

¿Cómo crear un nodo con Linux?



Pablo Iranzo Gómez (Pablo.Iranzo@uv.es)

19 de junio de 2003

Índice general

1. Introducción	2
2. Hardware	3
3. Sistema Operativo y Paquetes	5
4. Red	17
5. Masquerading	23
6. Túneles	34
7. Consejos Finales	35
8. Ficheros de configuración	37
8.1. Squid	37
8.2. PPTPD (para las VPN entrantes)	92
8.2.1. /etc/ppp/pptpd-options:	92
8.2.2. /etc/ppp/chap-secrets:	93
9. 9 Créditos	94

Capítulo 1

Introducción

Un nodo nos permitirá unir a personas conectadas desde sus tarjetas inalámbricas. Un nodo tiene que ser el punto de unión entre distintos clientes y a su vez enlazar con otros nodos para así crear una red.

En el caso de las redes inalámbricas, existen varias aproximaciones.

Un nodo puede ser un aparato denominado Punto de Acceso o por sus siglas en inglés AP (Access Point) que contiene en su interior un interfaz wireless y una conexión Ethernet RJ45 para enlazarlo o bien con un equipo o con una red.

Un aparato que realice dichas funciones no es especialmente asequible además de que está bastante limitado...

La solución más aceptada es utilizar un PC para que haga de nodo, teniendo así muchísima más flexibilidad a la hora de configurarlo (más problemático también, no es "enchufar y listo" como con un AP de hardware).

Las tarjetas a la venta en el mercado tienen tres modos de operación:

1. Ad-hoc: Es el modo estándar, en este tipo de modo de operación, es el equivalente a las redes con Windows en las que se trabaja de igual a igual, todos son clientes y servidores.
2. Managed: un servidor independiente (un AP) es el lugar al que se conectan todas las demás tarjetas inalámbricas, el AP gestiona todas las conexiones, enrutadas, etc.
3. Master: Es el modo en el que trabaja el AP del punto anterior para que las tarjetas puedan trabajar como "Managed".

En redes con Windows sólo es posible trabajar en los dos primeros modos, el Ad-Hoc y el Managed, para ello, como se ve, es necesario un AP por hardware que trabaje como Master al que las tarjetas que trabajan en Managed se puedan conectar.

Y este es el punto en el que Linux marca la diferencia... con Linux, existe un controlador (hostap) para las tarjetas basadas en Prism2 (p.ej. las Conceptronic). Con Linux y con esos controladores, es posible poner en modo Master las tarjetas de red (generalmente PCMCIA con un adaptador PCI para equipos de sobremesa) y de ese modo, hacer un AP por software.

Quisera expresar mi agradecimiento a Ghfjdksl de #wireless, a Jorti de #guadawireless y a Hilario y a Hawkmoon de #valenciawireless por la ayuda prestada para montar este nodo. Saludos también al resto de gente de #valenciawireless por sus larguísimas tertulias ;P

Capítulo 2

Hardware

Para la creación del nodo, necesitaremos unos componentes de hardware tanto de software.

En la creación del nodo de La Creu Wireless, el hardware es el siguiente:

- Placa AT Pentium 120
- 500 Mb HD
- 420 Mb HD (porque lo tenía a mano, pero posiblemente lo quite... lo que menos quiero es que el ordenador haga más ruido...)
- 64 mb Ram (iba a tener 16, pero a última hora conseguí más...)
- Tarjeta gráfica ATI Mach 64 Pro Turbo PCI
- Ethernet ISA P'n'P Intel PRO 10/10+
- Ethernet PCI basada en Realtek 8029 (10 mbps)
- Ethernet PCI basada en Realtek 8139 (10/100)
- Ethernet ISA P'n'P Compatible con NE 2000
- Tarjeta wireless todavía no la tengo, utilizo la ISA P'n'P compatible con NE2000 como si fuese la wireless (A la que en la actualidad tengo conectado un USR2450 en modo bridge con LinuxAP)
- Tarjeta de sonido SB 16 P'n'P (no se para qué porque no la tengo conectada ni nada, pero el caso es que como la tenía....)
- CD-ROM (instalé desde disco duro porque no tenía cd-rom en ese momento... pero bueno.. ahora sí ;))
- RJ45 (latiguillos), uno directo para conectar al módem y a la eth0 y luego tres cruzados para conectar a cada uno de los pc's sin necesidad de hub... si tienes hub, pues todos directos y listo...

Se aceptan regalos ;) (Portátiles incluidos... ;))

Volviendo a la realidad... memoria le sobra por un tubo, pero bueno, puestos a ponerle, pues se le pone más :)

Disco duro me gustaría ponerle más, ya que como lo tengo que tener encendido todo el día para que sea un nodo "fijo", pues aprovechar y dejar las largas descargas bajándose en ese ordenador.

Lo de las tarjetas de red puede asustar al principio... pero bueno, tiene una clara explicación... en casa tengo dos ordenadores y ahora con el nodo tres, la conexión a Internet requiere otra tarjeta de red y no me permite compartirla directamente conectándola a un HUB, así que me parecía algo tonto el tener que gastarme dinero en un HUB cuando las tarjetas de red ya las tenía...

Así que el montaje de la falla es el siguiente:

- Intel Pro ISA 10/10+: eth0 conectada a mi conexión a Internet
- Realtek 8139 (10/100): eth1 conectada a mi equipo principal (también con otra tarjeta 10/100, así que red local a 100 mbps ;))
- Realtek 8029: eth2 conectada al otro pc que tengo en casa (también con una ethernet a 10 mbps)
- Isa P'n'P NE2000: eth3 hace las labores de wireless hasta que tenga una de verdad (la utilizo para conectar otro equipo y hacer las pruebas como si estuviese conectado por wireless), en su momento será utilizada para conexiones "temporales" por cable, siendo entonces eth4 la wireless "real".

Realmente con un HUB podría quitar dos tarjetas de red del ordenador (ya que el ordenador principal y el secundario irían conectados al hub y no serían necesarias las tarjetas correspondientes y sólo utilizaría la otra que quedase para Internet y la otra para conectar al HUB (si mi conexión me permitiese hacerlo, hasta podría conectar mi módem al HUB y ahorrarme otra tarjeta.. pero bueno, el caso es que no se puede...)

Como se puede ver, no digo nada de ratón ni de monitor... el caso es que realmente no es necesario porque para hacer la instalación inicial se hará conectándolo a otro monitor, pero una vez puesto en marcha no hará falta para nada, de hecho hasta el teclado sobraría, pero como he dicho que es un P120 con AT, en los casos en los que tengo que apagar o reiniciar la máquina necesito algo que no implique un reset total (o sea, necesito tres teclas: CTRL-ALT-SUPR).

El teclado que tengo es uno de 84 teclas que será como un teclado normal pero sin el bloque numérico y sin el bloque de av-pag, etc de tamaño, tiene sólo dos leds, Bloq may y bloq despl, pero lo dicho... sobra :)

Ahora el ordenador es un P200 MMX con 128 Mb de RAM, las tarjetas son una Intel Pro ISA 10/10+, una SMC 100, 3Com 3c509 y NE2000, discos tiene uno de 10 Gb, y otro de 20, configurados como:

- 128 Mb Swap
- 10 Mb boot
- 3 Gb sistema, usuarios
- Resto = 26 Gb para descargas diversas... (pero con ánimo educativo, no os vayais a pensar...)

Ahora vamos a la configuración del software...

Capítulo 3

Sistema Operativo y Paquetes

El primer paso previo a todos, es que una vez configurado el hardware para que no se queje (IRQ's, puertos IO, etc) (si la placa es realmente P'n'P no habrá problemas pero la mía aunque es P'n'P dio algunos problemas así que... ajo y agua...) Si has de comprar tarjetas y puedes, cómpralas PCI y mejor todavía si compras un HUB porque así no tienes que tener el ordenador relleno en todas sus ranuras con ethernet's ;).

Lo primero a configurar entonces es la BIOS.

Como cada BIOS es de un papá y de una mamá distintos (aunque a veces la mamá sea siempre Award o Ami), daré la explicación general y que cada uno se las apañe con el manual de su placa base para hacer lo que digo...

PD: Si no sabes lo que es la BIOS, olvídate de seguir... esto no está hecho para tí... Mucho mejor que veas esto.

Configura las tarjetas en el orden que quieras, pero ten bien claro cuál será cada una (si son todas iguales pues será más problemático, pero no por ello imposible).

Primero pon el ordenador en fecha y hora, y en la opción que pone "Halt on" ponle que nada, es decir, que bajo ningún concepto el ordenador no arranque (por ejemplo por falta de teclado, falta de monitor, etc porque al fin y al cabo luego no lo tendrá...)

En la BIOS dile que de PNP su tía... le dices que el sistema operativo no es P'N'P y que no se maree (puedes probar diciendo que sí, no debería ser muy tortuoso y si lo es, siempre lo puedes cambiar otra vez ;))

Como dispositivo de arranque se le pondrá primero la disquetera, pero en el momento en que todo esté configurado, se pondrá fijo siempre el disco duro.

Si la placa es ATX, dile que tras un corte de luz, que se encienda sola... así evitarás los tiempos de apagón del nodo al mínimo.

Salva todos los cambios y reiniciará el ordenador...

En mi caso, como no tenía en un primer momento ni disquetera ni cd-rom, aproveché que el disco duro contenía un Windows 95 y bajé la imagen de CD-ROM de instalación de Internet de Debian Potato, la convertí del .ISO a carpetas y directorios normales y la guardé por la red en el segundo HD del ordenador.

La imagen iso son unos 30 Mb y la descargué de la página de debian. (woody_netinst-20020626-i386.iso)

La decisión de tomar "Debian" como sistema a instalar fue sencilla (lo de que tenía que ser Linux estaba cantado por lo explicado en la introducción):

- RedHat Linux: Está muy bien, de hecho es con la que más experiencia tengo, la tengo funcionando desde el año 96 en un servidor y me va genial, poquitos pro-

blemas excepto actualizaciones, etc (aunque con las nuevas herramientas es muy fácil). Las actualizaciones, en el momento de lanzarlas, si hay muchos clientes de pago, pues a los que no pagamos, pues ajo y agua y a esperarse a que no esté tan saturado... el formato de redhat, el RPM es el estándar más extendido. (Se puede bajar de internet).

- SuSE Linux: Está muy bien, si compras la distribución, tienes muchos cd's con programas, manuales, etc Bastante bien adaptada al Español (je, je), etc... Pero no se puede bajar la versión completa de internet porque tiene programas con licencias de distribución que no lo permiten, sólo si se compra. Utiliza el RPM que es un punto a favor, pero en contra es que no es compatible al 100 % con las estructuras de directorios y funcionamientos de RedHat, así que no hay forma de distinguir si ese RPM funcionará bien o no con SuSE. El peor inconveniente es la gran cantidad de recursos que necesita, tanto de RAM para la instalación como para el almacenamiento en disco (la mínima son unos 500 mb (pero mínima mínima mínima...)). La ventaja es que la configuración es toda en modo gráfico, con una autodetección genial, etc. Tenía experiencia con ella tanto de gastarla en casa como distribución linux, en la uni en un servidor como de haber colaborado en su traducción de manuales y de software de configuración.
- Debian: No tenía ni puta idea de ella, una vez que la intenté instalar casi me da algo con el programa de selección de paquetes a instalar (dselect de la Potato). La instalación es en modo texto. Tiene un sistema de actualización/instalación bastante bueno en modo texto. la decisión fue usar Debian... de paso que aprendía a gastarla, al ser en modo texto, tenía el aliciente de que facilita su administración remota (recordemos que el ordenador una vez instalado mínimamente no iba a tener ni teclado ni ratón ni monitor ni naa de na... excepto el cable de la luz y cuatro cables RJ45 para las tarjetas de red :))

Vale, como ya he comentado, en el nodo, en el momento de su creación no tenía ni disquetera ni CD-ROM, así que se copiaron los archivos a una instalación de Windows 95, se reinicia en modo-msdos y una vez en el, se cambia al directorio donde se extrajo la imagen de cd y se ejecuta el programa de instalación...

Si tienes grabadora y cd-rom en el nodo pues lo más sencillo es bajar la iso, grabarla a un CD y arrancar desde el CD-ROM (en la bios se puede configurar esa opción).

Como ya he dicho, la imagen de instalación es una NETINST, es decir, 30 mb mínimos que se utilizan para una vez respondidas unas preguntas copiar un sistema básico que se puede conectar a Internet y seguir bajando el resto de Internet... (Tengo una conexión a 128 kbps, así que que nadie me diga que es una locura porque es viable...)

Cuando yo hice la instalación, la netinst más habitual era la potato, pero hace nada que pasaron a Woody (3.0), el caso es que más o menos son parecidas y de hecho una vez instalé lo mínimo hice el cambio a Woody usando el programa que lleva para hacerlo...

No voy a explicar la instalación de Debian porque esto habla de configurar un nodo, no de instalar Linux, pero bueno, básicamente consistió en particionar el HD, crear una partición swap de 40 Mb (ya dije que mi disco duro era pequeñito..) y una de datos de 472 Mb como ext3

Es conveniente pensar un nombre para el nodo...la norma habitual, al menos en universidades, etc es ponerle nombres de estrellas o de científicos, pero bueno... Cada uno que le ponga el que le de la gana...

Se configura el teclado, la tarjeta de red (como tengo Internet por tarjeta de cable fue tan sencillo como decirle que cargase los módulos de las tarjetas de red (en el caso de las ISA indicando también la IO) y luego en la que estaba conectada a Internet, decirle que hiciese configuración automática por DHCP y ya estaba la conexión funcionando...

En la selección de paquetes puse lo mínimo mínimo, así que al ratito y tras un rearranque, me pidió asignar contraseñas y crear un usuario (no es recomendable estar siempre como administrador para trabajar con Linux...)

Vale, llegados a este punto, tenemos ya el Linux instalado y tenemos delante la consola para iniciar sesión en el sistema.

Vamos a necesitar varios programas además de los básicos para trabajar con el nodo... si nos gusta Linux, pues nos gustará instalar alguno más, pero bueno... por un lado están los básicos y los recomendados...

Como lo que estoy haciendo es explicar la configuración de lo que yo hice con mi ordenador, pues allá vamos:

Para la wireless hace falta el wavemon que es un monitor de estado de la tarjeta y el zebra para enrutar dinámicamente.

En mi caso, ya que ese ordenador iba a hacer de medio "servidor" en mi casa, pues me interesó ponerle:

- samba: para compartir archivos e impresoras en un formato compatible con Windows y así poder luego bajar archivos de un ordenador a otro (del principal al nodo) mediante el entorno de red
- pptpd: para permitir conexiones VPN, permite que desde internet (o en este caso, desde la wireless), se pueda conectar con el nodo, y se asigne otra dirección IP que permita hacer otras cosas que de normal no se podría... En mi caso, tengo límite de descarga en Internet y aunque por el momento no lo apliquen no quita que pudieran hacerlo... causa pues de que no comparta mi acceso a Internet a no ser que conozca directamente al equipo que se conecta... es decir, una vez conectado mediante la wireless, hará una conexión VPN desde su Windows o su Linux al nodo y mediante un login y un pass tendrá acceso a Internet a través del nodo, que de otra forma no tendría...) (un nodo no tiene que necesariamente tener acceso a Internet, pero es recomendable para unir los nodos y facilitar la integración en la red, etc)
- webmin: es un programa hecho por caldera y que funciona en multitud de plataformas y distribuciones, permite muy fácilmente configurar desde un navegador muchísimas cosas del sistema. No digo que vaya a reemplazar a la configuración en modo consola, pero para muchas cosas es muchísimo más fácil hacerlo desde navegador que tener que entrar, etc...
- Otros paquetes instalados (muchos de ellos se instalan solitos como dependencias a los ya instalados...):

```
Desired=Unknown/Install/Remove/Purge/Hold
| Estado=No/Instalado/Config-files/Unpacked/Failed-config/Half-installed
|/ Err?=(none)/Hold/Reinst-required/X=both-problems (Status,Err: mayúsc.=malo)
||/ Nombre          Versión          Descripción
+++-----+-----+-----+
rc  aalib1            1.4p5-17        ascii art library
ii  adduser            3.50            Add and remove users and groups
```

ii	adzapper	0.20030505-1	squid_redirect advertisement zapper
ii	afio	2.4.7.9beta4-1	Archive file manipulation program.
ii	alien	8.30	install non-native packages with dpkg
ii	alsa-base	0.9.3c-1	ALSA sound driver common files
ii	alsa-modules-2	0.9.3c+1	Advanced Linux Sound Architecture (drivers)
ii	alsa-utils	0.9.3-1	Advanced Linux Sound Architecture (utils)
ii	apache	1.3.27.0-1	Versatile, high-performance HTTP server
ii	apache-common	1.3.27.0-1	Support files for all Apache web servers
ii	apt	0.5.4	Advanced front-end for dpkg
ii	apt-utils	0.5.4	APT utility programs
ii	aptitude	0.2.11.1-3	curses-based apt frontend
ii	arpwatch	2.1a11-6.1	Ethernet/FDDI station activity monitor.
ii	ash	0.4.17	Compatibility package for the Debian Almquist
ii	at	3.1.8-11	Delayed job execution and batch processing
ii	aumix	2.8-3	Simple text-based mixer control program
ii	base-config	1.64	Debian base configuration package
ii	base-files	3.0.8	Debian base system miscellaneous files
ii	base-passwd	3.5.3	Debian base system master password and group
ii	bash	2.05b-7	The GNU Bourne Again SHell
ii	bc	1.06-11	The GNU bc arbitrary precision calculator la
ii	bind	8.3.4-4	Internet Domain Name Server
ii	binutils	2.14.90.0.4-0.	The GNU assembler, linker and binary utiliti
ii	bittorrent	3.2.1b-3	Scatter-gather network file transfer
ii	bsdmainutils	5.20030320-1	collection of more utilities from FreeBSD
ii	bsdutils	2.11z-1	Basic utilities from 4.4BSD-Lite.
ii	buffer	1.19-3	Buffering/reblocking program for tape backup
ii	bzip2	1.0.2-1	A high-quality block-sorting file compressor
ii	calamaris	2.57-1	Log analyzer for Squid or Oops proxy log fil
ii	cdcd	0.6.5-3	command line or console based CD player
ii	cdrecord	2.0+a15-1	A command line CD writing tool
ii	centericq	4.9.4-1	A text-mode multi-protocol instant messenger
ii	centericq-comm	4.9.4-1	A text-mode multi-protocol instant messenger
ii	checksecurity	1.0.1	basic system security checks
ii	console-common	0.7.22	Basic infrastructure for text console config
ii	console-data	2002.12.04dbs-	Keymaps, fonts, charset maps, fallback table
ii	console-tools	0.2.3dbs-32	Linux console and font utilities
ii	console-tools-	0.2.3dbs-32	Shared libraries for Linux console and font
ii	coreutils	5.0-3	The GNU core utilities
ii	cpio	2.5-1	GNU cpio -- a program to manage archives of
ii	cramfsprogs	1.1-4	Tools for CramFs (Compressed ROM File System
ii	cron	3.0p11-74	management of regular background processing
ii	dash	0.4.17	The Debian Almquist Shell
ii	debconf	1.2.35	Debian configuration management system
ii	debconf-utils	1.2.35	debconf utilities
ii	debhelper	4.1.45	helper programs for debian/rules
ii	debianutils	2.5.2	Miscellaneous utilities specific to Debian
ii	devfsd	1.3.25-13	Daemon for the device filesystem
ii	dhcp-client	2.0p15-15	DHCP Client
ii	dhcp3-common	3.0+3.0.1rc11-	Common files used by all the dhcp3* packages
ii	dhcp3-server	3.0+3.0.1rc11-	DHCP server for automatic IP address assignm

