguna imatge, reiniciarem el servidor fent `apachectl restart` i tornarem a comprovar en el navegador

si surt la pàgina correcta.

I per últim, crearem un usuari anomenat `web`, perquè després posarem un servidor FTP, i ens anirà molt bé poder administrar els fitxers de la pàgina web des de l'FTP. Cal fer un `adduser web` com a root, posar una bona contrasenya, i després modificar l' `/etc/passwd` per canviar el directori d'inici a `/var/www`, que és on es troben tots els arxius de la web. Podem esborrar el directori HOME creat per defecte, `/home/www`. També hem de fer seus tots els arxius de la web, ja que si el directori pertany a root cap usuari el podrà modificar. Ho fem amb `chown web.web /var/www -R` (el `R` fa que canviï els permisos de forma recursiva: al directori i a tot el que hi hagi dintre).

Aquest servidor és disponible només per a la xarxa interna; des d'Internet no es podrà accedir perquè el router no té obert el port 80. Més endavant estudiarem com posar dues pàgines diferents, una per a la xarxa interna i altra per a Internet.

2.3.3 Servidor d'FTP

El FTP serveix per transferir arxius de forma ràpida i senzilla entre ordinadors. Permet compartir -amb nom i contrasenya- tots els arxius o només algunes carpetes a cada usuari, podent establir els permisos que tenen sobre cada element. Per exemple, podem fer que només puguin baixar coses, o només pujar-ne i no canviar res.

Servidors de FTP per a Linux també tenim moltíssims. Alguns molt coneguts són `proftpd` i `wu-ftp`, però com que s'han trobat molts bugs que afecten a característiques extra normalment no utilitzades, molta gent prefereix servidors més senzills que facin només el seu propòsit de compartir arxius.

Per tant, buscarem un servidor amb poques funcions addicionals, senzill, i que ocupi poc (quant més petit sigui, menys errades pot tenir al codi). El que més s'adapta a aquestes condicions és PureFTPd.

Provem a fer `apt-get install pureftpd` (i amb `pure-ftpd`), però cap funciona. La raó és que PureFTPd no ve inclòs en aquesta versió de Debian. Ho podem comprovar amb `apt-cache search pure`: veurem que no hi ha cap paquet on surti “pure”. La forma d'instal·lar el programa és, doncs, anar a la pàgina web (<http://www.pureftpd.org>) a la secció de Downloads i baixar en format `.tar.gz` l'última versió. El que hem baixat són les fonts (en anglès, “sources”) del programa, en llenguatge C, per tant, les haurem de compilar per tal de crear l'executable que funcioni a la nostra màquina. Ens farà falta com a mínim el `gcc` (GNU C Compiler) i una mica de temps (depèn de l'ordinador, però és quasi segur que menys d'una hora).

Com a root, haurem de descomprimir l'arxiu a algun lloc (per exemple /

`usr/src`) amb:

```
tar zxvf arxiu.tar.gz -C /usr/src
```

Llavors podem entrar al directori creat i llegir els fitxers `README` i `INSTALL`, que normalment són a tots els programes que s'han de compilar. Si veiem que no s'ha de fer res especial -com és el cas- seguirem el procediment habitual per compilar:

1. Executar `./configure` per generar un script de compilació adequat per a la nostra màquina en concret. Si falta algun programa o llibreria, s'aturarà i ho mostrarà de forma clara. Podem desactivar o activar moltes opcions (veure `./configure --help`). En el nostre cas, hem comprovat que en compilar sense cap opció especial, el binari resultant no acceptava la opció `-n`, que serveix per limitar l'espai de cada usuari. Per tant, és bona idea activar directament la opció adequada amb `./configure --with-quotas`.
2. Executar `make`, la veritable compilació. Aquest procés -completament automatitzat- pot trigar segons, minuts, o hores depenent del programa i de l'ordinador. No ha de fallar; si ho fa, serà més complicat solucionar el problema ja que probablement es trobi dins del codi font d'algun fitxer del projecte (o sigui, que no tenim la culpa nosaltres). Això sí, és probable que apareguin missatges de “warning” (advertències), que podem ignorar a menys que puguin provocar cap problema greu.
3. Aquest pas és opcional. `make check` farà unes comprovacions per veure si s'ha compilat bé. Si alguna prova falla, no hem de continuar.
4. `make install` copiarà els executables (el resultat de la compilació) als directoris on es troben tots els altres programes de Linux. Fan falta

privilegis de root.

Fet això ja hem aconseguit crear el fitxer executable i intregar-ho al sistema; per tant podem esborrar el directori que hem creat a `/usr/src` (a menys que vulguem fer algun canvi al codi i tornar-ho a compilar).

Ara, cada vegada que encenguem l'ordinador haurem d'executar `pure-ftpd` (el podem deixar en segon pla) i el servei estarà actiu, amb els usuaris actuals del sistema. Per evitar repetir aquesta acció cada vegada, podem crear un script que arrenqui el servei amb les opcions més apropiades (que hem consultat a l'ajuda, a `pure-ftpd --help`):

```
/usr/local/sbin/pure-ftpd -B -A -u 100 -C 5 -n 1000:800  
echo "Arrencant el servidor pure-ftpd..."
```

Descripció de les opcions:

- El `-B` és per fer-lo córrer en segon pla
- El `-A` fa que tots els usuaris estiguin tancats en un entorn `chroot()`, o sigui, que veuen el seu directori personal com a `/` i no poden sortir-ne més.
- El `-u 100` és el mínim UID que cal per connectar-se. root (UID 0) no ho pot fer.
- `-C 5` fa que només es puguin fer 5 connexions per host. És poc probable que un usuari amb bones intencions necessiti fer més d'una a la vegada, però -tampoc hem de limitar-ho a una- podem deixar cinc.
- El `-n 1000:100` limita cada usuari a 1000 arxius o 800 Mb

Gravarem aquest script a `/etc/init.d/pure-ftpd`, donant-li permís d'execució a tothom amb `chmod a+x /etc/init.d/pure-ftpd`, i crearem enllaços a dins dels nivells d'execució en què ens interressi que corri aquest procés (1, 2, 3, i 5, per exemple)

Ja només queda provar de fer un FTP des de fora. Hem de recordar que al router s'han d'obrir els ports 20 i 21 TCP (el 20 és per la transmissió de dades). També és el moment de provar d'entrar com a usuari `web` des de qualsevol client d'FTP (o amb un navegador, amb `ftp://web@IP_DEL_SERVIDOR`), posar la contrasenya, i comprovar que podem pujar i baixar arxius al servidor web. Hem d'evitar posar contrasenyes importants.

2.3.4 Configuració del proxy

En qualsevol xarxa, els usuaris accedeixen a pàgines web prohibides, ja sigui perquè no estan relacionades amb la feina que s'ha de fer amb l'ordinador, o perquè són pornogràfiques o de continguts il·legals.

Una forma de denegar l'accés a certes pàgines és configurar cada navegador web de forma que demani contrasenya quan s'entra a unes direccions web o IPs preestablertes. També podem utilitzar software de filtrat de continguts que analitzin les paraules o fins i tot les imatges.

El problema que presenten aquests mètodes (apart de l'elevat preu) és que requereixen manipular cada un dels ordinadors de la xarxa individualment, quan l'ideal seria haver de crear la 'llista negra' només a un ordinador. És aquí on intervenen els proxy: un proxy és un ordinador que recull totes les peticions web de la xarxa, busca les pàgines corresponents a Internet, i retorna cada pàgina a l'ordinador que l'havia demanat. D'aquesta manera, és només aquest ordinador el que es connecta a Internet, i per tant és fàcil prohibir o admetre IPs.

Pot semblar que el proxy fa més lenta la navegació ja que un sol ordinador ha de fer la feina de molts, però de fet el proxy també està dissenyat per agilitzar-la actuant com a caché. 'Caché' fa referència a una zona de la memòria o del disc, que emmagatzema informació durant un temps i que va buidant-se a mesura que arriben noves dades per omplir-la.

Llavors, un proxy-caché funciona així:

1. Rep una petició HTTP d'un ordinador.
2. Consulta en la seva caché buscant la pàgina demanada.
3. Si no té la pàgina a la seva caché, la baixa d'Internet i queda gravada a la caché. En canvi, si ja tenia la pàgina no és necessari que es torni a connectar a Internet.
4. Retorna la pàgina a l'ordinador que l'havia demanada.

Periòdicament fa comprovacions per evitar que la caché s'ompli, esborrant les dades més antigues o menys sol·licitades. A més, enregistra els accessos als fitxers de log, com fan gairebé tots els altres servidors que hem vist.

En conclusió, un proxy-caché ajuda a reduir molt l'ample de banda utilitzat a una xarxa (degut principalment a les consultes web). No és cap problema la velocitat de transmissió; serà l'adequada perquè disposem de targetes de xarxa 10/100 Mbits/s per a enviar dades d'un ordinador a un altre (per la Intranet), mentre que la connexió cap a Internet arriba com a molt a 2 Mbits/s si és una línia ADSL. A més, cal recordar que gairebé tots els navegadors actuals tenen la seva pròpia caché web local, i per tant s'evitaran peticions web a Internet o, en aquest cas, a l'ordinador proxy.

El proxy-caché per Linux més conegut i més utilitzat és Squid. Té suport per a HTTP, FTP, SSL, SNMP, DNS, control d'accés (ACL), proxy transparent, jerarquies de proxies (protocol ICP), i moltes opcions més. La seva web és <http://www.squid-cache.org>

Farem un `apt-get install squid` per baixar-lo i instal·lar-lo, però hem de modificar la configuració per defecte., a `/etc/squid.conf`. Haurem de canviar els següents paràmetres:

```
http_port 3128
```

```
http_port 8080
```

Aquest és el port utilitzat pel proxy (per defecte 3128). Hem afegit el 8080 perquè és més comú a Europa, però tampoc és necessari treure el 3128; Squid pot rebre les peticions per ambdós ports. El deixem per compatibilitat amb altres programes.

```
cache_peer proxy.xtec.es parent 8080 0 no-query no-digest default
```

Això fa que el nostre proxy tingui un 'pare' i treballi en jerarquia amb ell: tot el que els usuaris de la xarxa demanin al servidor es buscarà a la seva caché, i, si no es troba, la petició passarà al proxy pare. Si aquest pare tampoc troba el que s'ha demanat a la seva caché, ho buscarà a Internet; per tant evitem en tot el possible les peticions innecessàries.

L'institut utilitza el proxy `proxy.xtec.es` pel port 8080 (l'habitual dels proxy-cachés). Hem desactivat les opcions ICP (protocol utilitzat pels proxys) posant els paràmetres `0` com a port ICP i `no-query`. El `no-digest` el posem perquè el fitxer de digests és innecessari quan hi ha un sol pare; i el `default` fa que el servidor utilitzi per defecte aquest parent per trobar tot el que no tingui a la caché.

Òbviament, si no disposem de proxy pare (o sigui, si volem tenir Internet com a pare), no hem de posar aquesta línia.


```
# cache_mem 8 MB
```

Deixem la opció per defecte de 8 Mb. per a la memòria RAM utilitzada com a caché. Es recomana deixar una quarta part de la RAM total de l'ordinador, però com que el nostre servidor en té poca (48 Mb) i a més ha de encarregar-se de moltes més coses, deixarem aquest valor, que de totes maneres és bastant gran com per a gravar els objectes més demanats (entenent objecte com a una imatge, una pàgina, un vídeo, o qualsevol altre element web). S'ha de tenir en compte que apart de la RAM utilitzada com a caché, el procés de Squid necessita més RAM per poder executar-se (com qualsevol altre procés), per tant cal no posar valors molt elevats.

```
cache_dir ufs /var/spool/squid 150 16 256
```

És aquí quan diem a Squid que utilitzi el directori `/var/spool/squid` com a caché. `ufs` és el sistema d'emmagatzematge de Squid. Als paràmetres li diem que pot utilitzar 150 Mb. i que organitzi els objectes creant 16 directoris, cadascun amb 256 subdirectoris i amb els fitxers dintre.

Hem assignat 150 Mb. perquè el disc dur de l'ordinador pot arribar a ser més lent que Internet si s'ha de buscar un arxiu entre milers d'ells, tenint en compte també que 150 Mb. són suficients per guardar gairebé totes les pàgines que s'utilitzaran sovint a l'institut.

Un detall: hem d'assegurar-nos que el propietari del directori utilitzat com a caché sigui el mateix que executi el procés squid, sinó no hi podrà accedir. És poc probable que no sigui així, ja que s'ha creat durant la instal·lació.

```
# ftp_passive on
```

Depenent del router darrere del qual estiguem, haurem de fer que Squid es

connecti als FTPs en mode passiu o no. La millor manera de comprovar-ho és provar-ho amb la configuració per defecte i si no funciona posar l'altra opció.

Al mateix fitxer hem de definir les llistes de control d'accés (ACL). Encara no prohibeixen res, només serveixen per a definir uns grups d'usuaris. Afegirem les següents:

Definim els ordinadors amb la IP 192.168.0.x (la xarxa local de l'institut).

Després farem que només siguin aquests els que tinguin accés al proxy:

```
acl xarxa src 192.168.0.0/255.255.255.0
```

Afegirem més llistes de control d'accés per prohibir certes pàgines. Fan falta tres fitxers, que crearem amb qualsevol editor i guardarem a `/etc/squid` (si no existeix el directori el creem amb `mkdir /etc/squid`). El nom de cada arxiu és de lliure elecció:

- `/etc/squid/dom_no.txt`: aquí escriurem els dominis o hosts d'Internet a què no es podrà accedir, un en cada línia. Per exemple, podem posar `playboy.com`, `penthouselive.com`, `mercabanner.com`, `doubleclick.com`, `doubleclick.net`, `linkexchange.com`, `tradedoubler.com`, `realmedia.com`. Aquests últims són pàgines que es dediquen a l'intercanvi de banners; per tant si les prohibim no en veurem més.

Nota: en comptes de posar desenes de dominis com 'cibersexo.com', 'supersexo.com', 'sexogratis.com', 'quierosexo.com', a continuació crearem una regla per prohibir tots els dominis que continguin la paraula 'sex'.

- `/etc/squid/url_no.txt`: posarem paraules (una per línia) que les URLs no poden tenir. Per exemple, `sex`, `porno`. Això ens estalvia molta feina.
- `/etc/squid/url_yes.txt`: aquí posarem les excepcions a la regla anterior: paraules que, encara contenint la paraula prohibida, són

acceptades. Per exemple: `sexuali` (per a què “sexualitat” i “sexualidad” siguin acceptades).

Definim aquestes tres llistes. Hem posat el `-i` a cada opció per a què la llista de paraules sigui 'case insensitive', o sigui, que no es fixi en les majúscules i minúscules:

```
# Dominis prohibits:
acl dom_no dstdom_regex -i "/etc/squid/dom_no.txt"

# Paraules prohibides:
acl url_no url_regex -i "/etc/squid/url_no.txt"

# Paraules autoritzades:
acl url_yes url_regex -i "/etc/squid/url_yes.txt"
```

Ja hem definit quatre llistes (la de IPs i aquestes tres). Ara hem de dir quins privilegis tenen aquests grups d'usuaris:

```
http_access deny !xarxa
http_access deny dom_no
http_access allow url_yes
http_access deny url_no
http_access allow all
```

Per entendre això cal tenir clar el següent: `deny` = 'denegar', `allow` = 'permetre', `!condició` = “no-condició” (NOT lògic).

És molt important que aquestes línies estiguin en un ordre lògic, ja que Squid fa les comprovacions començant pel principi, i quan una de les condicions es compleix, no continua llegint i fa l'acció associada (`allow` o `deny`). Tal com les hem

posades, els filtres pels que passa una petició que arriba al proxy són:

1. No és de la xarxa local? Doncs si no és de la xarxa local, denego l'accés; si ho és, continuo comprovant.
2. Vol entrar a un domini prohibit? Si és així, denego l'accés; si no, continuo comprovant.
3. Conté la URL una paraula permesa? (Una paraula que sembla prohibida però que l'administrador ha considerat que pot ser utilitzada). Si la té, no segueixo comprovant; deixo passar la petició directament. Si no la té, continuo comprovant.
4. Conté la URL una paraula prohibida? Si la té, denego l'accés.
5. Ja ha passat per tots els filtres: és de la xarxa local, i no conté paraules prohibides ni vol accedir a un domini prohibit. Permeto l'accés a la petició.

Les ACL (llistes de control d'accés) ja estan definides. Continuem amb la configuració restant:

```
cache_mgr institut@domini.com
```

Si hi ha cap problema amb la caché del proxy es mostrarà un missatge d'error amb aquesta adreça d'e-mail. Per defecte posa “[webmaster](#)”, el podem deixar així si no volem donar la nostra direcció de correu.

Si també volem que Squid actuï com a accelerador [httpd](#) (dimoni HTTP), haurem d'afegir aquestes línies. Així aconseguirem que la caché també s'utilitzi en les peticions web que es facin al nostre servidor Apache, i agilitzarà més el seu

funcionament.

```
httpd_accel_host virtual
httpd_accel_port 0
httpd_accel_with_proxy on
```

Per últim, anirà molt bé canviar l'idioma de Squid a espanyol sobretot perquè els usuaris de la xarxa comprendran millor els missatges d'error que puguin aparèixer.

Ho fem amb:

```
error_directory /usr/lib/squid/errors/Spanish
```

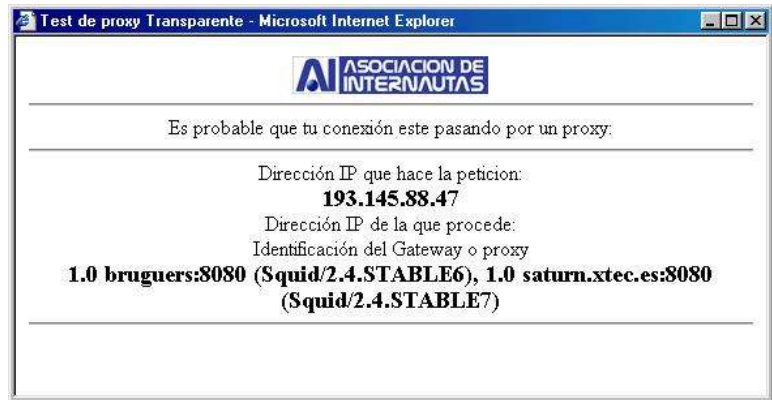
Ja hem acabat amb el fitxer de configuració. No fa falta reiniciar el procés `squid`; només cal executar la següent ordre per què llegeixi la nova configuració:

```
squid -k reconfigure
```

Ara queda la millor part: provar-ho i comprovar que funciona tot a la primera. Només hem d'anar a un altre ordinador de la xarxa, obrir el navegador (sigui quin sigui) i posar a la configuració de 'Proxy' la IP del servidor i el port 8080. Podem crear un host a cada ordinador per no haver d'escriure la IP; de totes maneres deixarem aquest pas per més endavant perquè quan tinguem servidor de DNS, el nom de host es traduirà directament a la IP del servidor (com si fos una direcció d'Internet).

Per provar-ho del tot haurem d'entrar en pàgines web, en FTPs, en llocs que requereixin connexions segures (SSL); entrarem a pàgines lentes per veure si triga menys a tornar a entrar, entrarem a pàgines inexistents per veure els missatges d'error, entrarem des de IPs no permeses, etc. En el nostre cas hem comprovat una millora de la

velocitat apreciable en tots els sentits: les pàgines web carreguen abans, els FTPs van més ràpid, els errors que han d'aparèixer apareixen abans, i res no falla.



Comprovant que funciona el proxy i el pare de la xtec

Ara que ja funciona, haurem de veure com evoluciona el seu estat amb el pas del temps. Els fitxers de log aniran creixent, però no ens hem de preocupar perquè a [/etc/cron.daily/](#) (l'listat de les tasques que s'executen cada dia) hi ha un script anomenat [logrotate](#), que està configurat a [/etc/logrotate.conf](#) per a rotar els logs, entre d'altres, del Squid (a [/etc/logrotate.d/squid](#)).

'Rotar els logs' es refereix a comprimir el fitxer de logs (d'accessos registrats), canviar-li el nom, i tornar-ne a crear un però buit. Per tant, quan veiem fitxers com [/var/log/squid/access.log.7.gz](#), sabem que es tracta d'un log bastant antic (més antic quan més gran sigui el número), i també sabem on trobar el log d'aquell mateix moment: al seu lloc, [/var/log/squid/access.log](#) (sense comprimir ni canviar de nom). L'operació de la rotació de logs de tots els servidors la controla [logrotate](#).

Per acabar, cal tenir en compte que si volem que els usuaris no puguin accedir a Internet per altre mitjà que no sigui el proxy, es pot crear un filtre al router, o es pot desconfigurar cada ordinador client per a què no funcionin els DNS de proveïdor d'Internet contractat, ja que és el proxy també qui es fa càrrec de resoldre noms de host.

2.4 Serveis secundaris

2.4.1 Suport de PHP a la web

Podrem aprofitar molt més el servidor si utilitzem pàgines web que incloguin scripts en PHP. A més, pot ser una pràctica molt útil per als alumnes que han acabat d'estudiar HTML i volen orientar-se a la programació de webs de comerç electrònic i similars.