ii	dialog	0.9b-20030308-	Displays user-friendly dialog boxes from she
ii	dictionaries-c	0.9.51	Common utilities for spelling dictionary too
ii	diff	2.8.1-2	File comparison utilities
ii	dpkg	1.10.10	Package maintenance system for Debian
ii	dpkg-awk	1.0.1	Gawk script to parse /var/lib/dpkg/{status,a
ii	dpkg-dev	1.10.10	Package building tools for Debian
ii	dselect	1.10.10	a user tool to manage Debian packages
ii	e2fsprogs	1.33+1.34-WIP-	The EXT2 file system utilities and libraries
ii	ed	0.2-20	The classic unix line editor
ii	eject	2.0.13-1	ejects CDs and operates CD-Changers under Li
ii	esound-common	0.2.29-1	Enlightened Sound Daemon - Common files
ii	ethereal-commo	0.9.12-2	Network traffic analyser (common files)
ii	etherwake	1.06-2	A little tool to send magic Wake-on-LAN pack
ii	euro-support	1.19	Use euro character in your Debian system
ii	euro-support-c	1.19	Use euro character in your console environme
ii	exim	3.36-6	An MTA (Mail Transport Agent)
ii	fdutils	5.4-20021102-1	Linux floppy utilities
ii	file	4.02-4	Determines file type using "magic" numbers
ii	fileutils	5.0-3	The GNU file management utilities (transitio
ii	findutils	4.1.7-2.1	utilities for finding files--find, xargs, an
ii	fmirror	0.8.4-11	memory efficient ftp mirror program
ii	fontconfig	2.2.0-2	generic font configuration library
ii	ftp	0.17-10	The FTP client.
ii	gawk	3.1.2-4	GNU awk, a pattern scanning and processing 1
ii	gcc-3.0-base	3.0.4-14	The GNU Compiler Collection (base package)
ii	gcc-3.2-base	3.2.3-0pre9	The GNU Compiler Collection (base package)
ii	gcc-3.3-base	3.3-2	The GNU Compiler Collection (base package)
ii	gettext	0.11.5-1	GNU Internationalization utilities
ii	gettext-base	0.11.5-1	GNU Internationalization utilities for the b
ii	grep	2.5.1-5	GNU grep, egrep and fgrep
ii	groff-base	1.18.1-9	GNU troff text-formatting system (base syste
ii	gzip	1.3.5-6	The GNU compression utility
ii	hdparm	5.3-0.1	Tune hard disk parameters for high performan
ii	hostname	2.10	A utility to set/show the host name or domai
ii	html2text	1.3.1-1	An advanced HTML to text converter.
ii	ifupdown	0.6.4-4.4	High level tools to configure network interf
ii	info	4.3-1	Standalone GNU Info documentation browser
ii	initrd-tools	0.1.47	Tools to generate an initrd image.
ii	ipchains	1.3.10-15	Network firewalling for Linux 2.2.x
ii	iptables	1.2.8-1	IP packet filter administration tools for 2.
ii	isapnptools	1.26-1	ISA Plug-And-Play configuration utilities.
ii	jigdo-file	0.7.0-2	Download Debian CD images from any Debian mi
ii	joe	2.8-21	user friendly full screen text editor
ii	john	1.6-17	An active password cracking tool
ii	kernel-image-2	2.4.20-8	Linux kernel image for version 2.4.20 on Pen
ii	klogd	1.4.1-11	Kernel Logging Daemon
ii	language-env	0.43	simple configuration tool for native languag
ii	less	381-2	A file pager program, similar to more(1)
ii	libacl1	2.2.9-1	Access control list shared library
ii	libao2	0.8.3-1	Cross Platform Audio Output Library

ii	libapache-mod-	1.6.2-6	Frontpage support for apache
ii	libapache-mod-	1.27-4	Integration of perl with the Apache web serv
ii	libapache-mod-	2.8.14-3	Strong cryptography (HTTPS support) for Apac
ii	libartsflow-da	1.1.0.k1-2	aRts Sound system artsflow (data files)
ii	libasound2	0.9.3-2	Advanced Linux Sound Architecture (libraries
ii	libattr1	2.4.4-1	Extended attribute shared library
ii	libaudiofile0	0.2.3-4	The Audiofile Library
ii	libauthen-pam-	0.13-3	This module provides a Perl interface to the
ii	libblkid1	1.33+1.34-WIP-	Block device id library
ii	libbz2-1.0	1.0.2-1	A high-quality block-sorting file compressor
ii	libc-client200	2002b.debian-5	UW c-client library for mail protocols
ii	libc-client200	2003debian0.03	UW c-client library for mail protocols
ii	libc6	2.3.1-16	GNU C Library: Shared libraries and Timezone
ii	libcap1	1.10-12	support for getting/setting POSIX.1e capabil
ii	libcdaudio0	0.99.9-2	library for controlling a CD-ROM when playin
rc	libcdparanoia0	3a9.8-7	Shared libraries for cdparanoia (runtime lib
ii	libcupsys2	1.1.18-2	Common UNIX Printing System(tm) - libs
ii	libcurl2	7.10.5-1	Multi-protocol file transfer library, now wi
ii	libdb1-compat	2.1.3-7	The Berkeley database routines [glibc 2.0/2.
ii	libdb2	2.7.7.0-8	The Berkeley database routines (run-time fil
ii	libdb3	3.2.9-17	Berkeley v3 Database Libraries [runtime]
ii	libdb3-util	3.2.9-17	Berkeley v3 Database Utilities
ii	libdb4.0	4.0.14-1.2	Berkeley v4.0 Database Libraries [runtime]
ii	libdb4.1	4.1.25-4	Berkeley v4.1 Database Libraries [runtime]
ii	libdbd-mysql-p	2.1026-3	A Perl5 database interface to the MySQL data
ii	libdbi-perl	1.35-1	The Perl5 Database Interface by Tim Bunce
ii	libdc1394-9	0.9.0-3	high level programming interface for IEEE139
ii	libdevel-symdu	2.03-3	Perl module for inspecting perl's symbol tab
ii	libdigest-hmac	1.01-1	create standard message integrity checks
ii	libdigest-nils	0.06-1	Nilsimsa message digest algorithm
ii	libdigest-sha1	2.01-0.1	NIST SHA-1 message digest algorithm
ii	libdirectfb8	0.9.8-2	frame buffer graphics library (for 2.4+ kern
rc	libdvnav1	0.1.3-1	The DVD navigation library
rc	libdvread2	0.9.3-2	Simple foundation for reading DVDs.
ii	libesd0	0.2.29-1	Enlightened Sound Daemon - Shared libraries
ii	libexpat1	1.95.6-4	XML parsing C library - runtime library
rc	libfaad2-0	1.1-sarge0.0	Freeware Advanced Audio Decoder - runtime fi
ii	libfontconfig1	2.2.0-2	generic font configuration library (shared 1
ii	libfreetype6	2.1.4-2	FreeType 2 font engine, shared library files
ii	libgcc1	3.3-2	GCC support library
ii	libgcrypt1	1.1.12-3	LGPL Crypto library - runtime library
ii	libgd1	1.8.4-33	GD Graphics Library (transitional package)
ii	libgd1-xpm	1.8.4-33	GD Graphics Library (old version)
ii	libgd2-noxpm	2.0.12-2	GD Graphics Library version 2 (without XPM s
ii	libgdbmg1	1.7.3-27.1	GNU dbm database routines (runtime version).
rc	libggi2	2.0.2-1	General Graphics Interface runtime libraries
rc	libgii0	0.8.2-1	General Input Interface runtime libraries
ii	libglib1.2	1.2.10-9	The GLib library of C routines
ii	libglib2.0-0	2.2.1-3	The GLib library of C routines
ii	libgnutls5	0.8.6-4	GNU TLS library - runtime library

ii	libgpmg1	1.19.6-12.1	General Purpose Mouse Library [libc6]
ii	libgsm1	1.0.10-11.2	Shared libraries for GSM speech compressor.
rc	libgtk1.2	1.2.10-14	The GIMP Toolkit set of widgets for X
ii	libgtk1.2-comm	1.2.10-16	Common files for the GTK+ library
ii	libhtml-parser	3.26-0.1	A collection of modules that parse HTML text
ii	libhtml-tagset	3.03-1	Data tables pertaining to HTML
ii	libhtml-tree-p	3.17-0.1	represent and create HTML syntax trees
ii	libid3tag0	0.14.2b-8	ID3 tag reading library from the MAD project
ii	libident	0.22-2	simple RFC1413 client library - runtime
ii	libio-stringy-	2.108-1	Perl5 modules for IO from scalars and arrays
ii	libjpeg62	6b-7	The Independent JPEG Group's JPEG runtime li
ii	libkrb53	1.2.7-4	MIT Kerberos runtime libraries
ii	libldap2	2.0.27-4	OpenLDAP libraries (without TLS support).
rc	liblircclient0	0.6.6-5	LIRC client library
ii	liblockfile1	1.03	NFS-safe locking library, includes dotlockfi
ii	liblzol	1.07-1	A real-time data compression library.
ii	libmad0	0.14.2b-8	MPEG audio decoder library
ii	libmagic1	4.02-4	File type determination library using "magic
ii	libmail-audit-	2.1-1	Perl library for creating easy mail filters
ii	libmailtools-p	1.58-1	Manipulate email in perl programs
ii	libmcp-data	1.1.0.k1-2	aRts Multimedia COmmunications Protocol data
ii	libmd5-perl	2.02-3.1	backwards-compatible wrapper for Digest::MD5
ii	libmime-perl	5.411-2	Perl5 modules for MIME-compliant messages (M
ii	libmm13	1.3.0-1	Shared memory library - runtime
ii	libmp3-info-pe	1.01-1	Perl MP3::Info - Manipulate / fetch info fro
ii	libmysqlclient	3.23.56-2	LGPL-licensed client library for MySQL datab
ii	libmysqlclient	4.0.13-1	mysql database client library
ii	libncurses5	5.3.20030510-1	Shared libraries for terminal handling
ii	libnet-dns-per	0.33-1	Perform DNS queries from a Perl script
ii	libnet-perl	1.12-1	Implementation of Internet protocols for Per
ii	libnet-ssleay-	1.22-1	Perl module for Secure Sockets Layer (SSL)
ii	libnewt0	0.50.17-12	Not Erik's Windowing Toolkit - text mode win
ii	libnewt0.51	0.51.4-6	Not Erik's Windowing Toolkit - text mode win
ii	libogg0	1.0.0-1	Ogg Bitstream Library
ii	libopencdk4	0.4.2-3	Open Crypto Development Kit (OpenCDK) (runti
ii	libopenh323-1.	1.9.10-1	H.323 aka VoIP library
ii	libpam-modules	0.76-9	Pluggable Authentication Modules for PAM
ii	libpam-runtime	0.76-9	Runtime support for the PAM library
ii	libpam0g	0.76-9	Pluggable Authentication Modules library
ii	libpaper-utils	1.1.13	Library for handling paper characteristics (
ii	libpaper1	1.1.13	Library for handling paper characteristics
ii	libpcap0	0.6.2-2	System interface for user-level packet captu
ii	libpcap0.7	0.7.1-1	System interface for user-level packet captu
ii	libpcre3	3.9-1	Philip Hazel's Perl Compatible Regular Expre
ii	libperl5.8	5.8.0-17	Shared Perl library.
ii	libpng10-0	1.0.15-3	PNG library, older version - runtime
ii	libpng12-0	1.2.5.0-3	PNG library - runtime
ii	libpng2	1.0.15-3	PNG library, older version - runtime
ii	libpopt0	1.7-2	lib for parsing cmdline parameters
ii	libpostproc0	0.90rc4-sarge0	Mplayer postproc shared libraries

ii	libpt-1.3.11	1.3.11-1	Portable Windows Library
ii	libraw1394-5	0.9.0-4	library for direct access to IEEE 1394 bus (
ii	libreadline4	4.3-5	GNU readline and history libraries, run-time
ii	librpm4	4.0.4-14	RPM shared library
ii	libsasl7	1.5.27-3.5	Authentication abstraction library.
rc	libSDL1.2deb1a	1.2.4-1	Simple DirectMedia Layer (with X11 and OSS o
ii	libsensors1	2.6.5-4	Library to read temperature/voltage/fan sens
ii	libsigc++0	1.0.4-3	Type-safe Signal Framework for C++ - runtime
ii	libsnmp-base	5.0.7-1.1	NET SNMP (Simple Network Management Protocol
ii	libsnmp-session	0.95-0.1	Perl support for accessing SNMP-aware device
ii	libsnmp4.2	4.2.5-3.3	NET SNMP (Simple Network Management Protocol
ii	libsnmp5	5.0.7-1.1	NET SNMP (Simple Network Management Protocol
ii	libsoundsystem	1.1.0.kl-2	aRts Sound system (sound server data files)
ii	libspeex	1.0.rc3-1	The Speex Speech Codec
ii	libssl0.9.6	0.9.6j-1	SSL shared libraries (old version)
ii	libssl0.9.7	0.9.7b-2	SSL shared libraries
ii	libssl09	0.9.4-6.woody.	SSL shared libraries (old version)
ii	libstdc++2.10-	2.95.4-17	The GNU stdc++ library
ii	libstdc++3	3.0.4-14	The GNU stdc++ library version 3
ii	libstdc++5	3.3-2	The GNU Standard C++ Library v3
ii	libstring-shell	1.00-4	quote strings for passing through the shell
ii	libtasn1-0	0.1.2-1	Manage ASN.1 structures (runtime)
ii	libtiff3g	3.5.7-2	Tag Image File Format library
ii	libtime-date-perl	1.1400-1	Time and date functions for Perl
ii	libuc10	1.01-3	Portable compression library - runtime
rc	libungif4g	4.1.0b1-4	shared library for GIF images (runtime lib)
ii	liburi-perl	1.23-1	Manipulates and accesses URI strings
rc	libvorbis0	1.0.0-1	The Vorbis General Audio Compression Codec
ii	libvorbis0a	1.0.0-3	The Vorbis General Audio Compression Codec
ii	libvorbisenc2	1.0.0-3	The Vorbis General Audio Compression Codec
ii	libvorbisfile3	1.0.0-3	The Vorbis General Audio Compression Codec
ii	libwrap0	7.6-ipv6.1-3	Wietse Venema's TCP wrappers library
ii	libwww-perl	5.69-2	WWW client/server library for Perl (aka LWP)
rc	libxvidcore0	0.9.1-sarge0.1	MPEG-4 Video encoder
ii	lilo	22.5.4-1	LIinux LOader - The Classic OS loader can loa
ii	locales	2.3.1-16	GNU C Library: National Language (locale) da
ii	lockfile-progs	0.1.9	Programs for locking and unlocking files and
ii	logcheck	1.2.14	Mails anomalies in the system logfiles to th
ii	logcheck-datab	1.2.14	A database of system log rules for the use o
ii	login	4.0.3-8	System login tools
ii	logrotate	3.6.5-2	Log rotation utility
ii	logtail	1.2.14	Returns parts of logfiles that have not alre
ii	lpr	2000.05.07-4.2	BSD lpr/lpd line printer spooling system
ii	lsof	4.64-1	List open files.
ii	lynx	2.8.4.1b-5	Text-mode WWW Browser
ii	lzop	1.00-4	A real-time compressor.
ii	mailscanner	3.27.1-1	An email virus scanner and spam tagger.
ii	mailx	8.1.2-0.200305	A simple mail user agent
ii	make	3.80-1	The GNU version of the "make" utility.
ii	makedev	2.3.1-62	Creates device files in /dev.

ii	man-db	2.4.1-8	The on-line manual pager
ii	man2html	1.5k-4	Turns a web-browser and an httpd-server into
ii	manpages	1.48-2	Manual pages about using a GNU/Linux system
ii	manpages-es	1.28-10	Spanish man pages
ii	mawk	1.3.3-11	a pattern scanning and text processing langu
ii	mbr	1.1.5-1	Master Boot Record for IBM-PC compatible com
ii	mc	4.6.0-4	Midnight Commander - a powerful file manager
ii	memtest86	3.0-3	A thorough real-mode memory tester
ii	mime-support	3.23-1	MIME files 'mime.types' & 'mailcap', and sup
ii	mindl	0.81-1.1	Creates boot/root disks based on your system
ii	mkisofs	2.0+a15-1	Creates ISO-9660 CD-ROM filesystem images
ii	mlock	2003debian0.03	Mailbox locking program from UW
ii	modconf	0.2.44	Device Driver Configuration
ii	modutils	2.4.21-2	Linux module utilities.
ii	mondo	1.61-2	System to backup your filesystem to CDs
ii	mount	2.11z-1	Tools for mounting and manipulating filesyst
ii	mp3ogg	0.11-1	Converts MP3 file to Ogg Vorbis
ii	mpg123	0.59r-13	MPEG layer 1/2/3 audio player
ii	mpg321	0.2.10.1-1.1	A Free command-line mp3 player, compatible w
ii	mrtg	2.9.29-0.1	Multi Router Traffic Grapher
ii	mrtg-contrib	2.9.29-0.1	Multi Router Traffic Grapher (contributed fi
ii	mrtg-ping-prob	2.1.0-1	Ping module for Multi Router Traffic Grapher
ii	mrtgutlils	0.4	Utilities to generate statistics for mrtg
ii	mysql-client	4.0.13-1	mysql database client binaries
ii	mysql-common	4.0.13-1	mysql database common files (e.g. /etc/mysql
ii	mysql-server	4.0.13-1	mysql database server binaries
ii	nano	1.2.1-2	free Pico clone with some new features
ii	ncurses-base	5.3.20030510-1	Descriptions of common terminal types
ii	ncurses-bin	5.3.20030510-1	Terminal-related programs and man pages
ii	net-tools	1.60-6	The NET-3 networking toolkit
ii	netbase	4.09	Basic TCP/IP networking system
ii	netcat	1.10-22	TCP/IP swiss army knife
ii	netkit-inetd	0.10-9	The Internet Superserver
ii	netkit-ping	0.10-9	The ping utility from netkit
ii	nfs-common	1.0.3-1	NFS support files common to client and serve
ii	nfs-kernel-ser	1.0.3-1	Kernel NFS server support
ii	nmap	3.27-1	The Network Mapper
ii	ntop	2.1.0-3	display network usage in top-like format
ii	openmosix	0.2.4-5	Utilities to administer an openmosix node
ii	openssl	0.9.7b-2	Secure Socket Layer (SSL) binary and related
ii	passwd	4.0.3-8	Change and administer password and group dat
ii	patch	2.5.9-1	Apply a diff file to an original
ii	pciutils	2.1.11-2	Linux PCI Utilities (for 2.[12345].x kernels
ii	perl	5.8.0-17	Larry Wall's Practical Extraction and Report
ii	perl-base	5.8.0-17	The Pathologically Eclectic Rubbish Lister.
ii	perl-modules	5.8.0-17	Core Perl modules.
ii	php4	4.2.3-14	A server-side, HTML-embedded scripting langu
ii	php4-cgi	4.2.3-14	A server-side, HTML-embedded scripting langu
ii	php4-mysql	4.2.3-14	MySQL module for php4
ii	pine	4.56L-1	An e-mail reader with MIME and IMAP support

ii	po-debconf	0.6.9	Manage translated Debconf templates files wi
ii	popularity-con	1.3-1.1	Vote for your favourite packages automatical
ii	portmap	5-2	The RPC portmapper
ii	portsentry	1.1-3	Portscan detection daemon
ii	ppp	2.4.1.uus-5	Point-to-Point Protocol (PPP) daemon.
ii	pppconfig	2.1	A text menu based utility for configuring pp
ii	pptp-linux	1.2.0-2	Point-to-Point Tunneling Protocol (PPTP) Cli
ii	pptpd	1.1.4-0.b3.2	PoPToP Point to Point Tunneling Server
ii	procps	3.1.9-1	The /proc file system utilities
ii	psmisc	21.3-1	Utilities that use the proc filesystem
ii	python	2.2.2-6	An interactive object-oriented scripting lan
ii	python2.2	2.2.2-6	An interactive object-oriented scripting lan
ii	quota	3.08-8	An implementation of the disk quota system
ii	raidtools2	1.00.3-2	Utilities to support 'new-style' RAID disks
ii	razor	2.220-3	spam-catcher using a collaborative filtering
ii	rdate	1.4-4	Set the system's date from a remote host.
ii	reaim	0.8-1	Enable AIM and MSN file transfer on Linux ip
ii	rpm	4.0.4-14	Red Hat package manager
ii	rsync	2.5.6-0.1	fast remote file copy program (like rcp)
ii	samba	2.999+3.0.alph	A LanManager like file and printer server fo
ii	samba-common	2.999+3.0.alph	Samba common files used by both the server a
ii	screen	3.9.15-1	a terminal multiplexor with VT100/ANSI termi
ii	sed	4.0.7-1	The GNU sed stream editor
ii	setserial	2.17-33	Controls configuration of serial ports
ii	sharutils	4.2.1-10	shar, unshar, uuencode, uudecode
ii	shellutils	5.0-3	The GNU shell programming utilities (transit
ii	slangl	1.4.5-2	The S-Lang programming library - runtime ver
ii	slangla-utf8	1.4.5-2	The S-Lang programming library with utf8 sup
ii	smbclient	2.999+3.0.alph	A LanManager like simple client for Unix.
ii	smbfs	2.999+3.0.alph	mount and umount commands for the smbfs (for
ii	spamassassin	2.55-2	Perl-based spam filter using text analysis
ii	spamc	2.55-2	Client for perl-based spam filtering daemon
ii	spong-common	2.7.6a-12	A systems and network monitoring system -- c
ii	squid	2.5.2-1	Internet Object Cache (WWW proxy cache)
ii	ssh	3.6.1p2-2	Secure rlogin/rsh/rcp replacement (OpenSSH)
ii	stunnel	3.22-1	Universal SSL tunnel for network daemons
rc	svgalibg1	1.4.3-10	Console SVGA display utilities
ii	sysklogd	1.4.1-11	System Logging Daemon
ii	syslinux	2.00-2	Bootloader for Linux/i386 using MS-DOS flopp
ii	sysvinit	2.84-3	System-V like init.
ii	tar	1.13.25-5	GNU tar
ii	tasksel	1.25	Tool for selecting tasks for installation on
ii	tcpd	7.6-ipv6.1-3	Wietse Venema's TCP wrapper utilities
ii	tcpdump	3.7.1-1.2	A powerful tool for network monitoring and d
ii	telnet	0.17-20	The telnet client.
ii	tethereal	0.9.12-2	Network traffic analyzer (console)
ii	textutils	5.0-3	The GNU text file processing utilities (tran
ii	traceroute	1.4a12-12	Traces the route taken by packets over a TCP
ii	unarj	2.65-1	arj unarchive utility
ii	unison	2.9.1-1	A file-synchronization tool for Unix and Win