El PHP és un llenguatge molt potent que s'executa al servidor quan algú demana una pàgina, i que genera un codi HTML que inclou en la pàgina de retorn. Per veure què ens proporciona aquest llenguatge, farem un exemple que després ens servirà per provar si s'ha instal·lat correctament al servidor.

Suposem que al servidor hi pugem un fitxer anomenat `prova.php` amb el següent contingut:

```
<BODY BGCOLOR=YELLOW>
<BR><BR>Això és <B>HTML</B> normal.<BR>
<?
echo "Aquí comença el codi PHP<br>";
for ($size=1; $size<=7; $size++) {
    echo "<font size=$size color=#".(2+$size)."90000>";
    echo "Tmany $size</font><br>\n";
}
?> <BR> Això torna a ser HTML; el PHP ja s'ha acabat.
</BODY>
```

Si carreguéssim aquest fitxer al navegador (a la direcció

http://IP_DEL_SERVIDOR/prova.php), el codi que rebríem seria:

```
<body BGCOLOR=YELLOW>
<br><br>Això és <b>HTML</b> normal.<br>
Aquí comença el codi PHP<br><font size=1 color=#390000>Tamany 1</font><br>
<font size=2 color=#490000>Tamany 2</font><br>
<font size=3 color=#590000>Tamany 3</font><br>
<font size=4 color=#690000>Tamany 4</font><br>
<font size=5 color=#790000>Tamany 5</font><br>
<font size=6 color=#890000>Tamany 6</font><br>
<font size=7 color=#990000>Tamany 7</font><br>
<br> Això torna a ser HTML; el PHP ja s'ha acabat.
</body>
```

S'ha de destacar que res del codi PHP que hi havia escrit el podrà veure mai l'usuari (encara que intenti descarregar el [.php](#)), ja que el servidor genera i envia només codi HTML.

Per tant, hem fet un codi que automatitza una operació mitjançant un bucle `for`; en aquest cas el que fa és mostrar diferents textos de mida cada vegada més grans, i de diferent color. Però les possibilitats són inesgotables: es poden crear dinàmicament pàgines que demana l'usuari, es poden fer servir cookies, contrasenyes, sessions que expiren al cap d'un temps, comptadors, es pot accedir a bases de dades amb el motor MySQL, i moltíssimes coses més. Hi ha moltes pàgines importants fetes en llenguatge PHP, com els 'blog' ('web log', xarxes de pàgines interrelacionades mitjançant enllaços), o portals com ara Barrapunto (www.barrapunto.com) i similars. També hi ha eines especials, com PHPNuke, que donen les bases per crear un d'aquests portals sense haver d'aprendre a programar en PHP.

Per al nostre servidor, ens ocuparem d'instal·lar PHP, integrar-ho a Apache, i

comprovar que funcioni amb l'exemple anterior (el de les fonts). La instal·lació de MySQL, la seva configuració, i el procés per instal·lar utilitats com PHPNuke no el descriurem aquí, però no és gaire difícil; només cal fer com en els altres programes (utilitzant `apt-get`) i seguir les indicacions del fitxers `INSTALL` i/o `README`.

Un `apt-cache search php` ens revela que el paquet més apropiat s'anomena `php4`; per tant fem un `apt-get install php4` i es baixarà junt amb les seves dependències. A la seva web www.php.net, trobarem tota la informació sobre el llenguatge, i és on llegir les instruccions d'instal·lació. També hi trobarem notes dels usuaris sobre tots i cada un dels temes de la documentació; per tant és poc probable que no puguem resoldre els problemes que ens surtin.

En aquest cas, veiem a la web que hem d'afegir unes línies a l' `/etc/apache/httpd.conf` per tal que entengui el format PHP. Anem al fitxer i trobem aquestes línies, comentades:

```
#AddType application/x-httpd-php .php
#AddType application/x-httpd-php-source .phps
```

Només cal descomentar-les (traient el `#` del principi) per afegir el tipus d'arxiu PHP. També busquem la línia que ha afegit el `apt-get` i li traiem el `#` si el té. Ha de quedar semblant a:

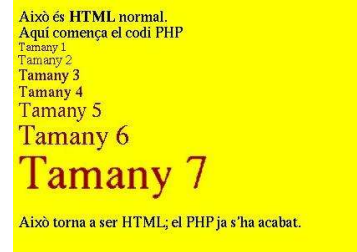
```
LoadModule php4_module /usr/lib/apache/1.3/libphp4.so
```

També hem de dir-li a apache que la pàgina a carregar per defecte si no es posa res ja no és només `index.htm`, sinó que també val `index.php`. Busquem un fragment on posa:

```
<IfModule mod_dir.c>
    DirectoryIndex index.html index.htm index.shtml index.cgi
</IfModule>
```

I afegim `index.php` al final de la llista. Així, intentarà carregar primer `index.html`, si no la troba `index.htm`, després `index.shtml` i així successivament. Podem posar-ho en l'ordre que vulguem, encara que no hauria d'haver més d'un fitxer `index` al directori.

Ara reiniciem Apache (`apachectl restart`) i tornem a carregar la pàgina PHP en el navegador. Aquesta vegada ha de funcionar, i veurem el codi HTML esmentat abans.



Pàgina PHP de prova

2.4.2 Suport de CGIs a la web

CGI fa referència a Common Gateway Interface, i és un estàndard per a poder executar scripts (petits programes) a les pàgines web. És molt semblant a PHP, amb la diferència de que un CGI pot estar escrit en molts llenguatges de programació, entre ells Perl, C/C++, Fortran, TCL, Visual Basic, AppleScript, o shells de Linux. La majoria, però, estan escrits en Perl, i gran part de la resta són scripts per a la shell de Linux.

Veurem un programa CGI més endavant, després d'haver-ne preparat el suport a Apache. No calen instal·lar nous paquets; només hem de comprovar que tinguem instal·lat el Perl amb `perl --version` i després editarem el fitxer de configuració de l'Apache `/etc/apache/httpd.conf`:

Veiem que la línia `LoadModule cgi_module /usr/lib/apache/1.3/mod_cgi.so` està descomentada. Perfecte. També llegim prop de la línia `ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/` que els CGIs no van a `/var/www/cgi-bin/`, sinó que aquesta direcció està redirigida a `/usr/lib/cgi-bin/`, probablement per evitar que al directori web hi hagi programes executables, deixant d'aquesta manera només els elements web. Més endavant trobem les següents línies comentades:

```
# To use CGI scripts:
#AddHandler cgi-script .cgi .sh .pl
```

Hem de treure el signe de comentari a la línia de `AddHandler` per fer que entengui els `.cgi` com a programes i no com a fitxers de text normals.

Gravem, i reiniciem l'Apache amb `apachectl restart`

Per a provar si funciona, creem un fitxer amb el següent contingut:

```
#!/usr/bin/perl

print "Content-type: text/plain", "\n\n";

print "Aquest script CGI fet en Perl funciona!", "\n";

$uptime = `/usr/bin/uptime` ;

($load_average) = ($uptime =~ /average: ([^,]*)/);

print "Càrrega de la CPU: ", $load_average, ".", "\n";

print "Actualitza la pàgina i veuràs com canvia.", "\n";

exit (0);
```

Hem de guardar aquest CGI a `/usr/lib/cgi-bin/primer.cgi` (no a `/var/www/cgi-bin/primer.cgi`!, ja que Apache té l'`alias` que hem comentat abans), i hem de donar-li permís d'execució amb `chmod a+x /usr/lib/cgi-bin/*` (aquest és un pas molt important; si no ho fem sortiran errors a la web). Quan ja ho hem fet, obrim un navegador i entrem a `http://IP_DEL_SERVIDOR/cgi-bin/primer.cgi`. I ara sí que ens hem de referir al CGI com si realment estigués al directori `/var/www/cgi-bin`; l'`àlias` sabrà on trobar el veritable fitxer.

Si tot funciona bé, veurem el missatge en format text (sense HTML ni altres codis pel mig). Si no funcionés, haurem de comprovar si podem executar el script localment (`perl /usr/lib/cgi-bin/primer.cgi`), si té permís d'execució, si existeixen els mòduls de Perl necessaris, si ens hem deixat de mirar alguna cosa relacionada amb `cgi` a la configuració de l'Apache, i si l'hem reiniciat. En tot cas, podem consultar `/var/log/apache/error.log` per veure els missatges d'error detallats.

Amb els CGI podem fer eines molt potents que interactuïn amb el sistema i que ens permetin configurar-ho remotament des de la mateixa pàgina web. De fet, de programes d'aquest tipus ja n'hi ha; el més usat és Webmin (<http://www.webmin.com>). Però també és important saber que moltes tècniques per a accedir a servidors web i treure'n informació privada aprofiten CGIs poc usats o mal programats. Encara que gairebé tots aquests errors afecten a altres sistemes operatius, convé no abusar dels CGI.

2.4.3 Estadístiques d'accés a la web

Per veure com marxa la nostra pàgina -un dels serveis més importants de l'ordinador- no podem estar revisant els logs d'Apache a cada moment, ja que la informació que mostren és molt extensa i detallada, probablement més del que nosaltres volem.

Aprofitarem que hem afegit el suport per a executar programes CGI i posarem un analitzador de fitxers de registre d'accés. L'únic que fan és interpretar el fitxer `/var/log/apache/access.log` i generar una pàgina HTML amb estadístiques i gràfiques sobre els documents més demanats, la procedència dels visitants i les hores en què entren. Així és molt més fàcil saber si tot funciona com ho hem previst,

D'analitzadors de logs també n'hi ha molts (gratuïts i comercials); es diferencien sobretot en la velocitat a la que treballen amb els logs, que poden ocupar molts gigabytes. El que utilitzarem nosaltres és AWStats, un de gratuït, molt ràpid, configurable i senzill d'utilitzar. Està escrit en Perl, i té suport per a molts idiomes, entre ells l'espanyol i el català; per tant és apte per a que també els visitants puguin veure i entendre les estadístiques..

Un `apt-cache search awstats` ens revela que Debian incorpora aquest paquet en la seva distribució de Woody, però un `apt-cache show awstats` ens mostra que aquesta és la versió 4.0. Com que a la seva web <http://awstats.sourceforge.net> diu que la versió estable actual és -en el moment d'escriure això- la 5.3, val la pena baixar el programa de la web i instal·lar-ho nosaltres. Podem consultar el ChangeLog per veure

que hi ha hagut molts canvis importants de la versió 4.0 a la 5.3

Si ho hem instal·lat amb `apt-get` i hem descobert la nova versió després, podem desinstal·lar el programa tan fàcilment com ho hem instal·lat: `apt-get remove awstats`

Per tant, hem d'anar a la web <http://awstats.sourceforge.net> i a la secció de 'Download' baixar el `.tar.gz` (o `.tgz`) que hi ha a la secció 'Last Stable'. El descomprimim i seguim les indicacions detallades que hi ha a la mateixa web (al fitxer que hem baixat també hi són, però potser no són tan actualitzades). Destaquem la opció de posar `Lang="es_cat"` per veure les pàgines en català. Per fer l'actualització automàtica de les dades, programarem el sistema mitjançant `crontab` aprofitant el `cron.daily` que hi ha a `/etc/crontab`. Creem un script a `/etc/cron.daily/awstats` amb el següent contingut:

```
#!/bin/sh
/usr/lib/cgi-bin/awstats.pl -config=bruguers -update
```

Li donarem permís d'execució amb `chmod a+x /etc/cron.daily/awstats`, i ja haurem fet que cada dia, a l'hora especificada a `/etc/crontab` (que vam canviar després de la instal·lació de Debian), s'actualitzin les estadístiques.

Ja està tot preparat; només queda provar-ho entrant a la pàgina `http://IP_DEL_SERVIDOR/cgi-bin/awstats.pl`, on veurem unes completes estadístiques sobre les visites al nostre servidor. Podem treure informació, si volem, editant el fitxer de configuració anterior `/usr/lib/cgi-bin/awstats.bruguers.conf`

2.4.4 Connexions per Secure Shell

Volem que els usuaris normals i els administradors es puguin connectar al PC remotament i utilitzar-lo com si hi fossin davant: que puguin obrir una consola, identificar-se amb el nom i contrasenya, i tenir accés a totes les comandes, programes i fitxers que els pertanyen. Si un usuari amb més privilegis es connecta, pot fer-ho absolutament tot: crear nous usuaris, desconnectar-ne d'altres, apagar l'ordinador, posar música, i fins i tot obrir i tancar el CD.

La forma tradicional de fer això és muntant un servidor de Telnet, i després connectant des d'altra màquina amb `telnet IP`. Aquest sistema només té un inconvenient: tota la informació viatja per la xarxa en format text, o sigui, sense encriptar. Per tant, qualsevol persona que pugui interceptar paquets (calen privilegis de root) podria veure tot el que envia i rep cada un dels usuaris que hi han connectats, fins i tot les contrasenyes.

La solució a aquest perillós sistema consisteix a utilitzar el protocol SSH en comptes de Telnet. SSH (sigles de Secure Shell) és un tipus de connexió idèntica a Telnet, però encriptada amb claus RSA. A més, té molts altres sistemes de seguretat que eviten atacs que es poden fer amb Telnet, com per exemple l'IP spoofing. Fa servir el port 22 de TCP, a diferència del Telnet, que utilitza el 23.

Per al servidor necessitem la versió lliure i multiplataforma del protocol SSH, anomenada OpenSSH. Com que és un programa molt utilitzat en màquines Linux professionals, probablement ja el tenim instal·lat i amb el dimoni funcionant; ho podem comprovar mirant si el procés `sshd` s'està executant, amb `ps axu | grep sshd`, o

també podem mirar si tenim el port 22 de TCP obert, amb `nmap bruguers -p 22`

En cas que no el tinguem instal·lat, només cal fer un `apt-get install ssh` i respondre les preguntes que es fan. Sempre podem tornar a fer aquesta configuració guiada per preguntes amb un `dpkg-reconfigure ssh`.

Una vegada instal·lat i en funcionament, haurem de comprovar que funcioni: el primer serà fer un `ssh` des de la mateixa màquina. Com a qualsevol usuari, fem un `ssh bruguers` i ens apareixerà un missatge dient que no som un usuari autoritzat. Els usuaris autoritzats són aquells que no han d'escriure la seva contrasenya per accedir-hi. Per seguretat, no en crearem cap d'aquests. Diem `yes` per continuar amb la connexió, i el missatge no tornarà a aparèixer més en aquesta màquina.

Sempre que connectem a altra màquina, intentarà accedir-hi amb el nom de l'usuari que està realitzant la connexió. Si volem especificar l'usuari, hem de posar `ssh màquina -l usuari` (l=login). De qualsevol manera, ens demanarà una contrasenya. Òbviament, haurem de posar la contrasenya de de la compta que té l'usuari en l'ordinador a què es connecta (no en el propi ordinador). Després d'escriure la contrasenya, entrarem al sistema de forma normal. Podem sortir de la shell amb el comand `exit`.

Podem connectar-hi des de qualsevol ordinador i sistema operatiu, usant per exemple `ssh` a Linux, Putty a Windows o NiftyTelnet a Macintosh. Cada nou host que connecti al servidor farà aparèixer el missatge anterior de “Màquina desconeguda; ara l'he afegida a la llista de màquines conegudes.” També hem de provar a accedir des d'Internet; si no funciona, s'ha de comprovar que el router filtri el port 22 TCP i el redirigeixi cap al servidor.

Finalment, cal dir que no hem d'estar totalment tranquils només per haver instal·lat SSH; hem de vigilar que les màquines que utilitzem per fer la connexió no tinguin cap programa que pugui enregistrar les contrasenyes (com per exemple un keylogger).

2.4.5 Web per a Internet

Ja hem posat anteriorment el programa Apache servint una web als ordinadors de la xarxa interna. A aquesta web no es pot accedir des d'Internet posant la IP de la línia ADSL, perquè tota la xarxa de l'institut està darrere d'un router.

Ara volem que hi hagi dues webs diferents, una amb informació general de l'institut per als visitants d'Internet i una altra només per als alumnes i professors (usuaris de la xarxa interna) on puguin veure informació que no estigui a l'abast del públic general, com per exemple notes, horaris de guàrdia dels professors, dates de les avaluacions, etc.

Hi ha molts mètodes per servir pàgines diferents a xarxes diferents, però absolutament tots requereixen que s'obri el port 80 al router redirigit cap a un port privat del servidor (normalment, també el 80). Algunes de les idees que podem implementar són:

- Una solució absurda i poc factible: posar dos Apache al servidor, cadascun escoltant a un port diferent (ex. 80 i 81), i utilitzar el 80 per a Internet i el 81 per a la xarxa interna. Com hem dit, és absurd perquè donaria conflictes en la configuració, consumiria el doble de memòria, i faria que s'hagués d'escriure el número de port als navegadors web.
- Posar un Apache i altre servidor web al mateix ordinador: és igual d'absurd, perquè el que volem és un programa que serveixi pàgines web, i això ja ho fa Apache. Per tant, l'altre servidor no aportaria res de nou.

- Posar una altra màquina amb un altre Apache (tots dos escoltant al port 80) i utilitzar-ne un per a Internet i altre per la xarxa. No és intel·ligent, perquè estariem desaprofitant els dos ordinadors. Només fa falta un per fer de servidor d'una xarxa senzilla.
- Utilitzar el mateix ordinador, però posar-hi dues targetes de xarxa per tenir dues direccions IP. És massa complicat, però funcionaria; Apache ho permet.
- Utilitzar només una targeta de xarxa, però amb IPs virtuals. És més simple que la solució anterior, però continua sent massa complicat. Tant aquest mètode com l'anterior són suportats per Apache i es coneixen amb el nom de 'IP-based Virtual Host Support'. El que fa és mirar quina IP s'ha demanat i mostrar una pàgina o una altra.
- Afegir àlies al servidor DNS que muntarem en el següent apartat per fer que dos noms, per exemple [interna.com](#) i [externa.com](#), apuntin a la mateixa IP, la del servidor web; i configurar Apache per què distingeixi què s'ha posat per accedir-hi i mostrar una pàgina o una altra en conseqüència. Això és conegut a Apache com a 'Name-based Virtual Host Support'.
- Utilitzar només una pàgina, però fer que el contingut canviï depenent de la procedència del visitant.
- Utilitzar només una pàgina i posar un link a una 'zona restringida' que demani contrasenya. Com que és molt complicat de cara als usuaris, tampoc no és el millor mètode.

Les solucions més senzilles i efectives són les dels 'Virtuals Hosts' i les de la

pàgina dinàmica, però la primera possibilitat l'haurem d'eliminar perquè, llegint tota la documentació oficial a <http://httpd.apache.org/docs/vhosts>, veiem que només funciona en navegadors que ho suportin. Per tant, utilitzant un navegador senzill o fent nosaltres mateixos la petició HTTP, aconseguiríem veure qualsevol de les dues pàgines! La solució que ens dóna Apache és la d'executar dos dimonis `httpd`, però això requereix el doble de memòria, i és molt més complicat de configurar: dos processos, dos configuracions, dos registres de log, etc. I el doble de problemes. (No obstant, és apte per a les webs de dues empreses que no vulguin estar juntes de cap manera perquè volen diferents configuracions).

Per tant, el millor és deixar Apache tal com està i utilitzar el PHP i CGIs que ja hem configurat per detectar si el visitant ve de la xarxa interna o d'Internet. Aquest mètode presenta altres avantatges:

- Es poden compartir recursos amb molta facilitat ja que tot es troba al mateix directori, `/var/www`. Per tant, podem tenir zones comunes (cosa que amb els altres mètodes no podríem fer). També podem fer que els usuaris d'una xarxa ho puguin veure absolutament tot, i els altres només una part.
- Els visitants no se n'adonen que estan sent classificats en dos grups, ja que no veuen res d'estrany (a menys que hi posem missatges d'informació a la web). Això és bo perquè si un hacker veu des d'Internet que se l'ha classificat com a usuari no autoritzat, l'entraran més ganes d'entrar com a usuari privilegiat.
- Si ho fem amb PHP, podrem fer pàgines dinàmiques (amb informació no constant) o inclús que accedeixen a una base de dades MySQL per fer llistats, consultes, afegir registres, ...

- És molt senzill i és fàcil de personalitzar.

Aquest mètode, però, també té alguns inconvenients, bàsicament perquè l'autenticació per IP mai dona molta seguretat; hi ha moltes tècniques per fer creure a un ordinador que som una IP determinada quan en realitat no ho som. L'IP spoofing i l'ARP poison són les més comuns, però evolucionen constantment per crear atacs cada vegada més sofisticats, fins al punt que és possible gairebé convertir-se en l'ordinador desitjat i aprofitar-se dels seus privilegis.

Per evitar problemes, a la zona privada només deixarem accedir a ordinadors de la xarxa interna (amb IP 192.168.0.x) exceptuant 192.168.0.1 (router), 192.168.0.0, 192.168.0.255. El router NO pot accedir a la pàgina, encara que sigui de la nostra xarxa. Ho hem de tenir en compte perquè el router és l'element que separa Internet de la xarxa interna, per tant si algú aconsegueix entrar al router (192.168.0.1) podria accedir-hi als llocs on es deixa entrar a les IP que comencen per 192.168.0. Les direccions acabades en 0 i en 255 són especials i no les poden utilitzar els ordinadors com a IP, per tant millor fem que no puguin accedir tampoc per si de cas.