ii	unrar	3.1.3-1	Unarchiver for .rar files
ii	unzip	5.50-1	De-archiver for .zip files
ii	upx-ucl	1.24-2	an efficient live-compressor for executables
ii	usemod-wiki	0.92-3	Perl-based Wiki clone
ii	user-euro-es	0.27	Settings for european Spanish speaking users
ii	usermin	1.020-1	A web interface for user tasks
ii	util-linux	2.11z-1	Miscellaneous system utilities.
ii	uw-imapd	2003debian0.03	remote mail folder access server
ii	uw-imapd-ssl	2003debian0.03	Dummy upgrade package for uw-imapd
ii	vnc-common	3.3.6-4	Virtual network computing server software
ii	vorbis-tools	1.0.0-2	Several Ogg Vorbis Tools
ii	vpnd	1.1.0-6	Virtual Private Network Daemon
ii	webalizer	2.01.10-15	Web server log analysis program
ii	webmin	1.090-1	Web-based administration toolkit
ii	webmin-apache	1.090-1	apache control module for webmin
ii	webmin-bind	1.090-1	bind 8+ control module for webmin
ii	webmin-core	1.090-1	core modules for webmin
ii	webmin-cpan	1.090-1	CPAN module for webmin
ii	webmin-dhcpd	1.090-1	dhcpd control module for webmin
ii	webmin-exports	1.090-1	NFS exports control module for webmin
ii	webmin-fileman	0.980.6-1	file manager module for webmin
ii	webmin-firewal	1.090-1	iptables control module for webmin
ii	webmin-fsdump	1.090-1	dump/restore module for webmin
ii	webmin-inetd	1.090-1	inetd control module for webmin
ii	webmin-lilo	1.090-1	lilo control module for webmin
ii	webmin-lpadmin	1.090-1	printer control module for webmin
ii	webmin-mysql	1.090-1	mysql-server control module for webmin
ii	webmin-quota	1.090-1	disk quota control module for webmin
ii	webmin-raid	1.090-1	raid control module for webmin
ii	webmin-samba	1.090-1	samba control module for webmin
ii	webmin-sentry	1.090-1	portsentry module for webmin
ii	webmin-softwar	1.090-1	software packages control module for webmin
ii	webmin-squid	1.090-1	squid control module for webmin
ii	webmin-sshd	1.090-1	SSH server control module for webmin
ii	webmin-status	1.090-1	server and system status control module for
ii	webmin-stunnel	1.090-1	stunnel control module for webmin
ii	webmin-telnet	0.980-3	telnet module for webmin
ii	webmin-updown	1.090-1	File transfer module for webmin
ii	webmin-usermin	1.090-1	usermin control module for webmin
ii	webmin-webaliz	1.090-1	webalizer control module for webmin
ii	wget	1.8.2-10	retrieves files from the web
ii	whiptail	0.51.4-6	Displays user-friendly dialog boxes from she
ii	whois	4.6.5	The GNU whois client
ii	wspanish	1.0.11.8	The Spanish dictionary words for /usr/share/
ii	wwwconfig-comm	0.0.26	Debian web auto configuration
ii	xfree86-common	4.3.0-0woody4	X Window System (XFree86) infrastructure
ii	xlibmesa3	4.2.1-6	XFree86 Mesa libraries pseudopackage
ii	xlibmesa3-gl	4.2.1-6	Mesa 3D graphics library [XFree86]
ii	xlibmesa3-glu	4.2.1-6	Mesa OpenGL utility library [XFree86]
ii	xlibs	4.3.0-0woody4	X Window System client libraries

```
ii xlibs-data      4.3.0-0woody4 X Window System client data
ii xutils         4.3.0-0woody4 X Window System utility programs
ii yaps           0.96-1        Yet Another Pager Software
ii zebra          0.93b-3       A GPL'd, BGP/OSPF/RIP capable routing daemon
ii zlib1g        1.1.4-12      compression library - runtime
```

A ver, en resumen, interesa instalar el ssh porque con el podremos entrar desde fuera al servidor, el webmin y los módulos listados arriba, etc

Lo dicho, esto es lo que yo tengo puesto... los paquetes se instalan poniendo:

```
apt-get install <paquetes>
```

por ejemplo: apt-get install wavemon zebra webmin-stunnel webmin-status wget vtun

Automáticamente el programa se conectará a Internet y comenzará a bajar esos paquetes y todos los necesarios para que esos funcionen, es decir, si instalas webmin-status, para eso te hará falta primero el webmin y el programa lo instalará también solito tras pedir confirmación e indicar los megas a descargar y lo que ocupará una vez descomprimido.

Capítulo 4

Red

Vale, se supone que ahora ya tenemos el sistema funcionando y bueno... algo es algo :) ahora viene lo serio... configurarlo para que se adapte a nuestras necesidades...

Lo primero es tener un Kernel modernito (a mi con el estándar no me iba, pero como veréis en la lista de paquetes con el 2.4 funciona bastante bien el P'N'P)

Tenemos que por un lado configurar las tarjetas de red:

Para eso, entramos en /etc/modutils como root y tendremos varios archivos de configuración, para ser representativo pondré el de una tarjeta ISA que son los más difíciles:

Contenido de : /etc/modutils/eepro (entre —)

```
options eeepro io=0x210
```

Como veréis es altamente jodidísimo configurar una tarjeta de red... sólo indicando el puerto de entrada salida Linux ya le busca la IRQ apropiada (en casos chungos, se le puede indicar también)

Con el resto de tarjetas editaremos los ficheros para ver que está todo bien y los llamaremos con el nombre del módulo necesario (sale durante la instalación o en páginas de ayuda)

Luego, editaremos el fichero /etc/modules y pondremos las tarjetas en el orden en el que queremos que se llamen:

```
# /etc/modules: kernel modules to load at boot time.
#
# This file should contain the names of kernel modules that are
# to be loaded at boot time, one per line.  Comments begin with
# a "#", and everything on the line after them are ignored.

unix
af_packet
eeepro
tulip
3c59x
ne
sb
```

Es decir, primero se cargará la EtherExpress Pro 10/10+ de Intel, que se llamará eth0 (tal como se explicó en el apartado de hardware), luego la 8139 (10/100) como eth2, luego la ne2000 PCI (RTL 8029), luego la Novel 2000 ISA y por último la SoundBlaster...

En el caso de haber dos tarjetas con el mismo controlador, se irán creando consecutivamente... es decir, si hay dos RTL8029, pues serán eth1 y eth2 y así el resto... en mi caso como cada una es distinta no tuve ese "problema".

Tras editar ese fichero, tendremos que hacer que al siguiente arranque se tome esta configuración, así que ejecutaremos update-modules, para que el ordenador cree el /etc/modules.conf adecuado conforme a nuestros deseos., al siguiente arranque tendremos algo como:

```
Real Time Clock Driver v1.10e
id: 0xb4 <7>io: 0x210 <6>eth0: Intel EtherExpress Pro/10+ ISA
at 0x210,<6>00<6>:aa<6>:00<6>:c9<6>:9d<6>:1a<6>, IRQ 11, 10BaseT.
eeepro.c: v0.13 11/08/2001 aris@cathedrallabs.org
8139too Fast Ethernet driver 0.9.24
eth1: RealTek RTL8139 Fast Ethernet at 0xc486b000, 00:50:fc:4d:c7:d9, IRQ 9
eth1: Identified 8139 chip type 'RTL-8139C'
ne2k-pci.c:v1.02 10/19/2000 D. Becker/P. Gortmaker
http://www.scyld.com/network/ne2k-pci.html
eth2: RealTek RTL-8029 found at 0x6100, IRQ 9, 00:C0:DF:E3:46:F9.
isapnp: Scanning for PnP cards...
isapnp: Calling quirk for 01:00
isapnp: SB audio device quirk - increasing port range
isapnp: Card 'Creative SB16 PnP'
isapnp: 1 Plug & Play card detected total
ne.c:v1.10 9/23/94 Donald Becker (becker@scyld.com)
Last modified Nov 1, 2000 by Paul Gortmaker
NE*000 ethercard probe at 0x340: 00 40 33 94 ad 7d
eth3: NE2000 found at 0x340, using IRQ 10.
Soundblaster audio driver Copyright (C) by Hannu Savolainen 1993-1996
sb: Creative SB16 PnP detected
sb: ISAPnP reports 'Creative SB16 PnP' at i/o 0x220, irq 5, dma 1, 5
SB 4.13 detected OK (220)
sb: 1 Soundblaster PnP card(s) found.
```

Como vemos, ya ha ido asignando tarjetas.. eth0 la Intel, eth1 la 8139, eth2 la 8029 y eth3 la Ne2000, lo de la SoundBlaster es accesorio, pero mola que la pille el solito ;) (con un kernel de la 2.4 aviso...)

Ahora falta configurar las direcciones para cada tarjeta... editamos el fichero /etc/network/interfaces:

```
# The first network card - this entry was created during the Debian installation
auto eth0
iface eth0 inet dhcp

auto eth1
```

```

iface eth1 inet static
    address 1.1.1.1
    netmask 255.255.255.0

auto eth2
iface eth2 inet static
    address 1.1.2.1
    netmask 255.255.255.0

auto eth3
iface eth3 inet static
    address 10.34.12.129
    netmask 255.255.255.224

#auto eth3
#iface eth3 inet static
#    address 1.1.3.1
#    netmask 255.255.255.0

```

Esto en cristiano viene a decir que cargue las tarjetas en el arranque (auto eth?), y lo de abajo, pues cómo configurarlas... la eth0 es la que estaba conectada a internet, por lo tanto como mi proveedor configura por DHCP, pues eso pone... la eth1 es la de mi red local con el primero ordenador, le asigna una ip 1.1.1.1 y una máscara de subred de tipo C.

En realidad la dirección 1.1.1.1 está asignada en internet y sería una cabronada el conectarme con eso configurado así, pero como hemos dicho, es una red local y no tiene porqué afectar a nadie, así que así se queda...

Con la segunda para la red local, pues lo mismo, pero con otra dirección 1.1.2.1 y con subred de tipo C y la ethernet 3 es lo que sería la wireless y le asigno una dirección real y válida, en mi caso, esta ip pertenece a Valencia Wireless y está asignada a mi nodo, cada uno que consulte en <http://www.redlibre.net> el direccionamiento y que se ponga encargado con el responsable de su zona si lo hay y si no, directamente con Redlibre...

Ahora, al arrancar el ordenador que hace de nodo, ya debería coger automáticamente esos datos para cada tarjeta y funcionar así bien...

Me podéis preguntar que porqué asigno la 1.1.1.1 a una y la 1.1.2.1 a dos tarjetas que pertenecen a la misma red local... lo ideal sería 1.1.1.2, pero luego tenía problemas con el DHCP (con un hub nunca hubiera pasado... pero como nadie me ha dado ninguno...) (luego se verá el motivo)

El siguiente paso, es que ya puestos, que el ordenador asigne direcciones a los ordenadores que se conecten de forma automática, es decir.. montar un servidor DHCP... para ello instalaremos el paquete DHCP y el servidor de nombres BIND, para que los equipos remotos que se conecten puedan pedir configuración automáticamente y que puedan resolver nombres...

El fichero de configuración del DHCP es el siguiente: (/etc/dhcp3/dhcp.conf):

```

# dhcpd.conf
#

```

```
# Sample configuration file for ISC dhcpd
#

ddns-updates on;
use-host-decl-names on;
allow unknown-clients;
default-lease-time 3600;
max-lease-time 7200;
authoritative;

subnet 10.34.12.128 netmask 255.255.255.224 {
    option domain-name-servers 10.34.12.129;
    range 10.34.12.131 10.34.12.158;
    option broadcast-address 10.34.12.159;
    option routers 10.34.12.129;
}

subnet 192.168.1.0 netmask 255.255.255.0 {
    option domain-name-servers 192.168.1.1;
    range 192.168.1.2 192.168.1.254;
    option broadcast-address 192.168.1.255;
    option routers 192.168.1.1;
}

subnet 1.1.1.0 netmask 255.255.255.0 {
    option domain-name-servers 1.1.1.1;
    option domain-name alufis;
    range 1.1.1.3 1.1.1.254;
    option broadcast-address 1.1.1.255;
    option routers 1.1.1.1;
    option netbios-name-servers 1.1.1.1;
    option netbios-node-type 8;
}

subnet 1.1.2.0 netmask 255.255.255.0 {
    option domain-name alufis;
    option netbios-name-servers 1.1.2.1;
    option netbios-node-type 8;
    option domain-name-servers 1.1.2.1;
    range 1.1.2.3 1.1.2.254;
    option broadcast-address 1.1.2.255;
    option routers 1.1.2.1;
}

subnet 1.1.3.0 netmask 255.255.255.0 {
    option domain-name-servers 1.1.3.1;
    option domain-name alufis;
    range 1.1.3.3 1.1.3.254;
    option broadcast-address 1.1.3.255;
```

```

option routers 1.1.3.1;
option netbios-name-servers 1.1.3.1;
option netbios-node-type 8;
}

```

```

host deneb {
option host-name "deneb";
hardware ethernet 00:04:e2:33:4e:8d;
fixed-address 1.1.1.2;
server-name "deneb";
}

```

```

host darkstar {
option host-name "darkstar";
hardware ethernet 00:80:c8:16:de:59;
server-name "darkstar";
fixed-address 1.1.2.2;
}

```

```

host Alnilam {
option host-name "alnilam";
hardware ethernet 00:90:d1:06:57:dc;
fixed-address 10.34.12.130;
server-name "alnilam";
}

```

A ver... por partes...

Primero defino lo que será la wireless, con la ip asignada, la subred, el rango de ip's, etc...

```

subnet 10.34.12.128 netmask 255.255.255.224 {
option domain-name-servers 10.34.12.129;
range 10.34.12.130 10.34.12.158;
option broadcast-address 10.34.12.159;
option routers 10.34.12.129;
}

```

Esto viene a significar que la red 10.34.12.128 (valencia wireless, mi nodo), con máscara de subred (255.255.255.224), tiene un servidor de nombres situado en 10.34.12.129 (la asignada a eth3), una dirección de broadcast en .158 y un router en .129 (el mismo que el DNS, es decir, el nodo...).

El rango de ip's asignado van desde la siguiente al nodo (.33) hasta el .62 de esa forma los clientes que pidan configuración por DHCP por el interfaz con esa subred (eth3) obtendrán una IP dentro de ese rango...

192.168.*.* son direcciones para redes privadas y en este caso la utilizaré para los equipos que se conecten mediante la VPN, es decir, equipos que primero se conectan por la wireless y luego se conectan mediante VPN, realmente no creo que fuese necesario indicarlo, pero lo prefiero así :)

Luego vienen definidas las redes para las tarjetas locales 1.1.1.1 y 1.1.2.1, si ambas tarjetas estuviesen en la misma red, con una sola definición bastaría, pero el problema sería que si a la 1.1.1.3 que se conecta por la segunda tarjeta de red con IP 1.1.1.2 le digo que el router es 1.1.1.1 que es la primera tarjeta de red, pues habrían problemas... no iría ni a tiros... esto con un HUB no pasa y no hace falta poner estas dos definiciones, pero en mi caso sí... así cada una es una red distinta que va bien y asigna ip's automáticamente y resuelve nombres.

Luego vienen dos declaraciones de equipo, en mi caso, a mis dos equipos de casa, les asigno ip's fijas, porque no me van a hacer falta más (de hecho no haría falta ni el dhcp para ellas, pero bueno, puestos a hacerlo a lo grande... pues ya lo tengo listo para un día meter dos hubs y poder conectar tropocientos ordenadores en cada tarjeta de red ;))

Para asignar IP fija, se hace en base a la dirección MAC de la tarjeta de red, que son esos códigos en hexadecimal.

En el último caso defino otro ordenador pero sin ip fija (en su momento irá por la wireless y por eso no tiene nada puesto)

Respecto al servidor de nombres... no hice nada, sólo instalé el paquete y ya iba, así que no toqué nada :)

Hasta este punto los equipos cliente que se configuren para que pidan la configuración automáticamente, recibirán una configuración válida desde nuestro nodo, ahora sólo falta hacer alguna cosilla más ;)

Capítulo 5

Masquerading

Tenemos un servidor DHCP, las tarjetas configuradas y un servidor de nombres... ahora sólo falta que enrute!!! de esa forma tendremos acceso a Internet desde cualquiera de nuestros ordenadores...