Posarem el codi a un fitxer anomenat `aut.php`:

```
<?
$ip = GetHostByName($_REMOTE_ADDR); // Capturem la IP
echo "<BR> Accedeixes amb la IP $ip <BR>";

$aut=strpos($ip,"192.168.0."); // Si la IP conté el text 192.168.0. és
autoritzat

if ($aut === false || $ip=="192.168.0.1" || $ip=="192.168.0.0" ||
$ip=="192.168.0.255") { // Nota: sí, són TRES signes = al principi
```

```
    exit("No estàs autoritzat!"); // Tanca la connexió i no continua llegint
la resta de la pàgina
}
?>
```

I ara només cal afegir una línia a cada pàgina que vulguem protegir. Per exemple, a aquesta pàgina, [index.php](#), només hi poden accedir 192.168.0.x excepte .0 i .255:

```
Aquesta pàgina requereix una IP autoritzada... <BR>
```

```
<? include 'aut.php'; ?>
```

```
Perfecte, estàs autoritzat.<BR>
```

```
<U>Aquests continguts són per a tu.</U><BR>
```

Aquest sistema ja és bastant segur per als usuaris normals; de totes maneres, no haurem de posar informació privada ja que PHP també pot tenir bugs. Si cal, posarem una contrasenya a algunes pàgines, o utilitzarem 'sessions' (<http://www.php.net/manual/es/ref.session.php>).

2.4.6 Servidor DNS

Per evitar haver d'escriure la IP del servidor cada vegada que hi vulguem accedir, podem escriure el seu nom de host i fer que un DNS (Domain Name Server) el tradueixi a la IP adequada (exemple: `tecno.bruguers` a `192.168.0.53`). Com que els servidors DNS que el nostre ISP (Internet Service Provider) no els podem modificar, muntarem el nostre propi servidor DNS local. És un servei que utilitza el port 53 de TCP i el 53 d'UDP.

Utilitzarem la versió de la ISC (Internet Software Consortium, www.isc.org) del protocol DNS, que s'anomena BIND (Berkeley Internet Name Domain). Consta del servidor de noms de domini (`named`) i d'utilitats relacionades amb DNS. La versió actual és la 9, que té suport entre altres coses per fer caché dels DNS (un mateix domini el resoldrà molt més ràpid la segona vegada que la primera). De totes maneres, convé instal·lar sempre l'última versió de `bind`, perquè s'han trobat bugs importants a versions anteriors.

Farem un `apt-get install bind9 bind9-doc dnsutils`, encara que el paquet `dnsutils` probablement ja està instal·lat. El `bind9-doc` és opcional, però com que la configuració és difícil ens anirà bé la documentació (sobretot la que trobem a Internet).

Ara ja tenim el dimoni `named` funcionant a la perfecció i llist per a resoldre hosts; però el millor és revisar la configuració a `/etc/bind/named.conf`. Nota: als fitxers de configuració hem de respectar els signes i la tabulació (s'utilitzen sobretot tabuladors i punts).

Començarem afegint les IPs dels servidors DNS que ens ha donat el nostre ISP. S'utilitzaran per resoldre peticions de noms de domini que no siguin de l'Intranet.

```
forwarders {
    193.145.88.16;
    193.145.88.18;
};
```

Després afegirem el nom del domini, que serà `bruguers`. Coincideix amb el nom de host del servidor, però no passa res, ja que en definirem altres, com `www` o `proxy`, per no embolicar-se. Afegim les següents línies:

```
zone "bruguers" {
    type master;
    file "/etc/bind/bruguers.hosts";
};
```

Hem de crear una altra zona per a fer la operació inversa (traduir una IP a un nom de domini). Per això crearem la zona anomenada `0.168.192.in-addr.arpa`, que és la IP `192.168.0.*` invertida (tots els PCs de la xarxa tenen IP `192.168.0.qualsevolnombre` ja que la seva màscara de subxarxa és `255.255.255.0`). Per a aquesta zona posem:

```
zone "0.168.192.in-addr.arpa" {
    type master;
    file "/etc/bind/192.168.0.rev";
};
```

Ara cal, és clar, crear `/etc/bind/bruguers.hosts` i `/etc/bind/192.168.0.rev`. Començarem pel primer:

```
bruguers. IN      SOA      linux. root (
                  1          ; Serial
```

```

                                8H      ; Refresh
                                2H      ; Retry
                                4W      ; Expire
                                1D )    ; Minimum TTL
bruguers.      IN      NS      linux.
router.bruguers.  IN      A      192.168.0.1
linux.bruguers.  IN      A      192.168.0.200
proxy.bruguers.  IN      CNAME   linux
www.bruguers.   IN      CNAME   linux
t01.bruguers.   IN      A      192.168.0.131

```

Els paràmetres de la primera secció (comentats a partir del caràcter “;”) són més útils per a DNS públics (els que serveixen per a Internet). Especifiquen temps, que pot anar en segons o en múltiples majors (ex: 2H = dues hores, 1D = un dia, 4W = quatre setmanes). Deixarem aquests valors que hem trobat a configuracions per defecte a pàgines de documentació.

Cada nom de host que vulguem definir ha de seguir la mateixa estructura. Les paraules clau que hi ha al fitxer són:

IN NS: indica que aquest equip és el servidor DNS

IN A: afegeix un nom de host i la seva IP corresponents

IN CNAME: defineix un àlies per a un host ja conegut.

L'altre fitxer, `/etc/bind/192.168.0.rev`, ha de contenir el mateix però preparat d'una altra manera:

```

0.168.192.in-addr.arpa.  IN      SOA      linux. root (
                                1      ; Serial
                                8H     ; Refresh
                                2H     ; Retry

```

```

                                4W      ; Expire
                                1D )    ; Minimum TTL
0.168.192.in-addr.arpa.      IN      NS      linux.
1.0.168.192.in-addr.arpa.   IN      PTR     router.bruguers.
200.0.168.192.in-addr.arpa. IN      PTR     linux.bruguers.
131.0.168.192.in-addr.arpa. IN      PTR     t01.bruguers.

```

Els paràmetres són els mateixos, i els hosts definits també. Quan afegim més al primer fitxer també els haurem de posar a aquest segon, naturalment seguint el mateix model (recordem que s'ha d'utilitzar la tecla TAB per separar els camps).

Hem acabat la configuració. El que farem ara és buidar el fitxer `/etc/resolv.conf` del servidor per treure les IPs de l'ISP contractat, ja que ara ja no fan falta (el nostre servidor DNS es comunica directament amb els root servers). En deixar el fitxer en blanc, Linux intenta resoldre els noms de dominis a la mateixa màquina.

Reiniciem `named` executant `rndc reload`, anem a una altra màquina, configurem el DNS amb la IP privada del servidor, reiniciem si cal (depèn del sistema operatiu), i provem a fer un `ping` a `linux.bruguers`

```

PING linux.bruguers (192.168.0.200) from 192.168.0.132 : 56(84) bytes of
data.
64 bytes from linux.bruguers (192.168.0.200): icmp_seq=1 ttl=255
time=0.778 ms
64 bytes from linux.bruguers (192.168.0.200): icmp_seq=2 ttl=255
time=0.667 ms
64 bytes from linux.bruguers (192.168.0.200): icmp_seq=3 ttl=255
time=0.574 ms

```

Funciona! Ha traduït `linux.bruguers` a la IP corresponent, 192.168.0.200.

També hem de comprovar que funcionin els àlies `www.bruguers` i `proxy.bruguers`, i que

cadascuna de les estacions que hem afegit també sigui accessible (a l'exemple hem definit `t01.bruguers`). No hem d'oblidar provar també a resoldre pàgines d'Internet (ex. `google.com`). Si mesurem quant triga la primera vegada i quant les posteriors comprovarem l'eficàcia de la seva memòria caché.

Podem fer proves més detallades amb comandes com `dig router.bruguers` per veure si troba la IP o `dig -x 192.168.0.1` per veure si també funciona a la inversa (hauria de mostrar `router.bruguers`).

Quan tot funcioni, hem de tornar a editar el fitxer `/etc/resolv.conf` del servidor per fer que quedi així:

```
domain bruguers
nameserver 192.168.0.200
nameserver 193.145.88.16
nameserver 193.145.88.18
```

Això fa que les peticions DNS les resolgui el propi servidor (`192.168.0.200`), però que si per alguna raó falla (per exemple, si el procés `named` mor), el DNS funcioni de manera estàndard passant les peticions als DNS del nostre ISP. El `domain bruguers` fa que si no especifiquem res entengui els noms com a membres del domini anomenat `bruguers` (per exemple, `proxy` faria referència a `proxy.bruguers`, `tecno` a `tecno.bruguers`, etc.). Aquesta opció la podem posar a tots els ordinadors del domini (es farà de distintes formes depenent del sistema operatiu).

El servei de DNS ja funciona; però ens queden dues tasques per fer:

- Pensar en un nom per a cada ordinador de la xarxa. Si pot ser, que siguin més descriptius que `ord01`, `ord02`, `ord03`, ... Bons noms són: `tecno`, `bibliol1`, `biblio2`, `almaster`, `profes`, etc. També podem posar noms més

creatus; o -per què no?- diferents àlies a cada ordinador.

- Configurar cada ordinador per fer que utilitzi el servidor com a DNS. Ja que hem de fer aquest tediós pas, aprofitarem per fer les altres configuracions, com la del proxy. L'avantatge és que ara només cal posar [proxy.bruguers](#) en comptes de la IP sencera a cada navegador de cadascun dels PCs.

2.4.7 Firewall

En un ordinador amb tanta importància dins la xarxa no ens podem oblidar de la seguretat: com que és un servidor té molts ports oberts i ofereix moltes formes de connexió, d'entre elles potser algunes són indesitjades.

Per a fer que només s'hi accedeixi utilitzant els serveis que hem configurat, utilitzarem un tallafocs ('firewall' en anglès), que accepta, denega o ignora els paquets que han de passar per l'ordinador. Hi ha una altra eina més complexa i potent que els tallafocs: els IDS (Intrusion Detection Service) que detecten i registren patrons estranys en l'ús de la xarxa, com poden ser el tràfic a altes hores de la matinada o l'arribada de paquets TCP amb capçalera errònia.

Degut a la complexitat dels IDS, posarem només un tallafocs ja que fa tot el que necessitem. Ho farem a nivell de nucli (és el kernel qui controla el filtrat de paquets) utilitzant el programa [iptables](#). Una curiositat: com que han sortit bastants versions estables del kernel de Linux amb moltes diferències entre elles, es va anomenar de forma diferent al filtrat de paquets de cadascuna. Així, tenim que el programa és [ipfwadm](#) per a kernels 2.0, [ipchains](#) per a 2.2 i [iptables](#) per a 2.4. Nosaltres utilitzarem només [iptables](#), el del kernel 2.4 (ja s'ha explicat com actualitzar el kernel fàcilment a la secció de 'Configuracions bàsiques').

Com que és una eina del nucli, és probable que ja el tinguem instal·lat ([iptables](#) per provar-ho, `apt-get install iptables` per instal·lar-ho). Ara començarem per veure com funciona aquest tallafocs:

Tenim tres taules, [INPUT](#), [FORWARD](#) i [OUTPUT](#), per catalogar els paquets:

- [INPUT](#): els paquets que arriben al servidor des de l'exterior o des de la mateixa màquina passen per la taula d'[INPUT](#) ('entrada'). És d'on venen tots els atacs.
- [FORWARD](#): els paquets que arriben al servidor per ser enrutats cap a un altre lloc passen per la regla [FORWARD](#) ('fer que continuï'). S'ha d'activar una opció abans per permetre el forwarding.
- [OUTPUT](#): els paquets generats al mateix ordinador i llestos per a ser enviats cap al exterior passen per la taula [OUTPUT](#) ('sortida').

Cada taula consta d'una sèrie lògica de regles (ordenades) que decideixen el destí del paquet. El paquet acabarà bàsicament d'una d'aquestes formes:

- [ACCEPT](#): es deixa passar el paquet; el firewall no actua per res.
- [REJECT](#): es denega el paquet. El firewall contesta dient que no vol fer la connexió, per tant l'emissor del paquet se n'adona.
- [DROP](#): ignora el paquet. És com si l'ordinador estigués apagat i el paquet es perdés.

La nostra feina si volem crear un firewall és definir aquestes regles per a les tres taules ([INPUT](#), [FORWARD](#) i [OUTPUT](#)), i per fer-ho el primer és crear un esquema de cada taula. Per exemple, un servidor web podria fer el següent:

Paquets rebuts ([INPUT](#)):

1. De la interfaz `lo` (loopback, el propi ordinador): acceptar
2. Ve al port 80 TCP: acceptar

3. D'altra manera, descartar

Paquets de pas ([FORWARD](#)):

1. Descartar

Paquets enviats ([OUTPUT](#)):

1. De la interfaz loopback: acceptar

2. Surt del port 80 TCP: acceptar

3. D'altra manera, descartar

Això és només un exemple d'un servidor molt simple, però ara necessitem fer el tallafocs apropiat per a la nostra màquina. Com que el que hem de fer és tancar ports, primer hem de saber quins tenim oberts, mitjançant un escanejat de ports. Hi ha molts programes que ho fan, però utilitzarem [nmap](#), que és el més utilitzat pels hackers de Linux.

Anirem a altra màquina (les connexions les hem de fer des de fora), instal·lem [nmap](#) (preferiblement sota el sistema Linux), i executem el següent comandament com a root:

```
nmap -sS IP_DEL_SERVIDOR -p 1-65535
```

Això fa un escanejat de ports de tipus SYN (el tipus per defecte quan ho fa root), i prova tots els ports TCP (també ho podem fer per UDP). 65535 són molts ports; però com que això ho hem de fer només una vegada, val la pena esperar una mica més. El resultat és el següent: tenim oberts els ports 9, 13, 21, 22, 25, 37, 53, 80, 111, 113, 139, 515, 1024, 3128, i 8080. D'aquests ports hem de decidir quins volem que acceptin connexions. (Hi ha informació detallada a Internet sobre cada port).

Ara, doncs, hem d'escriure el firewall personalitzat usant comandaments

d'[iptables](#) ([man iptables](#) té tota la informació). Per entendre-ho s'ha de tenir en compte que al principi s'esborra la configuració anterior; que hi ha una regla per defecte per a cada cadena, que hi ha especificats molts ports a la vegada, i que les connexions ja establertes no es tornen a comprovar. Aquest script és:

```
echo Carregant regles d'iptables...

iptables -F INPUT

iptables -A INPUT -m state --state ESTABLISHED -j ACCEPT

iptables -A INPUT -m state --state NEW -i lo -j ACCEPT

iptables -A INPUT -m state --state NEW -i eth0 -p tcp -m multiport --
dports 21,22,53,80,139,515,3128,8080 -j ACCEPT

iptables -A INPUT -m state --state NEW -i eth0 -p udp -m multiport --
dports 53,137,138,3130 -j ACCEPT

iptables -P INPUT DROP

iptables -F FORWARD

iptables -P FORWARD DROP

iptables -F OUTPUT

iptables -P OUTPUT ACCEPT
```

Tot això només són una sèrie d'ordres que modifiquen paràmetres del kernel, per tant quan es reiniciï i es torni a carregar Linux (el nucli del sistema operatiu) ja no estaran activats. És molt fàcil solucionar això; només hem de gravar totes les comandes a un fitxer, fer-ho executable, posar-ho al directori [/etc/init.d/](#), i crear els enllaços als nivells d'execució que vulguem (a [/etc/rc*.d/](#)) per què s'executin cada vegada que s'encengui el servidor. El posarem als nivells 2, 3 i 5 (els modes multiusuari).

Per acabar, reiniciem i tornarem a escanejar els ports. Altre mètode per

tancar ports és desactivar serveis a [/etc/inetd.conf](#) (ho haurem de fer si no podem posar iptables).

3 CONCLUSIONS

Hem aconseguit un ordinador molt més potent que qualsevol altre de la xarxa, inclús quan només funciona a 100 Mhz. i té 48 Mb. de RAM. Per tant, hem aprofitat molt bé els recursos, sobretot gràcies a Linux. Si haguéssim instal·lat altre sistema operatiu (com Windows) no s'hauria pogut fer quasi res d'això a aquest ordinador.

Finalment, tots els serveis -proxy, DNS, SSH; PHP, CGI, FTP i servidor web amb estadístiques- funcionen perfectament; tenim alguns dubtes en quant a Samba pel problema comentat de la necessitat de crear el mateix usuari a cada màquina Windows, però com que això també passa entre Windows NT i Windows 98, podem pensar que també funciona bé.

El firewall està comprovat en altres ordinadors, però en aquest en concret no es va posar perquè el kernel era un 2.2 ([iptables](#) requereix un 2.4) i la targeta de xarxa necessitava un driver al nou kernel. Per tant, es va haver de tancar cada port a mà. El resultat és gairebé el mateix, però no queda tan professional.

Aquests petits problemes amb el hardware i amb els altres sistemes operatius els podem resoldre llegint tota la documentació inclosa i buscant-ne més per Internet. Després de provar coses durant un temps, aconseguirem deixar el servidor 'perfecte'. Funcionarà gairebé igual que ara, però serà més fàcil d'administrar.

També s'ha de destacar que ho hem fet tot amb software lliure (podem executar [vrms](#) per assegurar-nos) i que no ens hem gastat res de diners, encara que tot

és totalment legal. Si ho haguéssim fet amb altres sistemes operatius propietaris hauríem d'haver pagat quantitats enormes de dòlars o euros i els resultats no serien millors que els que hem aconseguit amb Linux (ben al contrari, encara tindríem més problemes que no podríem resoldre per nosaltres mateixos).

4 ANNEXOS

4.1 Comandes importants

Aquestes són algunes de les ordres necessàries per a utilitzar Linux. Podem veure l'ajuda completa amb l'ordre `man` seguida del nom del programa.

- `ls`: fa un llistat dels arxius. És convenient utilitzar `ls -l` per veure més propietats.
- `pwd`: escriu el nom directori actual
- `cd directori`: entra a un directori. Sense paràmetres, va al directori `$HOME`
- `mkdir/rmdir directori`: crear i esborrar directoris buits
- `cp origen destí`: copia el(s) arxiu(s) d'origen al directori de destí
- `mv origen destí`: mou els arxius, o canvia el nom (de fet és el mateix)
- `rm arxiu`: borra un arxiu. `rm -r` per directoris (cal anar amb compte)
- `touch arxiu`: crea un arxiu en blanc, o canvia la data de modificació si ja existia
- `cat arxiu`: mostra el contingut de l'arxiu
- `less arxiu`: mostra el contingut de l'arxiu fent pauses quan no cap en pantalla
- `file arxiu`: mostra el tipus d'arxiu fixant-se en el seu contingut
- `chmod permisos arxiu`: canvia els permisos d'un arxiu
- `chown/chgrp usuari arxiu`: canvia el propietari o el grup d'un arxiu
- `ln origen destí`: crea un enllaç simbòlic. Pot ser 'enllaç dur' o normal

- `alias ordre="ordre equivalent"`: ens estalvia escriure el mateix
- `echo Text`: escriu el text, normalment per pantalla
- `date`: mostra l'hora del sistema. També serveix per canviar-la
- `uptime`: mostra el temps que porta encès el PC i la càrrega de la CPU
- `ps`: mostra els processos actius. Interessant usar `ps aux` per veure'ls tots
- `top`: monitoritza els processos
- `kill pid_del_procés`: matar un procés o enviar-li una senyal
- `who`: mostra quins usuaris estan connectats
- `id`: mostra informació sobre l'usuari actual
- `su usuari`: permet identificar-se com a un altre usuari (per defecte root)
- `reset`: si veiem codis ASCII estranys a la consola l'hem d'executar per restaurar-la
- `reboot`: reinicia l'ordinador
- `halt`: apaga l'ordinador

Altres programes útils (que potser haurem d'instal·lar nosaltres) són:

- `vim`: versió millorada de `vi`, editor que hi és a tots els Linux
- `nano` o `pico`: editors de fàcil ús, no tant potent com `vi` o `vim`
- `elinks`: versió millorada de `lynx`, navegador d'Internet en mode text
- `wget`: per baixar coses d'Internet
- `ping`: per comprovar si un equip està actiu
- `telnet` i `ssh`: per connectar a un port d'un ordinador
- `ftp`: per connectar a un FTP
- `nmap`: escanejador de ports

- `nc`: netcat, per fer connexions de forma més avançada
- `netstat`: mostra les connexions que manté l'ordinador
- `gcc`: compilador de C/C++, utilitzat per a compilar el codi font dels programes
- `tar` i `gzip`: empaquetador i compressor/descompressor d'arxius.
- `cdrecord`: per gravar imatges de CD creats, per exemple, amb mkisofs
- `dict`: dona la definició (en anglès) de paraules i sigles, per Internet
- `aview`: per veure imatges en codis ASCII mitjançant la llibreria AALib
- `mpg123`: per escoltar MP3 des de consola
- `mplayer`: reproductor de vídeo i DVD. A consola els podem veure en ASCII
- `fsck`: escaneja un sistema d'arxius i busca i corregeix els errors