Para ello, si leemos el IP-MASQUERADING-HOWTO, sacaremos este interesante script:

```
#!/bin/sh
#
# rc.firewall-2.4
FWVER=0.63
#
#           Initial SIMPLE IP Masquerade test for 2.4.x kernels
#           using IPTABLES.
#
#           Once IP Masquerading has been tested, with this simple
#           ruleset, it is highly recommended to use a stronger
#           IPTABLES ruleset either given later in this HOWTO or
#           from another reputable resource.
#
#
#
# Log:
#   0.63 - Added support for the IRC IPTABLES module
#   0.62 - Fixed a typo on the MASQ enable line that used eth0
#         instead of $EXTIF
#   0.61 - Changed the firewall to use variables for the internal
#         and external interfaces.
#   0.60 - 0.50 had a mistake where the ruleset had a rule to DROP
#         all forwarded packets but it didn't have a rule to ACCEPT
#         any packets to be forwarded either
#         - Load the ip_nat_ftp and ip_contrack_ftp modules by default
#   0.50 - Initial draft
#
```

```

echo -e "\n\nLoading simple rc.firewall version $FWVER..\n"

# The location of the 'iptables' program
#
#   If your Linux distribution came with a copy of iptables, most
#   likely it is located in /sbin.  If you manually compiled
#   iptables, the default location is in /usr/local/sbin
#
# ** Please use the "whereis iptables" command to figure out
# ** where your copy is and change the path below to reflect
# ** your setup
#
IPTABLES=/sbin/iptables
#IPTABLES=/usr/local/sbin/iptables

#Setting the EXTERNAL and INTERNAL interfaces for the network
#
#   Each IP Masquerade network needs to have at least one
#   external and one internal network.  The external network
#   is where the natting will occur and the internal network
#   should preferably be addressed with a RFC1918 private address
#   scheme.
#
#   For this example, "eth0" is external and "eth1" is internal"
#
#   NOTE:  If this doesnt EXACTLY fit your configuration, you must
#           change the EXTIF or INTIF variables above.  For example:
#
#           EXTIF="ppp0"
#
#           if you are a modem user.
#
EXTIF="eth0"
EXTIF2="eth3"
EXTIF3="s1+"

INTIF="eth1"
INTIF2="eth2"
INTIF3="ppp+"

LAN1="1.1.1.0/16"

echo "   External Interface   : $EXTIF"
echo "   External Interface 2 : $EXTIF2"
echo "   External Interface 3 : $EXTIF3"
echo "   -----"
echo "   Internal Interface    : $INTIF"
echo "   Internal Interface 2 : $INTIF2"

```

```
echo "   Internal Interface 3 : $INTIF3"
echo "   -----"
echo "   Internal Network      : $LAN1"

#####
#== No editing beyond this line is required for initial MASQ testing ==

echo -en "   loading modules: "

# Need to verify that all modules have all required dependencies
#
#echo " - Verifying that all kernel modules are ok"
#/sbin/depmod -a

# With the new IPTABLES code, the core MASQ functionality is now either
# modular or compiled into the kernel. This HOWTO shows ALL IPTABLES
# options as MODULES. If your kernel is compiled correctly, there is
# NO need to load the kernel modules manually.
#
# NOTE: The following items are listed ONLY for informational reasons.
#       There is no reason to manual load these modules unless your
#       kernel is either mis-configured or you intentionally disabled
#       the kernel module autoloader.
#

# Upon the commands of starting up IP Masq on the server, the
# following kernel modules will be automatically loaded:
#
# NOTE: Only load the IP MASQ modules you need. All current IP MASQ
#       modules are shown below but are commented out from loading.
# =====

#Load the main body of the IPTABLES module - "iptables"
# - Loaded automatically when the "iptables" command is invoked
#
# - Loaded manually to clean up kernel auto-loading timing issues
#
echo -en "ip_tables, "
/sbin/insmod ip_tables

#Load the IPTABLES filtering module - "iptables_filter"
# - Loaded automatically when filter policies are activated

#Load the stateful connection tracking framework - "ip_conntrack"
#
# The conntrack module in itself does nothing without other specific
```

```
# conntrack modules being loaded afterwards such as the "ip_conntrack_ftp"
# module
#
# - This module is loaded automatically when MASQ functionality is
#   enabled
#
# - Loaded manually to clean up kernel auto-loading timing issues
#
echo -en "ip_conntrack, "
/sbin/insmod ip_conntrack
```

```
#Load the FTP tracking mechanism for full FTP tracking
#
# Enabled by default -- insert a "#" on the next line to deactivate
#
echo -en "ip_conntrack_ftp, "
/sbin/insmod ip_conntrack_ftp
```

```
#Load the IRC tracking mechanism for full IRC tracking
#
# Enabled by default -- insert a "#" on the next line to deactivate
#
echo -en "ip_conntrack_irc, "
/sbin/insmod ip_conntrack_irc
```

```
#Load the general IPTABLES NAT code - "iptable_nat"
# - Loaded automatically when MASQ functionality is turned on
#
# - Loaded manually to clean up kernel auto-loading timing issues
#
echo -en "iptable_nat, "
/sbin/insmod iptable_nat
```

```
#Loads the FTP NAT functionality into the core IPTABLES code
# Required to support non-PASV FTP.
#
# Enabled by default -- insert a "#" on the next line to deactivate
#
echo -en "ip_nat_ftp, "
/sbin/insmod ip_nat_ftp

echo -en "ip_nat_irc, "
/sbin/insmod ip_nat_irc
```

```
# Just to be complete, here is a list of the remaining kernel modules
# and their function. Please note that several modules should be only
# loaded by the correct master kernel module for proper operation.
# -----
#
# ipt_mark      - this target marks a given packet for future action.
#                This automatically loads the ipt_MARK module
#
# ipt_tcpmss    - this target allows to manipulate the TCP MSS
#                option for braindead remote firewalls.
#                This automatically loads the ipt_TCPMSS module
#
# ipt_limit     - this target allows for packets to be limited to
#                to many hits per sec/min/hr
#
# ipt_multiport - this match allows for targets within a range
#                of port numbers vs. listing each port individually
#
# ipt_state     - this match allows to catch packets with various
#                IP and TCP flags set/unset
#
# ipt_unclean   - this match allows to catch packets that have invalid
#                IP/TCP flags set
#
# iptable_filter - this module allows for packets to be DROPPed,
#                REJECTed, or LOGged. This module automatically
#                loads the following modules:
#
#                ipt_LOG - this target allows for packets to be
#                logged
#
#                ipt_REJECT - this target DROPS the packet and returns
#                a configurable ICMP packet back to the
#                sender.
#
# iptable_mangle - this target allows for packets to be manipulated
#                for things like the TCPMSS option, etc.

echo ". Done loading modules."

#CRITICAL: Enable IP forwarding since it is disabled by default since
#
#           Redhat Users: you may try changing the options in
#                       /etc/sysconfig/network from:
#
#                       FORWARD_IPV4=false
#                       to
```

```

#                               FORWARD_IPV4=true
#
echo "  enabling forwarding.."
echo "1" > /proc/sys/net/ipv4/ip_forward

# Dynamic IP users:
#
#   If you get your IP address dynamically from SLIP, PPP, or DHCP,
#   enable this following option. This enables dynamic-address hacking
#   which makes the life with Diald and similar programs much easier.
#
echo "  enabling DynamicAddr.."
echo "1" > /proc/sys/net/ipv4/ip_dynaddr

# Enable simple IP forwarding and Masquerading
#
# NOTE: In IPTABLES speak, IP Masquerading is a form of SourceNAT or SNAT.
#
# NOTE #2: The following is an example for an internal LAN address in the
#          192.168.0.x network with a 255.255.255.0 or a "24" bit subnet mask
#          connecting to the Internet on external interface "eth0". This
#          example will MASQ internal traffic out to the Internet but not
#          allow non-initiated traffic into your internal network.
#
#
# Borrar reglas anteriores tanto en normal como en nat y por defecto no enrutar
$IPTABLES -t nat -F
$IPTABLES -F
$IPTABLES -P FORWARD DROP

#Conectar la red local para transferencia de datos
$IPTABLES -A FORWARD -s $LAN1 -d $LAN1 -j ACCEPT

#Pasar datos desde las locales a inet
$IPTABLES -A FORWARD -s $LAN1 -o $EXTIF -j ACCEPT
# Aceptar paquetes de entrada a la local
$IPTABLES -A FORWARD -i $EXTIF -d $LAN1 -m state --state ESTABLISHED,RELATED -j ACCEPT

# Enmascarar a internet
$IPTABLES -t nat -A POSTROUTING -o $EXTIF -j MASQUERADE

#Pasar datos desde las locales a la wireless
$IPTABLES -A FORWARD -s $LAN1 -o $EXTIF2 -j ACCEPT
$IPTABLES -A FORWARD -i $EXTIF2 -d $LAN1 -m state --state ESTABLISHED,RELATED -j ACCEPT

# Enmascarar a la wireless

```

```

$IPTABLES -t nat -A POSTROUTING -s $LAN1 -o $EXTIF2 -j MASQUERADE

#Entrada de la red local a los túneles wireless (sl+)
$IPTABLES -A FORWARD -s $LAN1 -o $EXTIF3 -j ACCEPT
$IPTABLES -A FORWARD -i $EXTIF3 -d $LAN1 -m state --state ESTABLISHED,RELATED -j ACCEPT

# Enmascarar a los túneles
$IPTABLES -t nat -A POSTROUTING -s $LAN1 -o $EXTIF3 -j MASQUERADE

#Paquetes de entrada entre las wireless y los túneles Wireless (sl+)
$IPTABLES -A FORWARD -i $EXTIF2 -o $EXTIF3 -j ACCEPT
$IPTABLES -A FORWARD -i $EXTIF3 -o $EXTIF2 -j ACCEPT

# Paquetes de entrada de la VPN a internet y a la wireless
$IPTABLES -A FORWARD -i $INTIF3 -o $EXTIF -j ACCEPT
$IPTABLES -A FORWARD -i $EXTIF -o $INTIF3 -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A FORWARD -i $INTIF3 -o $EXTIF2 -j ACCEPT
$IPTABLES -A FORWARD -i $EXTIF2 -o $INTIF3 -m state --state ESTABLISHED,RELATED -j ACCEPT

#
#           Programas
#-----

# Proxy Transparente
echo "Transparent Proxy for SQUID"
$IPTABLES -t nat -A PREROUTING -p TCP --dport 80 -j REDIRECT --to-port 3128 -d ! $LAN1
$IPTABLES -t nat -A PREROUTING -p TCP --dport 25 -j REDIRECT --to-port 25 -d ! $LAN1
$IPTABLES -t nat -A PREROUTING -p TCP --dport 110 -j REDIRECT --to-port 110 -d ! $LAN1

# Gnome Meeting
echo "Gnome Meeting"
$IPTABLES -t nat -A PREROUTING -i $EXTIF -p TCP --dport 30000:30010 -j DNAT --to-dest 1.1.1.1
$IPTABLES -t nat -A PREROUTING -i $EXTIF -p TCP --dport 1720 -j DNAT --to-dest 1.1.1.2
$IPTABLES -A FORWARD -p tcp -i $EXTIF --dport 30000:30010 -d 1.1.1.2 -j ACCEPT
$IPTABLES -A FORWARD -p udp -i $EXTIF --dport 5000:5003 -d 1.1.1.2 -j ACCEPT
$IPTABLES -t nat -A PREROUTING -i $EXTIF -p UDP --dport 5000:5003 -j DNAT --to-dest 1.1.1.2
$IPTABLES -A FORWARD -p tcp -i $EXTIF --dport 1720 -d 1.1.1.2 -j ACCEPT

#Aim, MSN
echo "AIM, MSN"
$IPTABLES -t nat -A PREROUTING -i $EXTIF -p tcp --dport 5190 -j REDIRECT --to-ports 5190
$IPTABLES -t nat -A PREROUTING -i $EXTIF -p tcp --dport 1863 -j REDIRECT --to-ports 1863
$IPTABLES -A INPUT -i $EXTIF -p tcp --dport 5190 -j ACCEPT
$IPTABLES -A INPUT -i $EXTIF -p tcp --dport 1863 -j ACCEPT
$IPTABLES -A INPUT -i $EXTIF -p tcp --dport 4443 -j ACCEPT
$IPTABLES -A INPUT -i $EXTIF -p tcp --dport 5566 -j ACCEPT
$IPTABLES -A INPUT -i $EXTIF -p tcp --dport 1864 -j ACCEPT

#VNC Deneb :0

```

```

echo "VNC"
$IPTABLES -A FORWARD -i $EXTIF -p TCP --dport 5800 -j ACCEPT
$IPTABLES -A FORWARD -i $EXTIF -p TCP --dport 5900 -j ACCEPT
$IPTABLES -t nat -A PREROUTING -i $EXTIF -p TCP --dport 5800 -j DNAT --to 1.1.1.2:5800
$IPTABLES -t nat -A PREROUTING -i $EXTIF -p TCP --dport 5900 -j DNAT --to 1.1.1.2:5900
$IPTABLES -A FORWARD -i $EXTIF -p TCP --dport 5500 -j ACCEPT
$IPTABLES -t nat -A PREROUTING -i $EXTIF -p TCP --dport 5500 -j DNAT --to 1.1.1.2:5500

#Firewall
echo "Firewall"
$IPTABLES -A INPUT -p TCP --dport 3306 -i $EXTIF -j DROP
$IPTABLES -A INPUT -p TCP --dport 3306 -i $EXTIF2 -j DROP
$IPTABLES -A INPUT -p TCP --dport 3128 -i $EXTIF -j DROP
$IPTABLES -A INPUT -p TCP --dport 53 -i $EXTIF -j DROP
$IPTABLES -A INPUT -p TCP --dport 25 -i $EXTIF -j DROP
$IPTABLES -A INPUT -p TCP --dport 25 -i $EXTIF2 -j DROP
$IPTABLES -A INPUT -p TCP --dport 137 -i $EXTIF -j DROP
$IPTABLES -A INPUT -p TCP --dport 137 -i $EXTIF2 -j DROP
$IPTABLES -A INPUT -p TCP --dport 139 -i $EXTIF -j DROP
$IPTABLES -A INPUT -p TCP --dport 139 -i $EXTIF2 -j DROP
$IPTABLES -A INPUT -p TCP --dport 179 -i $EXTIF -j DROP
$IPTABLES -A INPUT -p TCP --dport 2600:2605 -i $EXTIF -j DROP
$IPTABLES -A INPUT -p TCP --dport 179 -i $EXTIF2 -j DROP
$IPTABLES -A INPUT -p TCP --dport 2600:2605 -i $EXTIF2 -j DROP

```

Si nos olvidamos hasta donde pone algo con IPTABLES, lo que tenemos es un script que habilita la funcionalidad de enrutado del kernel de linux, a partir de ese momento ya tendremos acceso "teórico" a internet...

El problema es que como sólo tendremos una dirección IP válida, lo que tendremos que hacer es hacer NAT (Network Address Translation), existen dos tipos de NAT, el de destino y el de origen, en nuestro caso es el de origen, es decir, al final, los datos que salgan tendrán que aparentar ser todos desde la misma IP...

Para eso utilizamos esas sentencias de iptables que es el firewall, etc incorporado en Linux:

Lo de hacerlo con variables (las definidas al principio), es por facilidad a la hora de modificarlo, como veréis las redes están definidas como IP/bits para hacer las conversiones acudid a: <http://www.linux-es.com/ipcalc.php> <http://www.linux-es.com/ipcalc.php>

Las reglas definidas en este fichero, permiten que todo lo que salga por la eth0 (Internet) sea con NAT y que a su vez, lo que salga por eth3 (la wireless) sea también con NAT (para tener acceso a la wireless desde la red local), luego permite el tráfico de paquetes entre las ip's de las redes locales.

El script original fue modificado para cumplir con los requisitos de mi red, así que cada uno que lo adapte a la suya para que le sea útil.

Vale, un punto importante, es que al tener un trasto conectado todo el día a Internet, lo mejor es tenerlo bien asegurado, por eso, contar con una buena configuración de firewall así como el portscanner vendrá bien. Port scanner analiza y registra todos los intentos de escaneo de puertos contra nuestro equipo y posteriormente impide todo

acceso desde las ip's originantes, tanto creando rutas nulas como vuelta a esas ip's como añadiéndola a los filtros de los TCP-wrappers

En /etc/portsenry.ignore.static, pondremos:

```
-----
127.0.0.1/32
0.0.0.0
1.1.0.0/16
-----
```

Para que no bloquee los locales (como indica en la ayuda) y añadimos los de nuestra red local... equipos con IP 1.1.*.* (esto permite bloquear a los graciositos que se conecten por la wireless...)

Luego, en el /etc/porsentry/portsenry.conf cambiaremos:

```
-----
BLOCK_UDP="1"
BLOCK_TCP="1"
-----
```

y entre las otras opciones escogeremos las aptas para nuestro sistema, el tipo de bloqueo a realizar, etc

Ahora si alguien intenta hacer el tonto haciendo escaneos de puertos, pues automáticamente le será impedido posterior acceso desde esa Ip...

Para arrancar el script del firewall crearemos un script con el formato estándar tal como sigue:

/etc/init.d/fire

```
-----
#!/bin/sh
#
# chkconfig: 2345 11 89
#
# description: Loads the rc.firewall-2.2 ruleset.
#
# processname: firewall-2.2
# pidfile: /var/run/firewall.pid
# config: /etc/rc.d/rc.firewall
# probe: true

#
#-----
# v02/09/02
#
# Part of the copyrighted and trademarked TrinityOS document.
# http://www.ecst.csuchico.edu/~dranch
#
# Written and Maintained by David A. Ranch
# dranch@trinnet.net
#
# Updates
# -----
#
```

```
#
#-----

# Source function library.
#. /etc/rc.d/init.d/functions

# Check that networking is up.

# This line no longer work with bash2
#[ ${NETWORKING} = "no" ] && exit 0
# This should be OK.
[ "XXXX${NETWORKING}" = "XXXXno" ] && exit 0

[ -x /sbin/ifconfig ] || exit 0

# The location of various iptables and other shell programs
#
# If your Linux distribution came with a copy of iptables, most
# likely it is located in /sbin. If you manually compiled
# iptables, the default location is in /usr/local/sbin
#
# ** Please use the "whereis iptables" command to figure out
# ** where your copy is and change the path below to reflect
# ** your setup
#

IPTABLES=/sbin/iptables

# See how we were called.
case "$1" in
start)
/etc/init.d/rc.firewall
;;

stop)
echo -e "\nFlushing firewall and setting default policies to DROP\n"
$IPTABLES -F INPUT
$IPTABLES -F OUTPUT
$IPTABLES -F FORWARD
$IPTABLES -F -t nat

# Delete all User-specified chains
$IPTABLES -X
#
# Reset all IPTABLES counters
```

```

    $IPTABLES -Z
    ;;

restart)
    $0 stop
    $0 start
    ;;

status)
    $IPTABLES -L
    ;;

mlist)
    cat /proc/net/ip_contrack
    ;;

*)
    echo "Usage: fire {start|stop|status|mlist}"
    exit 1
esac

exit 0
```

Si ahora creamos los enlaces simbólicos apropiados a este script en `/etc/rc?.d` pondremos especificar cuando queremos que se arranque el enrutamiento...

En mi caso:

```
rc0.d/K20fire
rc1.d/K20fire
rc2.d/S20fire
rc3.d/S20fire
rc4.d/S20fire
rc5.d/S20fire
rc6.d/K20fire
```

En esas carpetas dentro de `/etc`

Un problema que tendremos al apagar el ordenador, es que como no tendremos monitor, no podremos saber cuando está listo para apagar y no vamos a conectar un monitor cada vez para hacerlo...

Capítulo 6

Túneles

Ahora que parece que está más o menos esto en marcha, habría que hacer túneles entre los distintos nodos de otras personas, para eso os recomiendo el paquete VPND junto a los scripts de VPNS: VPN's <http://freshmeat.net/projects/vpns/>
Hay un bonito README en el VPNS.

Capítulo 7

Consejos Finales

Vale, nuestro servidor tiene las tarjetas configuradas, enruta, hace de servidor DHCP, DNS, bloquea los escaneos de puertos... ahora sólo queda algún detalle interesante:

Lo primero: activar el soporte para el sistema de ficheros ext3, su compatibilidad con el ext2 es total, el cambio se hace simplemente poniendo:

```
tune2fs -j /dev/hda1 (para la partición 1 del hd)
```

y luego editando el `/etc/fstab` y donde ponga `/dev/hda1` cambiar ext2 por ext3...

De ese modo, el sistema arrancará igual, pero con una ventaja... en caso de apagón, bloqueo, reseteo, etc, el sistema de ficheros ext3 crea un log de los cambios realizados y en esas situaciones, al siguiente arranque, los puede corregir en la mayoría de las situaciones el solito, provocando que si se va la luz, en cuanto vuelva, automáticamente intente arrancar el solito (y en la mayoría de los casos lo conseguirá ;))

Otras cosas útiles.... configurar el SMB para poder acceder desde la red local con Windows... así el servidor que hace de nodo, puede quedarse bajando cosas por la noche (si es nodo tiene que estar todo el día en marcha) y luego simplemente te las transfieres a tu equipo mediante el entorno de red... (Existen hotwots que lo explican muy bien así que.... a usar el Google que para eso está...)

Sería también recomendable que instalases el DNS2GO que permite asignar un nombre de dominio a una IP dinámica, de ese modo podrías aprovechar para poner una página web en tu nodo con el apache e instalar el webalizer para analizar sus visitas, etc

Como DNS2GO es ahora de pago, yo utilizo NO-IP (<http://www.no-ip.org> <http://www.no-ip.org>)

Para configurar el zebra, lo único que tengo hecho por el momento es editar el `/etc/zebra/daemons` y activar zebra, ospfd y el bgpd... cuando tenga tarjetas lo probaré y diré... (de esto se está encargando Hilario de ValenciaWireless (<http://www.valenciawireless.org> <http://www.valenciawireless.org>...))

Por el momento es todo... con esto tienes un nodo... puedes acceder por SSH a el para configurarlo remotamente (el putty es un buen cliente para Windows, WinSCP para transferencias seguras de archivos ;)), configurarlo por web (Webmin) y ya con eso para empezar está bien...

Respecto a la VPN (pptpd)... Pues antes no... pero ahora ya va a la perfección ;)

Si no te conectas mediante VPN mediante la tarjeta de la wireless (eth3 en mi caso) no hay Internet (excepto por página web al puerto 80), y si te validas sí... así que si tienes poco ancho de banda, puedes controlar si accedes o no desde fuera de tu red

casera... es decir.. gastas la red inalámbrica para acceder al de tu casita y una vez ahí accedes a través de ese a Internet ;)

Sería interesante activar también un Proxy tipo Squid para acelerar la navegación por Internet... tanto por la red local, como para los nodos (puedes restringir por ip's, etc) (yo lo tengo puesto además con el adzapper, que me permite eliminar muchísimos anuncios de las webs por las que navego)

¡¡Un saludo y suerte!!

Pablo

PD: Si alguien quiere ver fotos del server, teclado, etc que mande un mail para ver si a mucha gente le interesa o no :)

Capítulo 8

Ficheros de configuración

8.1. Squid

```
# WELCOME TO SQUID 2
# -----
#
# This is the default Squid configuration file. You may wish
# to look at the Squid home page (http://www.squid-cache.org/)
# for the FAQ and other documentation.
#
# The default Squid config file shows what the defaults for
# various options happen to be. If you don't need to change the
# default, you shouldn't uncomment the line. Doing so may cause
# run-time problems. In some cases "none" refers to no default
# setting at all, while in other cases it refers to a valid
# option - the comments for that keyword indicate if this is the
# case.
#

# NETWORK OPTIONS
# -----

# TAG: http_port
# Usage: port
# hostname:port
# 1.2.3.4:port
#
# The socket addresses where Squid will listen for HTTP client
# requests. You may specify multiple socket addresses.
# There are three forms: port alone, hostname with port, and
# IP address with port. If you specify a hostname or IP
# address, then Squid binds the socket to that specific
# address. This replaces the old 'tcp_incoming_address'
```

```
# option. Most likely, you do not need to bind to a specific
# address, so you can use the port number alone.
#
# The default port number is 3128.
#
# If you are running Squid in accelerator mode, then you
# probably want to listen on port 80 also, or instead.
#
# The -a command line option will override the *first* port
# number listed here. That option will NOT override an IP
# address, however.
#
# You may specify multiple socket addresses on multiple lines.
#
#Default:
# http_port 3128

# TAG: icp_port
# The port number where Squid sends and receives ICP queries to
# and from neighbor caches. Default is 3130. To disable use
# "0". May be overridden with -u on the command line.
#
#Default:
# icp_port 3130

# TAG: htcp_port
# The port number where Squid sends and receives HTCP queries to
# and from neighbor caches. To turn it on you want to set it 4827.
# By default it is set to "0" (disabled).
#
# To enable this option, you must use --enable-htcp with the
# configure script.
#
#Default:
# htcp_port 0

# TAG: mcast_groups
# This tag specifies a list of multicast groups which your server
# should join to receive multicasted ICP queries.
#
# NOTE! Be very careful what you put here! Be sure you
# understand the difference between an ICP _query_ and an ICP
# _reply_. This option is to be set only if you want to RECEIVE
# multicast queries. Do NOT set this option to SEND multicast
# ICP (use cache_peer for that). ICP replies are always sent via
# unicast, so this option does not affect whether or not you will
# receive replies from multicast group members.
#
# You must be very careful to NOT use a multicast address which
# is already in use by another group of caches.
```

```
#
# If you are unsure about multicast, please read the Multicast
# chapter in the Squid FAQ (http://www.squid-cache.org/FAQ/).
#
# Usage: mcast_groups 239.128.16.128 224.0.1.20
#
# By default, Squid doesn't listen on any multicast groups.
#
#Default:
# none

# TAG: tcp_outgoing_address
# TAG: udp_incoming_address
# TAG: udp_outgoing_address
# Usage: tcp_incoming_address 10.20.30.40
#        udp_outgoing_address fully.qualified.domain.name
#
# tcp_outgoing_address is used for connections made to remote
# servers and other caches.
# udp_incoming_address is used for the ICP socket receiving packets
# from other caches.
# udp_outgoing_address is used for ICP packets sent out to other
# caches.
#
# The default behavior is to not bind to any specific address.
#
# A *_incoming_address value of 0.0.0.0 indicates that Squid should
# listen on all available interfaces.
#
# If udp_outgoing_address is set to 255.255.255.255 (the default)
# then it will use the same socket as udp_incoming_address. Only
# change this if you want to have ICP queries sent using another
# address than where this Squid listens for ICP queries from other
# caches.
#
# NOTE, udp_incoming_address and udp_outgoing_address can not
# have the same value since they both use port 3130.
#
# NOTE, tcp_incoming_address has been removed. You can now
# specify IP addresses on the 'http_port' line.
#
#Default:
# tcp_outgoing_address 255.255.255.255
# udp_incoming_address 0.0.0.0
# udp_outgoing_address 255.255.255.255

# OPTIONS WHICH AFFECT THE NEIGHBOR SELECTION ALGORITHM
# -----
```

```

# TAG: cache_peer
# To specify other caches in a hierarchy, use the format:
#
# cache_peer hostname type http_port icp_port
#
# For example,
#
# #
# #           proxy icp
# #           hostname      type    port  port  options
# #           -----
# cache_peer parent.foo.net      parent  3128  3130  [proxy-only]
# cache_peer sib1.foo.net        sibling  3128  3130  [proxy-only]
# cache_peer sib2.foo.net        sibling  3128  3130  [proxy-only]
#
#           type: either 'parent', 'sibling', or 'multicast'.
#
# proxy_port: The port number where the cache listens for proxy
# requests.
#
# icp_port: Used for querying neighbor caches about
# objects. To have a non-ICP neighbor
# specify '7' for the ICP port and make sure the
# neighbor machine has the UDP echo port
# enabled in its /etc/inetd.conf file.
#
# options: proxy-only
# weight=n
# ttl=n
# no-query
# default
# round-robin
# multicast-responder
# closest-only
# no-digest
# no-netdb-exchange
# no-delay
# login=user:password
# connect-timeout=nn
# digest-url=url
# allow-miss
#
# use 'proxy-only' to specify that objects fetched
# from this cache should not be saved locally.
#
# use 'weight=n' to specify a weighted parent.
# The weight must be an integer. The default weight
# is 1, larger weights are favored more.
#
# use 'ttl=n' to specify a IP multicast TTL to use
# when sending an ICP queries to this address.