És molt important saber combinar aquests programes amb 'pipes' (traduït per 'canonades') i redireccionadors. Exemples (del shell `bash`):

```
apt-cache search php | less    # Pipe que envia la sortida del
primer comand al less per poder veure el llistat de paquets per pàgines
```

```
ps axu | grep ssh    # Pipe que mostra els processos actius que
contenen la paraula ssh
```

```
nmap pc > ports_oberts    # Grava el llistat de ports oberts del
host pc en un fitxer
```

```
nc -l -p 6000 <fitxer    # Escolta al port 6000 TCP i respon amb
el contingut del fitxer quan algú es connecta (útil per transmetre fitxers)
```


entre ordinadors sense FTP)

```
cp ~/* carpeta & calendar # Copia els arxius del nostre
directori personal (representat per $HOME o per ~ ) a la carpeta
especificada i, deixant en segon pla aquest procés, ens mostra les
efemèrides del dia (sense haver d'esperar a que acabi de copiar tots els
arxius)
```

```
mv b.bak b; cp b b2 # Fa una acció i després l'altra
```

```
apt-get update && apt-get upgrade # Baixa la nova llista de
paquets i, si no hi ha hagut cap problema, actualitza els nous paquets
```

4.2 Seguretat

El nostre ordinador ja és molt segur; és poc probable que algun alumne el pugui espatllar 'per error' o 'sense voler'. Si volem ser més paranoics (mai és mala idea) podem fer moltíssimes coses per millorar la seguretat del servidor i de totes les dades que hi ha a dins (tant pensant en intrusos com en accidents). Per exemple:

- Posar contrasenya a la BIOS per tal que l'ordinador no es pugui encendre sense escriure-la. Tot i això, per a un hacker que tingués accés físic a l'ordinador seria molt fàcil entrar-hi: només s'ha d'obrir l'ordinador i treure la pila de la placa base durant una estona per esborrar la contrasenya. Naturalment, ens donaríem compte que algú ho ha fet perquè veuríem que ja no la demana.
- Fer que la BIOS no arrenqui des de disquet també és molt interessant per protegir el sistema, però llavors no hem d'oblidar la contrasenya de la BIOS perquè sinó també s'haurà de treure la pila de la placa base.
- Per als més paranoics de tots: la capsa de l'ordinador es pot tancar amb clau (no la clau que ve junt amb el manual d'instruccions, sinó amb alguna de més qualitat). Si ho fem, ja ningú podrà obrir l'ordinador sense haver d'endur-se'l de l'institut. Això, que sembla una mica estrany, es fa als ordinadors importants (Pentàgon, Exèrcit, etc), on fins i tot alguns ordinadors tenen un 'closca' de ferro per resistir els possibles atacs nuclears..
- No treballar com a root per a la feina diària (navegar, escriure e-mails, programar, etc). És per diversos motius: primer; les equivocacions poden

sortir-nos més cares, ja que tenim tot el dret a esborrar qualsevol cosa, encara que sigui important; i segon, algun programa pot tenir un bug que doni accés a un hacker amb els privilegis de l'usuari que l'executa. Si aquest usuari té pocs privilegis, el problema no és tan important.

- No marxar i deixar l'ordinador amb sessions obertes a cap terminal. Podem fer `exit` per a sortir o instal·lar el programa `vlock`, que la bloqueja i demana contrasenya.
- Tenir molta cura dels programes 'setuid root' (amb privilegis de l'usuari root), ja que un bug -per exemple, un buffer overflow- podria elevar els privilegis d'un usuari normal. Podem veure quins tenim al sistema amb `find / -user root -perm -4000 -print`
- Posar bones contrasenyes (sobretot la de root): en elles no ha d'aparèixer el nom de login (ex: per a l'usuari `pepito`, `pepito23` és una mala contrasenya). Han de tenir números i lletres (si pot ser, intercalant majúscules i minúscules) i han de ser tan absurdes que ningú les pugui imaginar mai. Per tant, no han de ser paraules que es puguin trobar a un diccionari. Podem comprovar la qualitat de les contrasenyes intentant crackejar-les amb el programa líder en aquest sector; el John the Ripper.
- Actualitzar a la versió en proves de Debian. La versió 'testing' és la que després es convertirà en estable. No és el mateix que la inestable, que és en la qual treballen els programadors de Debian. Si actualitzem és per tenir les últimes versions de tots els paquets i poder fer actualitzacions més sovint. S'ha de pensar bé abans de donar aquest pas perquè tornar enrere costaria molt més (s'haurien d'actualitzar els paquets amb versions més antigues!). Per fer-ho només cal editar `/etc/apt/sources.list`,

canviar la paraula `stable` per `testing` i fer un `apt-get update && apt-get dist-upgrade -u`

- Si alguna vegada volem deixar que algú es connecti al servidor i trastegi una mica per veure què és Linux i com funciona, en comptes de crear un nou usuari i donar-li accés per SSH podem donar-li l'accés per Telnet (port 23) havent configurat abans `ttysnoop`, un programa que permet veure i participar en les connexions que fan els usuaris a la nostra màquina.
- Si volem que un usuari es pugui connectar habitualment per SSH o Telnet però tenim por que tingui coneixements suficients per fer coses que no volem; el primer serà crear un entorn idèntic a un sistema de Linux al seu directori i fer que en connectar-se es faci un `chroot` a aquell directori per tal que es converteixi en el seu directori arrel ('/'). Així, encara que crackejés `/etc/passwd` o aconseguís convertir-se en root d'altra manera, només ho hauria fet dins del seu arbre de fitxers, ja que el fitxer `/etc/passwd` que ha utilitzat està situat realment a `/home/carpeta/etc/passwd`
- De tant en tant hem de comprovar si realment és necessari tot el que hi ha instal·lat. Un `dpkg -l` mostra tots els paquets que hi ha al sistema. Hem d'evitar tenir coses que no sapiguem què són, i esborrar les que no utilitzarem més.
- Ens podem apuntar a les llistes de correu de Debian per saber si surt alguna vulnerabilitat important.

4.3 Manteniment

El servidor probablement continuarà funcionant a la perfecció durant molt de temps, fins i tot sense haver d'apagar-ho, però amb el temps s'anirà quedant antiquat.

Algunes de les coses que hem de fer, ordenades per importància, són:

- Actualitzar el sistema amb `apt-get update; apt-get upgrade -u` (el `-u` és només per veure el nom dels paquets). Això ho podem fer cada setmana o quan sapiguem que han sortit versions noves d'algun programa que corregeixen vulnerabilitats.
- Mirar els logs per veure errors de funcionament en algun programa.
- A cada nova versió estable de Debian que surti, fer un `apt-get dist-upgrade -u` per actualitzar tota la distribució.
- Veure si hi ha suficient espai al disc dur, i si no n'hi ha, què és el que l'ocupa per esborrar-ho. Si són logs, podem fer que es rotin.
- Comprovar de tant en tant l'estat del disc dur per veure si hi ha sectors erronis amb l'utilitat `fsck`. Compte!: la unitat ha d'estar desmuntada abans de corregir errors (ho podem fer arrencant des de disquet).
- Actualitzar a IPv6 quan sigui necessari. Poc a poc, tots els programes es van adaptant i molts ja accepten direccions en format IPv6 (la versió 6 del protocol IP, ampliat per a abastir a més usuaris d'arreu del món).
- Si volem donar exemple sobre l'ús del software lliure podem instal·lar el `vrms` ('virtual Richard M. Stallman') per què ens avisi si tenim programes no lliures a l'ordinador...

- El software no fallarà, però el hardware es pot fer malbé. Típicament, els equips s'apaguen com a mínim cada nou mesos per deixar descansar la memòria RAM.

4.4 Altres utilitats

Podem treure molt profit a un ordinador amb Linux instal·lat, per molt vell que sigui. Com que estarà molt temps encès sense parar, és ideal per fer feines com:

- Descarregar arxius grans i pàgines web senceres, deixant-les a l'FTP per poder recuperar-los després des de qualsevol ordinador de la xarxa. Ho podem fer amb [wget](#).
- Participar en xarxes d'intercanvi d'arxius; sempre que siguin legals, és clar. [mldonkey](#) és una bona opció.
- Fer càlculs matemàtics complexos. Hi ha molts programes matemàtics molt avançats per a Linux. Un exemple és [Scilab](#). El problema és que 100 Mhz. no és molta potència...
- Treballar en paral·lel amb altres ordinadors (clúster). Moltes empreses i centres d'investigació ho fan: posen a treballar a ordinadors Linux en tasques com la renderització d'imatges 3D, fractals (matemàtiques), càlcul d'òrbites de planetes (astrodinàmica), fluxes turbulents (hidràulica), i moltes coses més.
- Programar: a Linux hi ha les millors eines per fer codi estàndard, i suporta la majoria del llenguatges: C/C++, Perl, Python, MONO::, Tcl/Tk, assembler, etc.
- Programar-hi tasques com, per exemple, enviar e-mails a una llista de correu, modificar dades d'alguna pàgina externa periòdicament, etc.

També podem decidir-nos per instal·lar el mode gràfic i utilitzar programes

professionals com [blender](#) (disseny 3D), [gimp](#) (retoc fotogràfic), [mozilla](#) (navegador web), [openoffice](#) (suite d'aplicacions), [mplayer](#) (visualitzador de vídeos), etc., tot funcionant en un ordinador senzill però aconseguint resultats decents.

4.5 Glossari de termes

Com que per fer servir Linux s'han de tenir molts coneixements teòrics i pràctics, llistarem aquí les paraules que puguin causar confusió. No hi surten noms de marques comercials ni de programes; només dels més importants.

- **Administrador** : persona encarregada de portar una xarxa i els seus elements, entre ells l'ordinador central. A Linux l'administrador sol ser l'usuari `root`.
- **Buffer overflow (desbordament de buffer)** : bug molt comú que consisteix en posar més caràcters dels que caben quan un programa pregunta una dada. Aquests caràcters sobrants poden escriure parts de la memòria i fer que s'executi algun codi en concret (per exemple, el programa `/bin/sh`, per aconseguir una conta al sistema). Es poden explotar de qualsevol forma, fins i tot des de programes senzills com navegadors web.
- **Bug** : problema a un programa que fa que no funcioni bé. Sovint són amenaces per a la seguretat del sistema, i han de ser corregits ràpidament.
- **ChangeLog** : literalment, “registre dels canvis”. És un document que explica què hi ha de nou i què s'ha arreglat en cada versió d'un programa.
- **Dimonis (daemons)** : són programes que s'executen en segon pla en iniciar l'ordinador, i que fan de servidors. Aquesta és la raó per la qual porten la lletra `d` al final del nom: `telnetd`, `sshd`, `ftpd`, `httpd`, etc.
- **DoS** : Denial of Service (denegació de servei): resultat d'un atac a un

ordinador que fa que deixi d'oferir els serveis habituals. Un exemple de DDoS (Distributed DoS) és quan centenars d'ordinadors es connecten a la vegada a un de sol fins que aconseguen que es sobrecarregui i no treballi de forma normal.