```

```
# Only useful when sending to a multicast group.
# Because we don't accept ICP replies from random
# hosts, you must configure other group members as
# peers with the 'multicast-responder' option below.
#
# use 'no-query' to NOT send ICP queries to this
# neighbor.
#
# use 'default' if this is a parent cache which can
# be used as a "last-resort." You should probably
# only use 'default' in situations where you cannot
# use ICP with your parent cache(s).
#
# use 'round-robin' to define a set of parents which
# should be used in a round-robin fashion in the
# absence of any ICP queries.
#
# 'multicast-responder' indicates that the named peer
# is a member of a multicast group. ICP queries will
# not be sent directly to the peer, but ICP replies
# will be accepted from it.
#
# 'closest-only' indicates that, for ICP_OP_MISS
# replies, we'll only forward CLOSEST_PARENT_MISSES
# and never FIRST_PARENT_MISSES.
#
# use 'no-digest' to NOT request cache digests from
# this neighbor.
#
# 'no-netdb-exchange' disables requesting ICMP
# RTT database (NetDB) from the neighbor.
#
# use 'no-delay' to prevent access to this neighbor
# from influencing the delay pools.
#
# use 'login=user:password' if this is a personal/workgroup
# proxy and your parent requires proxy authentication.
#
# use 'connect-timeout=nn' to specify a peer
# specific connect timeout (also see the
# peer_connect_timeout directive)
#
# use 'digest-url=url' to tell Squid to fetch the cache
# digest (if digests are enabled) for this host from
# the specified URL rather than the Squid default
# location.
#
# use 'allow-miss' to disable Squid's use of only-if-cached
# when forwarding requests to siblings. This is primarily
# useful when icp_hit_stale is used by the sibling. To
```

```
#     extensive use of this option may result in forwarding
#     loops, and you should avoid having two-way peerings
#     with this option. (for example to deny peer usage on
#     requests from peer by denying cache_peer_access if the
#     source is a peer)
#
# NOTE: non-ICP neighbors must be specified as 'parent'.
#
#Default:
# none

# TAG: cache_peer_domain
# Use to limit the domains for which a neighbor cache will be
# queried. Usage:
#
# cache_peer_domain cache-host domain [domain ...]
# cache_peer_domain cache-host !domain
#
# For example, specifying
#
# cache_peer_domain parent.foo.net .edu
#
# has the effect such that UDP query packets are sent to
# 'bigserver' only when the requested object exists on a
# server in the .edu domain. Prefixing the domainname
# with '!' means that the cache will be queried for objects
# NOT in that domain.
#
# NOTE: * Any number of domains may be given for a cache-host,
#       either on the same or separate lines.
# * When multiple domains are given for a particular
#   cache-host, the first matched domain is applied.
# * Cache hosts with no domain restrictions are queried
#   for all requests.
# * There are no defaults.
# * There is also a 'cache_peer_access' tag in the ACL
#   section.
#
#Default:
# none

# TAG: neighbor_type_domain
# usage: neighbor_type_domain parent|sibling domain domain ...
#
# Modifying the neighbor type for specific domains is now
# possible. You can treat some domains differently than the the
# default neighbor type specified on the 'cache_peer' line.
# Normally it should only be necessary to list domains which
# should be treated differently because the default neighbor type
# applies for hostnames which do not match domains listed here.
```

```
#
#EXAMPLE:
# cache_peer parent cache.foo.org 3128 3130
# neighbor_type_domain cache.foo.org sibling .com .net
# neighbor_type_domain cache.foo.org sibling .au .de
#
#Default:
# none

# TAG: icp_query_timeout (msec)
# Normally Squid will automatically determine an optimal ICP
# query timeout value based on the round-trip-time of recent ICP
# queries. If you want to override the value determined by
# Squid, set this 'icp_query_timeout' to a non-zero value. This
# value is specified in MILLISECONDS, so, to use a 2-second
# timeout (the old default), you would write:
#
# icp_query_timeout 2000
#
#Default:
# icp_query_timeout 0

# TAG: maximum_icp_query_timeout (msec)
# Normally the ICP query timeout is determined dynamically. But
# sometimes it can lead to very large values (say 5 seconds).
# Use this option to put an upper limit on the dynamic timeout
# value. Do NOT use this option to always use a fixed (instead
# of a dynamic) timeout value. To set a fixed timeout see the
# 'icp_query_timeout' directive.
#
#Default:
# maximum_icp_query_timeout 2000

# TAG: mcast_icp_query_timeout (msec)
# For Multicast peers, Squid regularly sends out ICP "probes" to
# count how many other peers are listening on the given multicast
# address. This value specifies how long Squid should wait to
# count all the replies. The default is 2000 msec, or 2
# seconds.
#
#Default:
# mcast_icp_query_timeout 2000

# TAG: dead_peer_timeout (seconds)
# This controls how long Squid waits to declare a peer cache
# as "dead." If there are no ICP replies received in this
# amount of time, Squid will declare the peer dead and not
# expect to receive any further ICP replies. However, it
# continues to send ICP queries, and will mark the peer as
# alive upon receipt of the first subsequent ICP reply.
```

```
#
# This timeout also affects when Squid expects to receive ICP
# replies from peers.  If more than 'dead_peer' seconds have
# passed since the last ICP reply was received, Squid will not
# expect to receive an ICP reply on the next query.  Thus, if
# your time between requests is greater than this timeout, you
# will see a lot of requests sent DIRECT to origin servers
# instead of to your parents.
#
#Default:
# dead_peer_timeout 10 seconds

# TAG: hierarchy_stoplist
# A list of words which, if found in a URL, cause the object to
# be handled directly by this cache.  In other words, use this
# to not query neighbor caches for certain objects.  You may
# list this option multiple times.
#
#We recommend you to use at least the following line.
hierarchy_stoplist cgi-bin ?

# TAG: no_cache
# A list of ACL elements which, if matched, cause the reply to
# immediately removed from the cache.  In other words, use this
# to force certain objects to never be cached.
#
# You must use the word 'DENY' to indicate the ACL names which should
# NOT be cached.
#
#We recommend you to use the following two lines.
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY

# OPTIONS WHICH AFFECT THE CACHE SIZE
# -----

# TAG: cache_mem (bytes)
# NOTE: THIS PARAMETER DOES NOT SPECIFY THE MAXIMUM PROCESS SIZE.
# IT ONLY PLACES A LIMIT ON HOW MUCH ADDITIONAL MEMORY SQUID WILL
# USE AS A MEMORY CACHE OF OBJECTS.  SQUID USES MEMORY FOR OTHER
# THINGS AS WELL.  SEE THE SQUID FAQ SECTION 8 FOR DETAILS.
#
# 'cache_mem' specifies the ideal amount of memory to be used
# for:
# * In-Transit objects
# * Hot Objects
# * Negative-Cached objects
#
# Data for these objects are stored in 4 KB blocks.  This
```

```
# parameter specifies the ideal upper limit on the total size of
# 4 KB blocks allocated. In-Transit objects take the highest
# priority.
#
# In-transit objects have priority over the others. When
# additional space is needed for incoming data, negative-cached
# and hot objects will be released. In other words, the
# negative-cached and hot objects will fill up any unused space
# not needed for in-transit objects.
#
# If circumstances require, this limit will be exceeded.
# Specifically, if your incoming request rate requires more than
# 'cache_mem' of memory to hold in-transit objects, Squid will
# exceed this limit to satisfy the new requests. When the load
# decreases, blocks will be freed until the high-water mark is
# reached. Thereafter, blocks will be used to store hot
# objects.
#
#Default:
# cache_mem 8 MB

# TAG: cache_swap_low (percent, 0-100)
# TAG: cache_swap_high (percent, 0-100)
#
# The low- and high-water marks for cache object replacement.
# Replacement begins when the swap (disk) usage is above the
# low-water mark and attempts to maintain utilization near the
# low-water mark. As swap utilization gets close to high-water
# mark object eviction becomes more aggressive. If utilization is
# close to the low-water mark less replacement is done each time.
#
# Defaults are 90% and 95%. If you have a large cache, 5% could be
# hundreds of MB. If this is the case you may wish to set these
# numbers closer together.
#
#Default:
# cache_swap_low 90
# cache_swap_high 95

# TAG: maximum_object_size (bytes)
# Objects larger than this size will NOT be saved on disk. The
# value is specified in kilobytes, and the default is 4MB. If
# you wish to get a high BYTES hit ratio, you should probably
# increase this (one 32 MB object hit counts for 3200 10KB
# hits). If you wish to increase speed more than your want to
# save bandwidth you should leave this low.
#
# NOTE: if using the LFUDA replacement policy you should increase
# this value to maximize the byte hit rate improvement of LFUDA!
# See replacement_policy below for a discussion of this policy.
```

```
#
#Default:
maximum_object_size 32768 KB

# TAG: minimum_object_size (bytes)
# Objects smaller than this size will NOT be saved on disk. The
# value is specified in kilobytes, and the default is 0 KB, which
# means there is no minimum.
#
#Default:
# minimum_object_size 0 KB

# TAG: maximum_object_size_in_memory (bytes)
# Objects greater than this size will not be attempted to kept in
# the memory cache. This should be set high enough to keep objects
# accessed frequently in memory to improve performance whilst low
# enough to keep larger objects from hoarding cache_mem .
#
#Default:
# maximum_object_size_in_memory 8 KB

# TAG: ipcache_size (number of entries)
# TAG: ipcache_low (percent)
# TAG: ipcache_high (percent)
# The size, low-, and high-water marks for the IP cache.
#
#Default:
# ipcache_size 1024
# ipcache_low 90
# ipcache_high 95

# TAG: fqdn_cache_size (number of entries)
# Maximum number of FQDN cache entries.
#
#Default:
# fqdn_cache_size 1024

# TAG: cache_replacement_policy
# The cache replacement policy parameter determines which
# objects are evicted (replaced) when disk space is needed.
#
# lru : Squid's original list based LRU policy
# heap GDSF : Greedy-Dual Size Frequency
# heap LFUDA: Least Frequently Used with Dynamic Aging
# heap LRU : LRU policy implemented using a heap
#
# Applies to any cache_dir lines listed below this.
#
# The LRU policies keeps recently referenced objects.
#
```

```
# The heap GDSF policy optimizes object hit rate by keeping smaller
# popular objects in cache so it has a better chance of getting a
# hit. It achieves a lower byte hit rate than LFUDA though since
# it evicts larger (possibly popular) objects.
#
# The heap LFUDA policy keeps popular objects in cache regardless of
# their size and thus optimizes byte hit rate at the expense of
# hit rate since one large, popular object will prevent many
# smaller, slightly less popular objects from being cached.
#
# Both policies utilize a dynamic aging mechanism that prevents
# cache pollution that can otherwise occur with frequency-based
# replacement policies.
#
# NOTE: if using the LFUDA replacement policy you should increase
# the value of maximum_object_size above its default of 4096 KB to
# to maximize the potential byte hit rate improvement of LFUDA.
#
# For more information about the GDSF and LFUDA cache replacement
# policies see http://www.hpl.hp.com/techreports/1999/HPL-1999-69.html
# and http://fog.hpl.external.hp.com/techreports/98/HPL-98-173.html.
#
#Default:
cache_replacement_policy heap LFUDA

# TAG: memory_replacement_policy
# The memory replacement policy parameter determines which
# objects are purged from memory when memory space is needed.
#
# See cache_replacement_policy for details.
#
#Default:
# memory_replacement_policy lru

# LOGFILE PATHNAMES AND CACHE DIRECTORIES
# -----

# TAG: cache_dir
# Usage:
#
# cache_dir Type Directory-Name Fs-specific-data [options]
#
# You can specify multiple cache_dir lines to spread the
# cache among different disk partitions.
#
# Type specifies the kind of storage system to use. Most
# everyone will want to use "ufs" as the type. If you are using
# Async I/O (--enable async-io) on Linux or Solaris, then you may
# want to try "aufs" as the type. Async IO support may be
```

```
# buggy, however, so beware.
#
# 'Directory' is a top-level directory where cache swap
# files will be stored. If you want to use an entire disk
# for caching, then this can be the mount-point directory.
# The directory must exist and be writable by the Squid
# process. Squid will NOT create this directory for you.
#
# The ufs store type:
#
# "ufs" is the old well-known Squid storage format that has always
# been there.
#
# cache_dir ufs Directory-Name Mbytes L1 L2 [options]
#
# 'Mbytes' is the amount of disk space (MB) to use under this
# directory. The default is 100 MB. Change this to suit your
# configuration.
#
# 'Level-1' is the number of first-level subdirectories which
# will be created under the 'Directory'. The default is 16.
#
# 'Level-2' is the number of second-level subdirectories which
# will be created under each first-level directory. The default
# is 256.
#
# The aufs store type:
#
# "aufs" uses the same storage format as "ufs", utilizing
# POSIX-threads to avoid blocking the main Squid process on
# disk-I/O. This was formerly known in Squid as async-io.
#
# cache_dir aufs Directory-Name Mbytes L1 L2 [options]
#
# see argument descriptions under ufs above
#
# The diskd store type:
#
# "diskd" uses the same storage format as "ufs", utilizing a
# separate process to avoid blocking the main Squid process on
# disk-I/O.
#
# cache_dir diskd Directory-Name Mbytes L1 L2 [options] [Q1=n] [Q2=n]
#
# see argument descriptions under ufs above
#
# Q1 specifies the number of unacknowledged I/O requests when Squid
# stops opening new files. If this many messages are in the queues,
# Squid won't open new files. Default is 64
#
```

```
# Q2 specifies the number of unacknowledged messages when Squid
# starts blocking.  If this many messages are in the queues,
# Squid blocks until it receives some replies. Default is 72
#
# Common options:
#
# read-only, this cache_dir is read only.
#
# max-size=n, refers to the max object size this storidir supports.
# It is used to initially choose the storidir to dump the object.
# Note: To make optimal use of the max-size limits you should order
# the cache_dir lines with the smallest max-size value first and the
# ones with no max-size specification last.
#
#Default:
cache_dir ufs /var/spool/squid 512 16 256

# TAG: cache_access_log
# Logs the client request activity.  Contains an entry for
# every HTTP and ICP queries received.
#
#Default:
# cache_access_log /var/log/squid/access.log

# TAG: cache_log
# Cache logging file. This is where general information about
# your cache's behavior goes. You can increase the amount of data
# logged to this file with the "debug_options" tag below.
#
#Default:
# cache_log /var/log/squid/cache.log

# TAG: cache_store_log
# Logs the activities of the storage manager.  Shows which
# objects are ejected from the cache, and which objects are
# saved and for how long.  To disable, enter "none". There are
# not really utilities to analyze this data, so you can safely
# disable it.
#
#Default:
# cache_store_log /var/log/squid/store.log

# TAG: cache_swap_log
# Location for the cache "swap.log."  This log file holds the
# metadata of objects saved on disk.  It is used to rebuild the
# cache during startup.  Normally this file resides in each
# 'cache_dir' directory, but you may specify an alternate
# pathname here.  Note you must give a full filename, not just
# a directory.  Since this is the index for the whole object
# list you CANNOT periodically rotate it!
```

```
#
# If %s can be used in the file name then it will be replaced with a
# a representation of the cache_dir name where each / is replaced
# with '.'. This is needed to allow adding/removing cache_dir
# lines when cache_swap_log is being used.
#
# If have more than one 'cache_dir', and %s is not used in the name
# then these swap logs will have names such as:
#
# cache_swap_log.00
# cache_swap_log.01
# cache_swap_log.02
#
# The numbered extension (which is added automatically)
# corresponds to the order of the 'cache_dir' lines in this
# configuration file. If you change the order of the 'cache_dir'
# lines in this file, then these log files will NOT correspond to
# the correct 'cache_dir' entry (unless you manually rename
# them). We recommend that you do NOT use this option. It is
# better to keep these log files in each 'cache_dir' directory.
#
#Default:
# none

# TAG: emulate_httpd_log on|off
# The Cache can emulate the log file format which many 'httpd'
# programs use. To disable/enable this emulation, set
# emulate_httpd_log to 'off' or 'on'. The default
# is to use the native log format since it includes useful
# information that Squid-specific log analyzers use.
#
#Default:
# emulate_httpd_log off

# TAG: log_ip_on_direct on|off
# Log the destination IP address in the hierarchy log tag when going
# direct. Earlier Squid versions logged the hostname here. If you
# prefer the old way set this to off.
#
#Default:
# log_ip_on_direct on

# TAG: mime_table
# Pathname to Squid's MIME table. You shouldn't need to change
# this, but the default file contains examples and formatting
# information if you do.
#
#Default:
# mime_table /usr/lib/squid/mime.conf
```

```
# TAG: log_mime_hdrs on|off
# The Cache can record both the request and the response MIME
# headers for each HTTP transaction. The headers are encoded
# safely and will appear as two bracketed fields at the end of
# the access log (for either the native or httpd-emulated log
# formats). To enable this logging set log_mime_hdrs to 'on'.
#
#Default:
# log_mime_hdrs off

# TAG: useragent_log
# Squid will write the User-Agent field from HTTP requests
# to the filename specified here. By default useragent_log
# is disabled.
#
#Default:
# none

# TAG: referer_log
# Squid will write the Referer field from HTTP requests to the
# filename specified here. By default referer_log is disabled.
#
#Default:
# none

# TAG: pid_filename
# A filename to write the process-id to. To disable, enter "none".
#
#Default:
# pid_filename /var/run/squid.pid

# TAG: debug_options
# Logging options are set as section,level where each source file
# is assigned a unique section. Lower levels result in less
# output, Full debugging (level 9) can result in a very large
# log file, so be careful. The magic word "ALL" sets debugging
# levels for all sections. We recommend normally running with
# "ALL,1".
#
#Default:
# debug_options ALL,1

# TAG: log_fqdn on|off
# Turn this on if you wish to log fully qualified domain names
# in the access.log. To do this Squid does a DNS lookup of all
# IP's connecting to it. This can (in some situations) increase
# latency, which makes your cache seem slower for interactive
# browsing.
#
#Default:
```

```
# log_fqdn off

# TAG: client_netmask
# A netmask for client addresses in logfiles and cachemgr output.
# Change this to protect the privacy of your cache clients.
# A netmask of 255.255.255.0 will log all IP's in that range with
# the last digit set to '0'.
#
#Default:
# client_netmask 255.255.255.255

# OPTIONS FOR EXTERNAL SUPPORT PROGRAMS
# -----

# TAG: ftp_user
# If you want the anonymous login password to be more informative
# (and enable the use of picky ftp servers), set this to something
# reasonable for your domain, like wwwuser@somewhere.net
#
# The reason why this is domainless by default is that the
# request can be made on the behalf of a user in any domain,
# depending on how the cache is used.
# Some ftp server also validate that the email address is valid
# (for example perl.com).
#
#Default:
# ftp_user Squid@

# TAG: ftp_list_width
# Sets the width of ftp listings. This should be set to fit in
# the width of a standard browser. Setting this too small
# can cut off long filenames when browsing ftp sites.
#
#Default:
# ftp_list_width 32

# TAG: ftp_passive
# If your firewall does not allow Squid to use passive
# connections, then turn off this option.
#
#Default:
# ftp_passive on

# TAG: ftp_sanitycheck
# For security and data integrity reasons Squid by default performs
# sanity checks of the addresses of FTP data connections ensure the
# data connection is to the requested server. If you need to allow
# FTP connections to servers using another IP address for the data
# connection then turn this off.
```

```
#
#Default:
# ftp_sanitycheck on

# TAG: cache_dns_program
# Note: This option is only available if Squid is rebuilt with the
#       --disable-internal-dns option
#
# Specify the location of the executable for dnslookup process.
#
#Default:
# cache_dns_program /usr/lib/squid/

# TAG: dns_children
# Note: This option is only available if Squid is rebuilt with the
#       --disable-internal-dns option
#
# The number of processes spawn to service DNS name lookups.
# For heavily loaded caches on large servers, you should
# probably increase this value to at least 10. The maximum
# is 32. The default is 5.
#
# You must have at least one dnsserver process.
#
#Default:
# dns_children 5

# TAG: dns_retransmit_interval
# Initial retransmit interval for DNS queries. The interval is
# doubled each time all configured DNS servers have been tried.
#
#
#Default:
# dns_retransmit_interval 5 seconds

# TAG: dns_timeout
# DNS Query timeout. If no response is received to a DNS query
# within this time then all DNS servers for the queried domain
# is assumed to be unavailable.
#
#Default:
# dns_timeout 5 minutes

# TAG: dns_defnames on|off
# Note: This option is only available if Squid is rebuilt with the
#       --disable-internal-dns option
#
# Normally the 'dnsserver' disables the RES_DEFNAMES resolver
# option (see res_init(3)). This prevents caches in a hierarchy
# from interpreting single-component hostnames locally. To allow
```

```
# dnsserver to handle single-component names, enable this
# option.
#
#Default:
# dns_defnames off

# TAG: dns_nameservers
# Use this if you want to specify a list of DNS name servers
# (IP addresses) to use instead of those given in your
# /etc/resolv.conf file.
#
# Example: dns_nameservers 10.0.0.1 192.172.0.4
#
#Default:
# none

# TAG: diskd_program
# Specify the location of the diskd executable.
# Note that this is only useful if you have compiled in
# diskd as one of the store io modules.
#
#Default:
# diskd_program /usr/lib/squid/diskd

# TAG: unlinkd_program
# Specify the location of the executable for file deletion process.
#
#Default:
# unlinkd_program /usr/lib/squid/unlinkd

# TAG: pinger_program
# Note: This option is only available if Squid is rebuilt with the
#      --enable-icmp option
#
# Specify the location of the executable for the pinger process.
# This is only useful if you configured Squid (during compilation)
# with the '--enable-icmp' option.
#
#Default:
# pinger_program /usr/lib/squid/

# TAG: redirect_program
# Specify the location of the executable for the URL redirector.
# Since they can perform almost any function there isn't one included.
# See the Release-Notes for information on how to write one.
# By default, a redirector is not used.
#
#Default:
# none
```

```
redirect_program /usr/lib/squid/squid_redirect

# TAG: redirect_children
# The number of redirector processes to spawn. If you start
# too few Squid will have to wait for them to process a backlog of
# URLs, slowing it down. If you start too many they will use RAM
# and other system resources.
#
#Default:
redirect_children 30

# TAG: redirect_rewrites_host_header
# By default Squid rewrites any Host: header in redirected
# requests. If you are running a accelerator then this may
# not be a wanted effect of a redirector.
#
#Default:
# redirect_rewrites_host_header on

# TAG: redirector_access
# If defined, this access list specifies which requests are
# sent to the redirector processes. By default all requests
# are sent.
#
#Default:
# none

# TAG: authenticate_program
# Specify the command for the external authenticator. Such a
# program reads a line containing "username password" and replies
# "OK" or "ERR" in an endless loop. If you use an authenticator,
# make sure you have 1 acl of type proxy_auth. By default, the
# authenticator_program is not used.
#
# If you want to use the traditional proxy authentication,
# jump over to the ../auth_modules/NCSA directory and
# type:
# % make
# % make install
#
# Then, set this line to something like
#
# authenticate_program /usr/bin/ncsa_auth /usr/etc/passwd
#
#Default:
# none

# TAG: authenticate_children
# The number of authenticator processes to spawn (default 5). If you
# start too few Squid will have to wait for them to process a backlog
```

```
# of usercode/password verifications, slowing it down. When password
# verifications are done via a (slow) network you are likely to need
# lots of authenticator processes.
#
#Default:
# authenticate_children 5

# TAG: authenticate_ttl
# The time a checked username/password combination remains cached.
# If a wrong password is given for a cached user, the user gets
# removed from the username/password cache forcing a revalidation.
#
#Default:
# authenticate_ttl 1 hour

# TAG: authenticate_ip_ttl
# With this option you control how long a proxy authentication
# will be bound to a specific IP address. If a request using
# the same user name is received during this time then access
# will be denied and both users are required to reauthenticate
# them selves. The idea behind this is to make it annoying
# for people to share their password to their friends, but
# yet allow a dialup user to reconnect on a different dialup
# port.
#
# The default is 0 to disable the check. Recommended value
# if you have dialup users are no more than 60 seconds to allow
# the user to redial without hassle. If all your users are
# stationary then higher values may be used.
#
# See also authenticate_ip_ttl_is_strict
#
#Default:
# authenticate_ip_ttl 0 seconds

# TAG: authenticate_ip_ttl_is_strict
# This option makes authenticate_ip_ttl a bit stricted. With this
# enabled authenticate_ip_ttl will deny all access from other IP
# addresses until the TTL has expired, and the IP address "owning"
# the userid will not be forced to reauthenticate.
#
#Default:
# authenticate_ip_ttl_is_strict on

# OPTIONS FOR TUNING THE CACHE
# -----

# TAG: wais_relay_host
# TAG: wais_relay_port
```

```
# Relay WAIS request to host (1st arg) at port (2 arg).
#
#Default:
# wais_relay_port 0

# TAG: request_header_max_size (KB)
# This specifies the maximum size for HTTP headers in a request.
# Request headers are usually relatively small (about 512 bytes).
# Placing a limit on the request header size will catch certain
# bugs (for example with persistent connections) and possibly
# buffer-overflow or denial-of-service attacks.
#
#Default:
# request_header_max_size 10 KB

# TAG: request_body_max_size (KB)
# This specifies the maximum size for an HTTP request body.
# In other words, the maximum size of a PUT/POST request.
# A user who attempts to send a request with a body larger
# than this limit receives an "Invalid Request" error message.
# If you set this parameter to a zero, there will be no limit
# imposed.
#
#Default:
# request_body_max_size 1 MB

# TAG: reply_body_max_size (KB)
# This option specifies the maximum size of a reply body. It
# can be used to prevent users from downloading very large files,
# such as MP3's and movies. The reply size is checked twice.
# First when we get the reply headers, we check the
# content-length value. If the content length value exists and
# is larger than this parameter, the request is denied and the
# user receives an error message that says "the request or reply
# is too large." If there is no content-length, and the reply
# size exceeds this limit, the client's connection is just closed
# and they will receive a partial reply.
#
# NOTE: downstream caches probably can not detect a partial reply
# if there is no content-length header, so they will cache
# partial responses and give them out as hits. You should NOT
# use this option if you have downstream caches.
#
# If you set this parameter to zero (the default), there will be
# no limit imposed.
#
#Default:
# reply_body_max_size 0

# TAG: refresh_pattern
```

```
# usage: refresh_pattern [-i] regex min percent max [options]
#
# By default, regular expressions are CASE-SENSITIVE. To make
# them case-insensitive, use the -i option.
#
# 'Min' is the time (in minutes) an object without an explicit
# expiry time should be considered fresh. The recommended
# value is 0, any higher values may cause dynamic applications
# to be erroneously cached unless the application designer
# has taken the appropriate actions.
#
# 'Percent' is a percentage of the objects age (time since last
# modification age) an object without explicit expiry time
# will be considered fresh.
#
# 'Max' is an upper limit on how long objects without an explicit
# expiry time will be considered fresh.
#
# options: override-expire
#          override-lastmod
#          reload-into-ims
#          ignore-reload
#
# override-expire enforces min age even if the server
# sent a Expires: header. Doing this VIOLATES the HTTP
# standard. Enabling this feature could make you liable
# for problems which it causes.
#
# override-lastmod enforces min age even on objects
# that was modified recently.
#
# reload-into-ims changes client no-cache or ''reload''
# to If-Modified-Since requests. Doing this VIOLATES the
# HTTP standard. Enabling this feature could make you
# liable for problems which it causes.
#
# ignore-reload ignores a client no-cache or ''reload''
# header. Doing this VIOLATES the HTTP standard. Enabling
# this feature could make you liable for problems which
# it causes.
#
# Please see the file doc/Release-Notes-1.1.txt for a full
# description of Squid's refresh algorithm. Basically a
# cached object is: (the order is changed from 1.1.X)
#
# FRESH if expires < now, else STALE
# STALE if age > max
# FRESH if lm-factor < percent, else STALE
# FRESH if age < min
# else STALE
```

```
#
# The refresh_pattern lines are checked in the order listed here.
# The first entry which matches is used.  If none of the entries
# match, then the default will be used.
#
# Note, you must uncomment all the default lines if you want
# to change one.  The default setting is only active if none is
# used.
#
#Default:
# refresh_pattern ^ftp: 1440 20% 10080
# refresh_pattern ^gopher: 1440 0% 1440
# refresh_pattern . 0 20% 4320

# TAG: reference_age
# As a part of normal operation, Squid performs Least Recently
# Used removal of cached objects.  The LRU age for removal is
# computed dynamically, based on the amount of disk space in
# use.  The dynamic value can be seen in the Cache Manager 'info'
# output.
#
# The 'reference_age' parameter defines the maximum LRU age.  For
# example, setting reference_age to '1 week' will cause objects
# to be removed if they have not been accessed for a week or
# more.  The default value is one year.
#
# Specify a number here, followed by units of time.  For example:
# 1 week
# 3.5 days
# 4 months
# 2.2 hours
#
# NOTE: this parameter is not used when using the enhanced
# replacement policies, GDSH or LFUDA.
#
#Default:
# reference_age 1 year

# TAG: quick_abort_min (KB)
# TAG: quick_abort_max (KB)
# TAG: quick_abort_pct (percent)
# The cache can be configured to continue downloading aborted
# requests.  This may be undesirable on slow (e.g. SLIP) links
# and/or very busy caches.  Impatient users may tie up file
# descriptors and bandwidth by repeatedly requesting and
# immediately aborting downloads.
#
# When the user aborts a request, Squid will check the
# quick_abort values to the amount of data transfered until
# then.
```

```
#
# If the transfer has less than 'quick_abort_min' KB remaining,
# it will finish the retrieval. Setting 'quick_abort_min' to -1
# will disable the quick_abort feature.
#
# If the transfer has more than 'quick_abort_max' KB remaining,
# it will abort the retrieval.
#
# If more than 'quick_abort_pct' of the transfer has completed,
# it will finish the retrieval.
#
#Default:
# quick_abort_min 16 KB
# quick_abort_max 16 KB
# quick_abort_pct 95

# TAG: negative_ttl time-units
# Time-to-Live (TTL) for failed requests. Certain types of
# failures (such as "connection refused" and "404 Not Found") are
# negatively-cached for a configurable amount of time. The
# default is 5 minutes. Note that this is different from
# negative caching of DNS lookups.
#
#Default:
# negative_ttl 5 minutes

# TAG: positive_dns_ttl time-units
# Time-to-Live (TTL) for positive caching of successful DNS lookups.
# Default is 6 hours (360 minutes). If you want to minimize the
# use of Squid's ipcache, set this to 1, not 0.
#
#Default:
# positive_dns_ttl 6 hours

# TAG: negative_dns_ttl time-units
# Time-to-Live (TTL) for negative caching of failed DNS lookups.
#
#Default:
# negative_dns_ttl 5 minutes

# TAG: range_offset_limit (bytes)
# Sets a upper limit on how far into the the file a Range request
# may be to cause Squid to prefetch the whole file. If beyond this
# limit then Squid forwards the Range request as it is and the result
# is NOT cached.
#
# This is to stop a far ahead range request (lets say start at 17MB)
# from making Squid fetch the whole object up to that point before
# sending anything to the client.
#
```

```
# A value of -1 causes Squid to always fetch the object from the
# beginning so that it may cache the result. (2.0 style)
#
# A value of 0 causes Squid to never fetch more than the
# client requested. (default)
#
#Default:
# range_offset_limit 0 KB

# TIMEOUTS
# -----

# TAG: connect_timeout time-units
# Some systems (notably Linux) can not be relied upon to properly
# time out connect(2) requests. Therefore the Squid process
# enforces its own timeout on server connections. This parameter
# specifies how long to wait for the connect to complete. The
# default is two minutes (120 seconds).
#
#Default:
# connect_timeout 2 minutes

# TAG: peer_connect_timeout time-units
# This parameter specifies how long to wait for a pending TCP
# connection to a peer cache. The default is 30 seconds. You
# may also set different timeout values for individual neighbors
# with the 'connect-timeout' option on a 'cache_peer' line.
#
#Default:
# peer_connect_timeout 30 seconds

# TAG: siteselect_timeout time-units
# For URN to multiple URL's URL selection
#
#Default:
# siteselect_timeout 4 seconds

# TAG: read_timeout time-units
# The read_timeout is applied on server-side connections. After
# each successful read(), the timeout will be extended by this
# amount. If no data is read again after this amount of time,
# the request is aborted and logged with ERR_READ_TIMEOUT. The
# default is 15 minutes.
#
#Default:
# read_timeout 15 minutes

# TAG: request_timeout
# How long to wait for an HTTP request after connection
```

```
# establishment. For persistent connections, wait this long
# after the previous request completes.
#
#Default:
# request_timeout 30 seconds

# TAG: client_lifetime time-units
# The maximum amount of time that a client (browser) is allowed to
# remain connected to the cache process. This protects the Cache
# from having a lot of sockets (and hence file descriptors) tied up
# in a CLOSE_WAIT state from remote clients that go away without
# properly shutting down (either because of a network failure or
# because of a poor client implementation). The default is one
# day, 1440 minutes.
#
# NOTE: The default value is intended to be much larger than any
# client would ever need to be connected to your cache. You
# should probably change client_lifetime only as a last resort.
# If you seem to have many client connections tying up
# filedescriptors, we recommend first tuning the read_timeout,
# request_timeout, pconn_timeout and quick_abort values.
#
#Default:
# client_lifetime 1 day

# TAG: half_closed_clients
# Some clients may shutdown the sending side of their TCP
# connections, while leaving their receiving sides open. Sometimes,
# Squid can not tell the difference between a half-closed and a
# fully-closed TCP connection. By default, half-closed client
# connections are kept open until a read(2) or write(2) on the
# socket returns an error. Change this option to 'off' and Squid
# will immediately close client connections when read(2) returns
# "no more data to read."
#
#Default:
# half_closed_clients on

# TAG: pconn_timeout
# Timeout for idle persistent connections to servers and other
# proxies.
#
#Default:
# pconn_timeout 120 seconds

# TAG: ident_timeout
# Maximum time to wait for IDENT requests. If this is too high,
# and you enabled 'ident_lookup', then you might be susceptible
# to denial-of-service by having many ident requests going at
# once.
```

```

#
# Only src type ACL checks are fully supported. A src_domain
# ACL might work at times, but it will not always provide
# the correct result.
#
# This option may be disabled by using --disable-ident with
# the configure script.
#
#Default:
# ident_timeout 10 seconds

# TAG: shutdown_lifetime time-units
# When SIGTERM or SIGHUP is received, the cache is put into
# "shutdown pending" mode until all active sockets are closed.
# This value is the lifetime to set for all open descriptors
# during shutdown mode. Any active clients after this many
# seconds will receive a 'timeout' message.
#
#Default:
# shutdown_lifetime 30 seconds

# ACCESS CONTROLS
# -----

# TAG: acl
# Defining an Access List
#
# acl aclname acltype string1 ...
# acl aclname acltype "file" ...
#
# when using "file", the file should contain one item per line
#
# acltype is one of src dst srcdomain dstdomain url_pattern
# urlpath_pattern time port proto method browser user
#
# By default, regular expressions are CASE-SENSITIVE. To make
# them case-insensitive, use the -i option.
#
# acl aclname src      ip-address/netmask ... (clients IP address)
# acl aclname src      addr1-addr2/netmask ... (range of addresses)
# acl aclname dst      ip-address/netmask ... (URL host's IP address)
# acl aclname myip     ip-address/netmask ... (local socket IP address)
#
# acl aclname srcdomain .foo.com ... # reverse lookup, client IP
# acl aclname dstdomain .foo.com ... # Destination server from URL
# acl aclname srcdom_regex [-i] xxx ... # regex matching client name
# acl aclname dstdom_regex [-i] xxx ... # regex matching server
# # For dstdomain and dstdom_regex a reverse lookup is tried if a IP
# # based URL is used. The name "none" is used if the reverse lookup

```

```
# # fails.
#
# acl aclname time [day-abbrevs] [h1:m1-h2:m2]
# day-abbrevs:
# S - Sunday
# M - Monday
# T - Tuesday
# W - Wednesday
# H - Thursday
# F - Friday
# A - Saturday
# h1:m1 must be less than h2:m2
# acl aclname url_regex [-i] ^http:// ... # regex matching on whole URL
# acl aclname urlpath_regex [-i] \.gif$ ... # regex matching on URL path
# acl aclname port 80 70 21 ...
# acl aclname port 0-1024 ... # ranges allowed
# acl aclname myport 3128 ... # (local socket TCP port)
# acl aclname proto HTTP FTP ...
# acl aclname method GET POST ...
# acl aclname browser [-i] regexp
# # pattern match on User-Agent header
# acl aclname ident username ...
# acl aclname ident_regex [-i] pattern ...
# # string match on ident output.
# # use REQUIRED to accept any non-null ident.
# acl aclname src_as number ...
# acl aclname dst_as number ...
# # Except for access control, AS numbers can be used for
# # routing of requests to specific caches. Here's an
# # example for routing all requests for AS#1241 and only
# # those to mycache.mydomain.net:
# # acl asexample dst_as 1241
# # cache_peer_access mycache.mydomain.net allow asexample
# # cache_peer_access mycache_mydomain.net deny all
#
# acl aclname proxy_auth username ...
# acl aclname proxy_auth_regex [-i] pattern ...
# # list of valid usernames
# # use REQUIRED to accept any valid username.
# #
# # NOTE: when a Proxy-Authentication header is sent but it is not
# # needed during ACL checking the username is NOT logged
# # in access.log.
# #
# # NOTE: proxy_auth requires a EXTERNAL authentication program
# # to check username/password combinations (see
# # authenticate_program).
# #
# # WARNING: proxy_auth can't be used in a transparent proxy. It
# # collides with any authentication done by origin servers. It may
```

```
# # seem like it works at first, but it doesn't.
#
# acl aclname snmp_community string ...
# # A community string to limit access to your SNMP Agent
# # Example:
# #
# # acl snmppublic snmp_community public
#
# acl aclname maxconn number
# # This will be matched when the client's IP address has
# # more than <number> HTTP connections established.
#
# acl req_mime_type mime-type1 ...
# # regex match againsts the mime type of the request generated
# # by the client. Can be used to detect file upload or some
# # types HTTP tunnelling requests.
# # NOTE: This does NOT match the reply. You cannot use this
# # to match the returned file type.
#
#Examples:
#acl myexample dst_as 1241
#acl password proxy_auth REQUIRED
#acl fileupload req_mime_type -i ^multipart/form-data$
#

acl local src 1.1.0.0/16
acl local2 src 192.168.0.0/16
acl local3 src 10.34.0.0/16

#Recommended minimum configuration:
acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl SSL_ports port 443 563
acl Safe_ports port 80 # http
acl Safe_ports port 21 # ftp
acl Safe_ports port 443 563 # https, snews
acl Safe_ports port 70 # gopher
acl Safe_ports port 210 # wais
acl Safe_ports port 1025-65535 # unregistered ports
acl Safe_ports port 280 # http-mgmt
acl Safe_ports port 488 # gss-http
acl Safe_ports port 591 # filemaker
acl Safe_ports port 631 # cups
acl Safe_ports port 777 # multiling http
acl Safe_ports port 901 # SWAT
acl purge method PURGE
acl CONNECT method CONNECT

# TAG: http_access
```

```
# Allowing or Denying access based on defined access lists
#
# Access to the HTTP port:
# http_access allow|deny [!]aclname ...
#
# NOTE on default values:
#
# If there are no "access" lines present, the default is to deny
# the request.
#
# If none of the "access" lines cause a match, the default is the
# opposite of the last line in the list. If the last line was
# deny, then the default is allow. Conversely, if the last line
# is allow, the default will be deny. For these reasons, it is a
# good idea to have an "deny all" or "allow all" entry at the end
# of your access lists to avoid potential confusion.
#
#Default:
# http_access deny all
#
#Recommended minimum configuration:
#
# Only allow cachemgr access from localhost
http_access allow local
http_access allow local2
http_access allow local3
always_direct allow local

http_access allow manager localhost
http_access deny manager
# Only allow purge requests from localhost
http_access allow purge localhost
http_access deny purge
# Deny requests to unknown ports
http_access deny !Safe_ports
# Deny CONNECT to other than SSL ports
http_access deny CONNECT !SSL_ports
#
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
#
http_access allow localhost
# And finally deny all other access to this proxy
http_access deny all

# TAG: icp_access
# Allowing or Denying access to the ICP port based on defined
# access lists
#
# icp_access allow|deny [!]aclname ...
#
```

```
# See http_access for details
#
#Default:
# icp_access deny all
#
#Allow ICP queries from eveyone
icp_access allow all

# TAG: miss_access
# Use to force your neighbors to use you as a sibling instead of
# a parent. For example:
#
# acl localclients src 172.16.0.0/16
# miss_access allow localclients
# miss_access deny !localclients
#
# This means that only your local clients are allowed to fetch
# MISSES and all other clients can only fetch HITS.
#
# By default, allow all clients who passed the http_access rules
# to fetch MISSES from us.
#
#Default setting:
# miss_access allow all

# TAG: cache_peer_access
# Similar to 'cache_peer_domain' but provides more flexibility by
# using ACL elements.
#
# cache_peer_access cache-host allow|deny [!]aclname ...
#
# The syntax is identical to 'http_access' and the other lists of
# ACL elements. See the comments for 'http_access' below, or
# the Squid FAQ (http://www.squid-cache.org/FAQ/FAQ-10.html).
#
#Default:
# none

# TAG: proxy_auth_realm
# Specifies the realm name which is to be reported to the client for
# proxy authentication (part of the text the user will see when
# prompted their username and password).
#
#Default:
# proxy_auth_realm Squid proxy-caching web server

# TAG: ident_lookup_access
# A list of ACL elements which, if matched, cause an ident
# (RFC 931) lookup to be performed for this request. For
# example, you might choose to always perform ident lookups
```

```
# for your main multi-user Unix boxes, but not for your Macs
# and PCs. By default, ident lookups are not performed for
# any requests.
#
# To enable ident lookups for specific client addresses, you
# can follow this example:
#
# acl ident_aware_hosts src 198.168.1.0/255.255.255.0
# ident_lookup_access allow ident_aware_hosts
# ident_lookup_access deny all
#
# This option may be disabled by using --disable-ident with
# the configure script.
#
#Default:
# ident_lookup_access deny all

# ADMINISTRATIVE PARAMETERS
# -----

# TAG: cache_mgr
# Email-address of local cache manager who will receive
# mail if the cache dies. The default is "webmaster."
#
#Default:
# cache_mgr webmaster

# TAG: cache_effective_user
# TAG: cache_effective_group
#
# If the cache is run as root, it will change its effective/real
# UID/GID to the UID/GID specified below. The default is to
# change to UID to proxy and GID to proxy.
#
# If Squid is not started as root, the default is to keep the
# current UID/GID. Note that if Squid is not started as root then
# you cannot set http_port to a value lower than 1024.
#
#Default:
# cache_effective_user proxy
# cache_effective_group proxy

# TAG: visible_hostname
# If you want to present a special hostname in error messages, etc,
# then define this. Otherwise, the return value of gethostname()
# will be used. If you have multiple caches in a cluster and
# get errors about IP-forwarding you must set them to have individual
# names with this setting.
#
```

```
#Default:
# none
visible_hostname Merak.Alufis

# TAG: unique_hostname
# If you want to have multiple machines with the same
# 'visible_hostname' then you must give each machine a different
# 'unique_hostname' so that forwarding loops can be detected.
#
#Default:
# none

# TAG: hostname_aliases
# A list of other DNS names that your cache has.
#
#Default:
# none

# OPTIONS FOR THE CACHE REGISTRATION SERVICE
# -----
#
# This section contains parameters for the (optional) cache
# announcement service. This service is provided to help
# cache administrators locate one another in order to join or
# create cache hierarchies.
#
# An 'announcement' message is sent (via UDP) to the registration
# service by Squid. By default, the announcement message is NOT
# SENT unless you enable it with 'announce_period' below.
#
# The announcement message includes your hostname, plus the
# following information from this configuration file:
#
# http_port
# icp_port
# cache_mgr
#
# All current information is processed regularly and made
# available on the Web at http://www.ircache.net/Cache/Tracker/.

# TAG: announce_period
# This is how frequently to send cache announcements. The
# default is '0' which disables sending the announcement
# messages.
#
# To enable announcing your cache, just uncomment the line
# below.
#
#Default:
```

```
# announce_period 0
#
#To enable announcing your cache, just uncomment the line below.
#announce_period 1 day

# TAG: announce_host
# TAG: announce_file
# TAG: announce_port
# announce_host and announce_port set the hostname and port
# number where the registration message will be sent.
#
# Hostname will default to 'tracker.ircache.net' and port will
# default default to 3131. If the 'filename' argument is given,
# the contents of that file will be included in the announce
# message.
#
#Default:
# announce_host tracker.ircache.net
# announce_port 3131

# HTTPD-ACCELERATOR OPTIONS
# -----

# TAG: httpd_accel_host
# TAG: httpd_accel_port
# If you want to run Squid as an httpd accelerator, define the
# host name and port number where the real HTTP server is.
#
# If you want virtual host support then specify the hostname
# as "virtual".
#
# If you want virtual port support then specify the port as "0".
#
# NOTE: enabling httpd_accel_host disables proxy-caching and
# ICP. If you want these features enabled also, then set
# the 'httpd_accel_with_proxy' option.
#
#Default:
httpd_accel_host virtual
httpd_accel_port 80

# TAG: httpd_accel_single_host on|off
# If you are running Squid as a accelerator and have a single backend
# server then set this to on. This causes Squid to forward the request
# to this server irregardles of what any redirectors or Host headers
# says.
#
# Leave this at off if you have multiple backend servers, and use a
# redirector (or host table or private DNS) to map the requests to the
```

```
# appropriate backend servers. Note that the mapping needs to be a
# 1-1 mapping between requested and backend (from redirector) domain
# names or caching will fail, as caching is performed using the
# URL returned from the redirector.
#
# See also redirect_rewrites_host_header.
#
#Default:
# httpd_accel_single_host off

# TAG: httpd_accel_with_proxy on|off
# If you want to use Squid as both a local httpd accelerator
# and as a proxy, change this to 'on'. Note however that your
# proxy users may have trouble to reach the accelerated domains
# unless their browsers are configured not to use this proxy for
# those domains (for example via the no_proxy browser configuration
# setting)
#
#Default:
httpd_accel_with_proxy on

# TAG: httpd_accel_uses_host_header on|off
# HTTP/1.1 requests include a Host: header which is basically the
# hostname from the URL. Squid can be an accelerator for
# different HTTP servers by looking at this header. However,
# Squid does NOT check the value of the Host header, so it opens
# a big security hole. We recommend that this option remain
# disabled unless you are sure of what you are doing.
#
# However, you will need to enable this option if you run Squid
# as a transparent proxy. Otherwise, virtual servers which
# require the Host: header will not be properly cached.
#
#Default:
httpd_accel_uses_host_header on

# MISCELLANEOUS
# -----

# TAG: dns_testnames
# The DNS tests exit as soon as the first site is successfully looked up
#
# This test can be disabled with the -D command line option.
#
#Default:
# dns_testnames netscape.com internic.net nlanr.net microsoft.com

# TAG: logfile_rotate
# Specifies the number of logfile rotations to make when you
```

```
# type 'squid -k rotate'. The default is 10, which will rotate
# with extensions 0 through 9. Setting logfile_rotate to 0 will
# disable the rotation, but the logfiles are still closed and
# re-opened. This will enable you to rename the logfiles
# yourself just before sending the rotate signal.
#
# Note, the 'squid -k rotate' command normally sends a USR1
# signal to the running squid process. In certain situations
# (e.g. on Linux with Async I/O), USR1 is used for other
# purposes, so -k rotate uses another signal. It is best to get
# in the habit of using 'squid -k rotate' instead of 'kill -USR1
# <pid>'.
#
# Note2, for Debian/Linux the default of logfile_rotate is
# zero, since it includes external logfile-rotation methods.
#
#Default:
# logfile_rotate 0

# TAG: append_domain
# Appends local domain name to hostnames without any dots in
# them. append_domain must begin with a period.
#
#Example:
# append_domain .yourdomain.com
#
#Default:
# none

# TAG: tcp_recv_bufsize (bytes)
# Size of receive buffer to set for TCP sockets. Probably just
# as easy to change your kernel's default. Set to zero to use
# the default buffer size.
#
#Default:
# tcp_recv_bufsize 0 bytes

# TAG: err_html_text
# HTML text to include in error messages. Make this a "mailto"
# URL to your admin address, or maybe just a link to your
# organizations Web page.
#
# To include this in your error messages, you must rewrite
# the error template files (found in the "errors" directory).
# Wherever you want the 'err_html_text' line to appear,
# insert a %L tag in the error template file.
#
#Default:
# none
```

```
# TAG: deny_info
# Usage: deny_info err_page_name acl
# Example: deny_info ERR_CUSTOM_ACCESS_DENIED bad_guys
#
# This can be used to return a ERR_ page for requests which
# do not pass the 'http_access' rules. A single ACL will cause
# the http_access check to fail. If a 'deny_info' line exists
# for that ACL then Squid returns a corresponding error page.
#
# You may use ERR_ pages that come with Squid or create your own pages
# and put them into the configured errors/ directory.
#
#Default:
# none

# TAG: memory_pools on|off
# If set, Squid will keep pools of allocated (but unused) memory
# available for future use. If memory is a premium on your
# system and you believe your malloc library outperforms Squid
# routines, disable this.
#
#Default:
# memory_pools on

# TAG: memory_pools_limit (bytes)
# Used only with memory_pools on:
# memory_pools_limit 50 MB
#
# If set to a non-zero value, Squid will keep at most the specified
# limit of allocated (but unused) memory in memory pools. All free()
# requests that exceed this limit will be handled by your malloc
# library. Squid does not pre-allocate any memory, just safe-keeps
# objects that otherwise would be free()d. Thus, it is safe to set
# memory_pools_limit to a reasonably high value even if your
# configuration will use less memory.
#
# If not set (default) or set to zero, Squid will keep all memory it
# can. That is, there will be no limit on the total amount of memory
# used for safe-keeping.
#
# To disable memory allocation optimization, do not set
# memory_pools_limit to 0. Set memory_pools to "off" instead.
#
# An overhead for maintaining memory pools is not taken into account
# when the limit is checked. This overhead is close to four bytes per
# object kept. However, pools may actually _save_ memory because of
# reduced memory thrashing in your malloc library.
#
#Default:
# none
```

```
# TAG: forwarded_for on|off
# If set, Squid will include your system's IP address or name
# in the HTTP requests it forwards. By default it looks like
# this:
#
# X-Forwarded-For: 192.1.2.3
#
# If you disable this, it will appear as
#
# X-Forwarded-For: unknown
#
#Default:
# forwarded_for on

# TAG: log_icp_queries on|off
# If set, ICP queries are logged to access.log. You may wish
# to disable this if your ICP load is VERY high to speed things
# up or to simplify log analysis.
#
#Default:
# log_icp_queries on

# TAG: icp_hit_stale on|off
# If you want to return ICP_HIT for stale cache objects, set this
# option to 'on'. If you have sibling relationships with caches
# in other administrative domains, this should be 'off'. If you only
# have sibling relationships with caches under your control, then
# it is probably okay to set this to 'on'.
#
#Default:
# icp_hit_stale off

# TAG: minimum_direct_hops
# If using the ICMP pingging stuff, do direct fetches for sites
# which are no more than this many hops away.
#
#Default:
# minimum_direct_hops 4

# TAG: minimum_direct_rtt
# If using the ICMP pingging stuff, do direct fetches for sites
# which are no more than this many rtt milliseconds away.
#
#Default:
# minimum_direct_rtt 400

# TAG: cachemgr_passwd
# Specify passwords for cachemgr operations.
#
```

```
# Usage: cachemgr_passwd password action action ...
#
# Some valid actions are (see cache manager menu for a full list):
# 5min
# 60min
# asndb
# authenticator
# cbdata
# client_list
# comm_incoming
# config *
# counters
# delay
# digest_stats
# dns
# events
# filedescriptors
# fqdn_cache
# histograms
# http_headers
# info
# io
# ipcache
# mem
# menu
# netdb
# non_peers
# objects
# pconn
# peer_select
# redirector
# refresh
# server_list
# shutdown *
# store_digest
# storedir
# utilization
# via_headers
# vm_objects
#
# * Indicates actions which will not be performed without a
#   valid password, others can be performed if not listed here.
#
# To disable an action, set the password to "disable".
# To allow performing an action without a password, set the
# password to "none".
#
# Use the keyword "all" to set the same password for all actions.
#
#Example:
```

```
# cachemgr_passwd secret shutdown
# cachemgr_passwd lessssssssecret info stats/objects
# cachemgr_passwd disable all
#
#Default:
# none

# TAG: store_avg_object_size (kbytes)
# Average object size, used to estimate number of objects your
# cache can hold. See doc/Release-Notes-1.1.txt. The default is
# 13 KB.
#
#Default:
# store_avg_object_size 13 KB

# TAG: store_objects_per_bucket
# Target number of objects per bucket in the store hash table.
# Lowering this value increases the total number of buckets and
# also the storage maintenance rate. The default is 50.
#
#Default:
# store_objects_per_bucket 20

# TAG: client_db on|off
# If you want to disable collecting per-client statistics, then
# turn off client_db here.
#
#Default:
# client_db on

# TAG: netdb_low
# TAG: netdb_high
# The low and high water marks for the ICMP measurement
# database. These are counts, not percents. The defaults are
# 900 and 1000. When the high water mark is reached, database
# entries will be deleted until the low mark is reached.
#
#Default:
# netdb_low 900
# netdb_high 1000

# TAG: netdb_ping_period
# The minimum period for measuring a site. There will be at
# least this much delay between successive pings to the same
# network. The default is five minutes.
#
#Default:
# netdb_ping_period 5 minutes

# TAG: query_icmp on|off
```

```
# If you want to ask your peers to include ICMP data in their ICP
# replies, enable this option.
#
# If your peer has configured Squid (during compilation) with
# '--enable-icmp' then that peer will send ICMP pings to origin server
# sites of the URLs it receives. If you enable this option then the
# ICP replies from that peer will include the ICMP data (if available).
# Then, when choosing a parent cache, Squid will choose the parent with
# the minimal RTT to the origin server. When this happens, the
# hierarchy field of the access.log will be
# "CLOSEST_PARENT_MISS". This option is off by default.
#
#Default:
# query_icmp off

# TAG: test_reachability on|off
# When this is 'on', ICP MISS replies will be ICP_MISS_NOFETCH
# instead of ICP_MISS if the target host is NOT in the ICP
# database, or has a zero RTT.
#
#Default:
# test_reachability off

# TAG: buffered_logs on|off
# Some log files (cache.log, useragent.log) are written with
# stdio functions, and as such they can be buffered or
# unbuffered. By default they will be unbuffered. Buffering them
# can speed up the writing slightly (though you are unlikely to
# need to worry).
#
#Default:
# buffered_logs off

# TAG: reload_into_ims on|off
# When you enable this option, client no-cache or ''reload''
# requests will be changed to If-Modified-Since requests.
# Doing this VIOLATES the HTTP standard. Enabling this
# feature could make you liable for problems which it
# causes.
#
# see also refresh_pattern for a more selective approach.
#
# This option may be disabled by using --disable-http-violations
# with the configure script.
#
#Default:
# reload_into_ims off

# TAG: always_direct
# Usage: always_direct allow|deny [!]aclname ...
```

```