- **IDS**: Intrusion Detection Service; programa que detecta les accions no permeses a una xarxa i actua en conseqüència. Per exemple, si veu que s'intenta fer un atac per un port, el tanca durant una estona a la IP de l'atacant i envia un missatge al mòbil de l'administrador. Són programes molt potents.
- **DNS**: Domain Name Server, o servidor de noms de dominis. És l'ordinador encarregat de traduir les direccions d'Internet a direccions IP. A aquesta operació se l'anomena 'resoldre' un nom de host. Per exemple, google.com es resol a 216.239.51.100. Actualment hi ha 14 'root servers' a tot el món; l'últim que es va crear és a Madrid.
- **Exploit (o xplloit)**: programa senzill que aprofita un bug per provocar un error a un programa, fent que l'usuari o hacker augmenti els seus privilegis (normalment la finalitat que es vol aconseguir és l'accés root). Se'n poden trobar molts per Internet, tots en llenguatge C.
- **FAQ**: Frequently Asked Questions (preguntes més freqüents). Ho trobem sobretot en la documentació de programes.
- **Firewall (tallafocs)**: programa que bloqueja els accessos a alguns ports de l'ordinador per evitar les intrusions no desitjades.
- **Flag ('bandera' en castellà, traduït per 'senyalador')**: en programació, variable de tipus booleà (cert/fals, 1/0, sí/no, etc) que s'utilitza per activar o desactivar una opció. Per exemple, trobem flags a la

capçalera dels paquets TCP, o als fitxers especials del directori [/proc/sys](#)

- **FSF**: Free Software Foundation. Fundació iniciada per Richard Stallman creadora del moviment del programari lliure, i que va començar el sistema operatiu GNU.
- **gcc**: GNU C Compiler, el compilador de llenguatge C de la GNU. Encara que sembli que hi han molts compiladors de C i C++, a Linux per defecte hi ha el [gcc](#) (o [cc](#)) i el [g++](#) (o [c++](#)) per a C++. [cc](#) és un link a [gcc](#), i [c++](#) és un link a [g++](#). Per tant, només hi ha el [gcc](#) i el [g++](#)
- **GID**: Group ID, semblant a l'UID però aplicat a grups.
- **GNU**: sigles de GNU's Not Unix. És un projecte de la Free Software Foundation per crear un UNIX lliure.
- **GPL**: General Public License. Una llicència que va crear la GNU per protegir el programari lliure. Debian incorpora únicament software GPL.
- **Hacker**: un hacker és una persona que disfruta amb el seu treball. Els hackers informàtics són experts en la seva matèria, i els interessen els temes avançats i difícils (com per exemple, demostrar que poden entrar a un lloc on teòricament ningú hi pot entrar).
- **Host (o hostname)**: nom d'un equip que l'identifica en la xarxa. Hi ha alguns predefinitos, com [localhost](#), però a [/etc/hosts](#) en podem definir més.
- **HOWTO**: manual d'instruccions sobre com fer alguna cosa. ('How to... ?' en anglès).
- **ISP**: Internet Service Provider. Empresa que ens dóna la connexió a Internet.
- **Kernel**: cor d'un sistema operatiu, que fa d'intermediari entre l'usuari i el

hardware. A aquest treball s'utilitza el kernel Linux, però n'hi ha d'altres com el Hurd (encara en una fase molt primitiva).

- **Keylogger** : tal com diu la paraula, és un programa que queda resident en segon pla i grava tot el que s'escriu amb el teclat. El millor mètode per interceptar contrasenyes; l'inconvenient és que el hacker ha de tenir accés a la màquina perquè ha de tornar per revisar el log.
- **Lilo** : gestor d'arrencada dels diferents sistemes operatius instal·lats. Podem fer, per exemple, que en encendre l'ordinador ens preguntin si volem entrar a Linux 2.4.18, Linux 2.2.20, BSD, BeOS, o qualsevol altre kernel o sistema. Tot això es configura a `/etc/lilo.conf` i executant després `/sbin/lilo` per aplicar els canvis.
- **Linux** : nom del kernel creat originàriament per Linus Torvalds. "Linux" és el nom del kernel, i no del sistema operatiu, ja que aquest es diu GNU. Per això s'hauria de parlar de GNU/Linux en comptes de Linux.
- **Locales** : variables d'entorn que controlen l'idioma dels programes. Si no es defineixen, tot sortirà en anglès encara que hi existeixen versions en altres idiomes.
- **Muntar i desmuntar** : procés pel qual un sistema d'arxius es fa o es deixa de fer accessible sota un directori del sistema d'arxius actual. Per exemple, podem muntar un disquet a `/mnt/floppy`. S'utilitzen les ordres `mount` i `umount`.
- **Log** : registre. A Linux es registren moltes accions i incidents, normalment al directori `/var/log`. Es guarda informació sobre els accessos a cada servidor i les operacions realitzades, els ingressos i sortides de cada usuari (especialment root), i els problemes que hi ha hagut. Un hacker ha

d'aconseguir esborrar els logs si vol que ningú sospiti.

- **Loopback** (`lo`): interfaç de xarxa que representa el nostre ordinador. Se l'assigna la IP 127.0.0.1 (també la 0.0.0.0) i s'utilitza a serveis interns.
- **PATH**: la variable `$PATH` conté els directoris als que es buscarà cada programa que executem abans de comprovar si està al directori actual. Per exemple, quan fem un `ls` el que realment estem executant és `/bin/ls`, perquè el directori `/bin` està al PATH. Un hacker pot canviar el PATH per a fer que programes mal fets executin comandes falses.
- **PID**: Process ID, o número que té cada procés en execució. El podem veure amb l'ordre `ps`. És necessari per poder matar (finalitzar) un procés.
- **Proxy**: programa instal·lat a un servidor que s'encarrega de captar les peticions HTTP (pàgines web), baixar-les d'Internet, i servir-les als usuaris. Per evitar haver de baixar la mateixa moltes vegades, la pot guardar en caché. També pot denegar l'accés a algunes pàgines.
- **Root**: l'administrador d'un sistema Linux. Se'l coneix com a superusuari, té UID 0, i ho pot fer absolutament tot (ningú té més privilegis).
- **Setuid**: quan es diu que un programa està amb el bit setuid activat, s'executa amb els privilegis del seu propietari. Per exemple, el programa `passwd`, que un usuari normal ha de poder executar per canviar la seva contrasenya, necessàriament ha d'accedir a `/etc/passwd` o `/etc/shadow`, on només hi pot escriure root. Per tant, l'única solució és el 'setuid root'. S'ha de tenir molta cura amb aquests programes; si es troba un buffer overflow es poden aconseguir privilegis de root. Per a més informació: el propietari el podem canviar amb `chown usuari arxiu`, el setuid el podem posar i treure amb `chmod a+s arxiu`, i podem veure si un programa ho

està amb un `ls -l` (veurem una `s` al lloc on normalment hi apareixen les `x`).

- **Shell**: intèrpret de comandes de Linux. Hi ha molts shells, la majoria molt semblants, però el més utilitzat és `bash`. Altres, són `sh`, `csch`, `tcsh`, `bsh`, `ash`, `ksh`, `rbash`, ... Cada usuari del sistema pot escollir el seu shell preferit; només cal modificar la entrada apropiada a l' `/etc/passwd`
- **Superusuari**: l'usuari amb més poder del sistema. S'anomena `root`, però es poden crear més (només cal que tinguin UID 0).
- **UID**: User ID, o número d'identificació de cada usuari del sistema. S'especifica al tercer paràmetre de cada registre de `/etc/passwd` NOTA: qui té UID 0 és superusuari (normalment anomenat `root`).
- **X**: l'X Windows System (implementat pel programa XFree86) és un servidor gràfic: un servidor com tots els altres al qual els programes es connecten de forma local (també pot ser remota) i mostren una interfície amb botons, menús, ratolí, etc. Al nostre ordinador no l'hem instal·lat, perquè no ens fa falta (el mode gràfic és opcional, no com en altres sistemes operatius).

4.6 Llicència FDL de la GNU

Aquest treball està escrit sota la Free Documentation License de la GNU. És semblant a la GPL, que fa que els programes de codi obert ho hagin de continuar sent sempre, però aplicat a documents. Per tant, qualsevol pot aprofitar tot el meu treball per a millorar-ho, però sempre haurà de fer que aquest continuï sent públic, i, és clar, no atribuir-s'ho a un mateix (ha de conservar el nom de l'autor, Daniel Clemente Laboreo).

La llicència FDL, necessària en el treball però que no incloc per raons d'espai, és disponible (en català) a <http://www.softcatala.org/licencies/fdl-ca.html>

La llicència, en anglès, és <http://www.gnu.org/licenses/fdl.html>

Aquest treball és una versió molt reduïda de l'original. El treball original es pot descarregar lliurement de la meva pàgina, a la direcció <http://www.danielclemente.com/servidor> i és disponible en castellà i català.

5 BIBLIOGRAFIA

- General

Serveis de xarxa amb GNU/Linux, Jordi Bort i Marc Guri. 2002.

<http://www.xtec.es/formacio/curstele/d70/>

Securing Debian Manual, Javier Fernández-Sanguino Peña. 2002.

<http://www.debian.org/doc/manuals/securing-debian-howto>

<http://www.debian.org>

- Instal·lació

The very verbose Debian 3.0 Installation Walkthrough, Clinton De Young.

2002. http://www.osnews.com/story.php?news_id=2016

<http://www.distrowatch.com>

- Servidor web Apache

<http://www.apache.org>

- Servidor FTP pure-ftpd

<http://www.pure-ftpd.org>

- Proxy-caché Squid

<http://www.squid-cache.org>

- PHP

Adding PHP to Apache on Linux, Ken Coar. 1999.

<http://www.linuxplanet.com/linuxplanet/tutorials/1374/1/>

<http://www.php.net>

- CGI

CGI Programming on the World Wide Web, Shishir Gundavaram. 1996.

O'Reilly. 1ª edició; disponible online a <http://www.oreilly.com/openbook/cgi/>

- DNS

DNS HOWTO, Nicolai Langfeldt, Jamie Norrish i altres. V 9.0, 2001

<http://www.linux.org/docs/ldp/howto/DNS-HOWTO.html>

- Samba

Using Samba, Robert Eckstein, David Collier-Brown, Peter Kelly. 1999.

O'Reilly. Disponible online a http://us2.samba.org/samba/oreilly/using_samba

- Firewall

Un firewall en 10' pero sabiendo lo que se hace, Guillermo Pereyra Irujo.

Gener 2003.