#
# Here you can use ACL elements to specify requests which should
# ALWAYS be forwarded directly to origin servers. For example,
# to always directly forward requests for local servers use
# something like:
#
# acl local-servers dstdomain my.domain.net
# always_direct allow local-servers
#
# To always forward FTP requests directly, use
#
# acl FTP proto FTP
# always_direct allow FTP
#
# NOTE: There is a similar, but opposite option named
# 'never_direct'. You need to be aware that "always_direct deny
# foo" is NOT the same thing as "never_direct allow foo". You
# may need to use a deny rule to exclude a more-specific case of
# some other rule. Example:
#
# acl local-external dstdomain external.foo.net
# acl local-servers dstdomain foo.net
# always_direct deny local-external
# always_direct allow local-servers
#
# This option replaces some v1.1 options such as local_domain
# and local_ip.
#
#Default:
# none

# TAG: never_direct
# Usage: never_direct allow|deny [!]aclname ...
#
# never_direct is the opposite of always_direct. Please read
# the description for always_direct if you have not already.
#
# With 'never_direct' you can use ACL elements to specify
# requests which should NEVER be forwarded directly to origin
# servers. For example, to force the use of a proxy for all
# requests, except those in your local domain use something like:
#
# acl local-servers dstdomain foo.net
# acl all src 0.0.0.0/0.0.0.0
# never_direct deny local-servers
# never_direct allow all
#
# or if squid is inside a firewall and there is local intranet
# servers inside the firewall then use something like:
#
```

```
# acl local-intranet dstdomain foo.net
# acl local-external dstdomain external.foo.net
# always_direct deny local-external
# always_direct allow local-intranet
# never_direct allow all
#
# This option replaces some v1.1 options such as inside_firewall
# and firewall_ip.
#
#Default:
# none

# TAG: anonymize_headers
# Usage: anonymize_headers allow|deny header_name ...
#
# This option replaces the old 'http_anonymizer' option with
# something that is much more configurable. You may now
# specify exactly which headers are to be allowed, or which
# are to be removed from outgoing requests.
#
# There are two methods of using this option. You may either
# allow specific headers (thus denying all others), or you
# may deny specific headers (thus allowing all others).
#
# For example, to achieve the same behavior as the old
# 'http_anonymizer standard' option, you should use:
#
# anonymize_headers deny From Referer Server
# anonymize_headers deny User-Agent WWW-Authenticate Link
#
# Or, to reproduce the old 'http_anonymizer paranoid' feature
# you should use:
#
# anonymize_headers allow Allow Authorization Cache-Control
# anonymize_headers allow Content-Encoding Content-Length
# anonymize_headers allow Content-Type Date Expires Host
# anonymize_headers allow If-Modified-Since Last-Modified
# anonymize_headers allow Location Pragma Accept
# anonymize_headers allow Accept-Encoding Accept-Language
# anonymize_headers allow Content-Language Mime-Version
# anonymize_headers allow Retry-After Title Connection
# anonymize_headers allow Proxy-Connection
#
# NOTE: You can not mix "allow" and "deny". All 'anonymize_headers'
# lines must have the same second argument.
#
# By default, all headers are allowed (no anonymizing is
# performed).
#
#Default:
```

```
# none

# TAG: fake_user_agent
# If you filter the User-Agent header with 'anonymize_headers' it
# may cause some Web servers to refuse your request. Use this to
# fake one up. For example:
#
# fake_user_agent Nutscape/1.0 (CP/M; 8-bit)
# (credit to Paul Southworth pauls@etext.org for this one!)
#
#Default:
# none

# TAG: icon_directory
# Where the icons are stored. These are normally kept in
# /usr/lib/squid/icons
#
#Default:
# icon_directory /usr/lib/squid/icons

# TAG: error_directory
# If you wish to create your own versions of the default
# (English) error files, either to customize them to suit your
# language or company copy the template English files to another
# directory and point this tag at them.
#
#Default:
# error_directory /usr/lib/squid/errors/English

# TAG: minimum_retry_timeout (seconds)
# This specifies the minimum connect timeout, for when the
# connect timeout is reduced to compensate for the availability
# of multiple IP addresses.
#
# When a connection to a host is initiated, and that host has
# several IP addresses, the default connection timeout is reduced
# by dividing it by the number of addresses. So, a site with 15
# addresses would then have a timeout of 8 seconds for each
# address attempted. To avoid having the timeout reduced to the
# point where even a working host would not have a chance to
# respond, this setting is provided. The default, and the
# minimum value, is five seconds, and the maximum value is sixty
# seconds, or half of connect_timeout, whichever is greater and
# less than connect_timeout.
#
#Default:
# minimum_retry_timeout 5 seconds

# TAG: maximum_single_addr_tries
# This sets the maximum number of connection attempts for a
```

```
# host that only has one address (for multiple-address hosts,
# each address is tried once).
#
# The default value is three tries, the (not recommended)
# maximum is 255 tries. A warning message will be generated
# if it is set to a value greater than ten.
#
#Default:
# maximum_single_addr_tries 3

# TAG: snmp_port
# Squid can now serve statistics and status information via SNMP.
# By default it listens to port 3401 on the machine. If you don't
# wish to use SNMP, set this to "0".
#
# Note: on Debian/Linux, the default is zero - you need to
# set it to 3401 to enable it.
#
# NOTE: SNMP support requires use the --enable-snmp configure
# command line option.
#
#Default:
# snmp_port 0

# TAG: snmp_access
# Allowing or denying access to the SNMP port.
#
# All access to the agent is denied by default.
# usage:
#
# snmp_access allow|deny [!]aclname ...
#
#Example:
# snmp_access allow snmppublic localhost
# snmp_access deny all
#
#Default:
# snmp_access deny all

# TAG: snmp_incoming_address
# TAG: snmp_outgoing_address
# Just like 'udp_incoming_address' above, but for the SNMP port.
#
# snmp_incoming_address is used for the SNMP socket receiving
# messages from SNMP agents.
# snmp_outgoing_address is used for SNMP packets returned to SNMP
# agents.
#
# The default snmp_incoming_address (0.0.0.0) is to listen on all
# available network interfaces.
```

```
#
# If snmp_outgoing_address is set to 255.255.255.255 (the default)
# then it will use the same socket as snmp_incoming_address. Only
# change this if you want to have SNMP replies sent using another
# address than where this Squid listens for SNMP queries.
#
# NOTE, snmp_incoming_address and snmp_outgoing_address can not have
# the same value since they both use port 3401.
#
#Default:
# snmp_incoming_address 0.0.0.0
# snmp_outgoing_address 255.255.255.255

# TAG: as_whois_server
# WHOIS server to query for AS numbers. NOTE: AS numbers are
# queried only when Squid starts up, not for every request.
#
#Default:
# as_whois_server whois.ra.net
# as_whois_server whois.ra.net

# TAG: wccp_router
# Use this option to define your WCCP ''home'' router for
# Squid. Setting the 'wccp_router' to 0.0.0.0 (the default)
# disables WCCP.
#
#Default:
# wccp_router 0.0.0.0

# TAG: wccp_version
# According to some users, Cisco IOS 11.2 only supports WCCP
# version 3. If you're using that version of IOS, change
# this value to 3.
#
#Default:
# wccp_version 4

# TAG: wccp_incoming_address
# TAG: wccp_outgoing_address
# wccp_incoming_address Use this option if you require WCCP
# messages to be received on only one
# interface. Do NOT use this option if
# you're unsure how many interfaces you
# have, or if you know you have only one
# interface.
#
# wccp_outgoing_address Use this option if you require WCCP
# messages to be sent out on only one
# interface. Do NOT use this option if
# you're unsure how many interfaces you
```

```
# have, or if you know you have only one
# interface.
#
#       The default behavior is to not bind to any specific address.
#
#       NOTE, wccp_incoming_address and wccp_outgoing_address can not have
#       the same value since they both use port 2048.
#
#Default:
# wccp_incoming_address 0.0.0.0
# wccp_outgoing_address 255.255.255.255

# DELAY POOL PARAMETERS (all require DELAY_POOLS compilation option)
# -----

# TAG: delay_pools
# This represents the number of delay pools to be used. For example,
# if you have one class 2 delay pool and one class 3 delays pool, you
# have a total of 2 delay pools.
#
# To enable this option, you must use --enable-delay-pools with the
# configure script.
#
#Default:
# delay_pools 0

# TAG: delay_class
# This defines the class of each delay pool. There must be exactly one
# delay_class line for each delay pool. For example, to define two
# delay pools, one of class 2 and one of class 3, the settings above
# and here would be:
#
#Example:
# delay_pools 2      # 2 delay pools
# delay_class 1 2    # pool 1 is a class 2 pool
# delay_class 2 3    # pool 2 is a class 3 pool
#
# The delay pool classes are:
#
# class 1 Everything is limited by a single aggregate
# bucket.
#
# class 2 Everything is limited by a single aggregate
# bucket as well as an "individual" bucket chosen
# from bits 25 through 32 of the IP address.
#
# class 3 Everything is limited by a single aggregate
# bucket as well as a "network" bucket chosen
# from bits 17 through 24 of the IP address and a
```

```
# "individual" bucket chosen from bits 17 through
# 32 of the IP address.
#
# NOTE: If an IP address is a.b.c.d
# -> bits 25 through 32 are "d"
# -> bits 17 through 24 are "c"
# -> bits 17 through 32 are "c * 256 + d"
#
#Default:
# none

# TAG: delay_access
# This is used to determine which delay pool a request falls into.
# The first matched delay pool is always used, i.e., if a request falls
# into delay pool number one, no more delay are checked, otherwise the
# rest are checked in order of their delay pool number until they have
# all been checked. For example, if you want some_big_clients in delay
# pool 1 and lotsa_little_clients in delay pool 2:
#
#Example:
# delay_access 1 allow some_big_clients
# delay_access 1 deny all
# delay_access 2 allow lotsa_little_clients
# delay_access 2 deny all
#
#Default:
# none

# TAG: delay_parameters
# This defines the parameters for a delay pool. Each delay pool has
# a number of "buckets" associated with it, as explained in the
# description of delay_class. For a class 1 delay pool, the syntax is:
#
#delay_parameters pool aggregate
#
# For a class 2 delay pool:
#
#delay_parameters pool aggregate individual
#
# For a class 3 delay pool:
#
#delay_parameters pool aggregate network individual
#
# The variables here are:
#
# pool a pool number - ie, a number between 1 and the
# number specified in delay_pools as used in
# delay_class lines.
#
# aggregate the "delay parameters" for the aggregate bucket
```

```
# (class 1, 2, 3).
#
# individual the "delay parameters" for the individual
# buckets (class 2, 3).
#
# network the "delay parameters" for the network buckets
# (class 3).
#
# A pair of delay parameters is written restore/maximum, where restore is
# the number of bytes (not bits - modem and network speeds are usually
# quoted in bits) per second placed into the bucket, and maximum is the
# maximum number of bytes which can be in the bucket at any time.
#
# For example, if delay pool number 1 is a class 2 delay pool as in the
# above example, and is being used to strictly limit each host to 64kbps
# (plus overheads), with no overall limit, the line is:
#
#delay_parameters 1 -1/-1 8000/8000
#
# Note that the figure -1 is used to represent "unlimited".
#
# And, if delay pool number 2 is a class 3 delay pool as in the above
# example, and you want to limit it to a total of 256kbps (strict limit)
# with each 8-bit network permitted 64kbps (strict limit) and each
# individual host permitted 4800bps with a bucket maximum size of 64kb
# to permit a decent web page to be downloaded at a decent speed
# (if the network is not being limited due to overuse) but slow down
# large downloads more significantly:
#
#delay_parameters 2 32000/32000 8000/8000 600/64000
#
# There must be one delay_parameters line for each delay pool.
#
#Default:
# none

# TAG: delay_initial_bucket_level (percent, 0-100)
# The initial bucket percentage is used to determine how much is put
# in each bucket when squid starts, is reconfigured, or first notices
# a host accessing it (in class 2 and class 3, individual hosts and
# networks only have buckets associated with them once they have been
# "seen" by squid).
#
#Default:
# delay_initial_bucket_level 50

# TAG: incoming_icp_average
# TAG: incoming_http_average
# TAG: incoming_dns_average
# TAG: min_icp_poll_cnt
```

```
# TAG: min_dns_poll_cnt
# TAG: min_http_poll_cnt
# Heavy voodoo here. I can't even believe you are reading this.
# Are you crazy? Don't even think about adjusting these unless
# you understand the algorithms in comm_select.c first!
#
#Default:
# incoming_icp_average 6
# incoming_http_average 4
# incoming_dns_average 4
# min_icp_poll_cnt 8
# min_dns_poll_cnt 8
# min_http_poll_cnt 8

# TAG: max_open_disk_fds
# To avoid having disk as the I/O bottleneck Squid can optionally
# bypass the on-disk cache if more than this amount of disk file
# descriptors are open.
#
# A value of 0 indicates no limit.
#
#Default:
# max_open_disk_fds 0

# TAG: offline_mode
# Enable this option and Squid will never try to validate cached
# objects.
#
#Default:
# offline_mode off

# TAG: uri_whitespace
# What to do with requests that have whitespace characters in the
# URI. Options:
#
# strip: The whitespace characters are stripped out of the URL.
# This is the behavior recommended by RFC2616.
# deny: The request is denied. The user receives an "Invalid
# Request" message.
# allow: The request is allowed and the URI is not changed. The
# whitespace characters remain in the URI. Note the
# whitespace is passed to redirector processes if they
# are in use.
# encode: The request is allowed and the whitespace characters are
# encoded according to RFC1738. This could be considered
# a violation of the HTTP/1.1
# RFC because proxies are not allowed to rewrite URI's.
# chop: The request is allowed and the URI is chopped at the
# first whitespace. This might also be considered a
# violation.
```

```
#
#Default:
# uri_whitespace strip

# TAG: broken_posts
# A list of ACL elements which, if matched, causes Squid to send
# a extra CRLF pair after the body of a PUT/POST request.
#
# Some HTTP servers has broken implementations of PUT/POST,
# and rely on a extra CRLF pair sent by some WWW clients.
#
# Quote from RFC 2068 section 4.1 on this matter:
#
# Note: certain buggy HTTP/1.0 client implementations generate an
# extra CRLF's after a POST request. To restate what is explicitly
# forbidden by the BNF, an HTTP/1.1 client must not preface or follow
# a request with an extra CRLF.
#
#Example:
# acl buggy_server url_regex ^http://....
# broken_posts allow buggy_server
#
#Default:
# none

# TAG: mcast_miss_addr
# Note: This option is only available if Squid is rebuilt with the
# -DMULTICAST_MISS_STREAM option
#
# If you enable this option, every "cache miss" URL will
# be sent out on the specified multicast address.
#
# Do not enable this option unless you are are absolutely
# certain you understand what you are doing.
#
#Default:
# mcast_miss_addr 255.255.255.255

# TAG: mcast_miss_ttl
# Note: This option is only available if Squid is rebuilt with the
# -DMULTICAST_MISS_TTL option
#
# This is the time-to-live value for packets multicasted
# when multicasting off cache miss URLs is enabled. By
# default this is set to 'site scope', i.e. 16.
#
#Default:
# mcast_miss_ttl 16

# TAG: mcast_miss_port
```

```
# Note: This option is only available if Squid is rebuilt with the
#       -DMULTICAST_MISS_STREAM option
#
# This is the port number to be used in conjunction with
# 'mcast_miss_addr'.
#
#Default:
# mcast_miss_port 3135

# TAG: mcast_miss_encode_key
# Note: This option is only available if Squid is rebuilt with the
#       -DMULTICAST_MISS_STREAM option
#
# The URLs that are sent in the multicast miss stream are
# encrypted. This is the encryption key.
#
#Default:
# mcast_miss_encode_key XXXXXXXXXXXXXXXXXXXX

# TAG: nonhierarchical_direct
# By default, Squid will send any non-hierarchical requests
# (matching hierarchy_stoplist or not cachable request type) direct
# to origin servers.
#
# If you set this to off, then Squid will prefer to send these
# requests to parents.
#
# Note that in most configurations, by turning this off you will only
# add latency to these request without any improvement in global hit
# ratio.
#
# If you are inside an firewall then see never_direct instead of
# this directive.
#
#Default:
# nonhierarchical_direct on

# TAG: prefer_direct
# Normally Squid tries to use parents for most requests. If you by some
# reason like it to first try going direct and only use a parent if
# going direct fails then set this to off.
#
# By combining nonhierarchical_direct off and prefer_direct on you
# can set up Squid to use a parent as a backup path if going direct
# fails.
#
#Default:
# prefer_direct off

# TAG: strip_query_terms
```

```
# By default, Squid strips query terms from requested URLs before
# logging. This protects your user's privacy.
#
#Default:
# strip_query_terms on

# TAG: coredump_dir
# By default Squid leaves core files in the first cache_dir
# directory. If you set 'coredump_dir' to a directory
# that exists, Squid will chdir() to that directory at startup
# and coredump files will be left there.
#
#Default:
# none

# TAG: redirector_bypass
# When this is 'on', a request will not go through the
# redirector if all redirectors are busy. If this is 'off'
# and the redirector queue grows too large, Squid will exit
# with a FATAL error and ask you to increase the number of
# redirectors. You should only enable this if the redirectors
# are not critical to your caching system. If you use
# redirectors for access control, and you enable this option,
# then users may have access to pages that they should not
# be allowed to request.
#
#Default:
# redirector_bypass off

# TAG: ignore_unknown_nameservers
# By default Squid checks that DNS responses are received
# from the same IP addresses that they are sent to. If they
# don't match, Squid ignores the response and writes a warning
# message to cache.log. You can allow responses from unknown
# nameservers by setting this option to 'off'.
#
#Default:
# ignore_unknown_nameservers on

# TAG: digest_generation
# This controls whether the server will generate a Cache Digest
# of its contents. By default, Cache Digest generation is
# enabled if Squid is compiled with USE_CACHE_DIGESTS defined.
#
#Default:
# digest_generation on

# TAG: digest_bits_per_entry
# This is the number of bits of the server's Cache Digest which
# will be associated with the Digest entry for a given HTTP
```

```
# Method and URL (public key) combination. The default is 5.
#
#Default:
# digest_bits_per_entry 5

# TAG: digest_rebuild_period (seconds)
# This is the number of seconds between Cache Digest rebuilds.
#
#Default:
# digest_rebuild_period 1 hour

# TAG: digest_rewrite_period (seconds)
# This is the number of seconds between Cache Digest writes to
# disk.
#
#Default:
# digest_rewrite_period 1 hour

# TAG: digest_swapout_chunk_size (bytes)
# This is the number of bytes of the Cache Digest to write to
# disk at a time. It defaults to 4096 bytes (4KB), the Squid
# default swap page.
#
#Default:
# digest_swapout_chunk_size 4096 bytes

# TAG: digest_rebuild_chunk_percentage (percent, 0-100)
# This is the percentage of the Cache Digest to be scanned at a
# time. By default it is set to 10% of the Cache Digest.
#
#Default:
# digest_rebuild_chunk_percentage 10

# TAG: chroot
# Use this to have Squid do a chroot() while initializing. This
# also causes Squid to fully drop root privileges after
# initializing. This means, for example, that if you use a HTTP
# port less than 1024 and try to reconfigure, you will get an
# error.
#
#Default:
# none

# TAG: client_persistent_connections
# TAG: server_persistent_connections
# Persistent connection support for clients and servers. By
# default, Squid uses persistent connections (when allowed)
# with its clients and servers. You can use these options to
# disable persistent connections with clients and/or servers.
#
```

```
#Default:
# client_persistent_connections on
# server_persistent_connections on

# TAG: pipeline_prefetch
# To boost the performance of pipelined requests to closer
# match that of a non-proxied environment Squid tries to fetch
# up to two requests in parallel from a pipeline.
#
#Default:
# pipeline_prefetch on

# TAG: extension_methods
# Squid only knows about standardized HTTP request methods.
# You can add up to 20 additional "extension" methods here.
#
#Default:
# none

# TAG: high_response_time_warning (msec)
# If the one-minute median response time exceeds this value,
# Squid prints a WARNING with debug level 0 to get the
# administrators attention. The value is in milliseconds.
#
#Default:
# high_response_time_warning 0

# TAG: high_page_fault_warning
# If the one-minute average page fault rate exceeds this
# value, Squid prints a WARNING with debug level 0 to get
# the administrators attention. The value is in page faults
# per second.
#
#Default:
# high_page_fault_warning 0

# TAG: high_memory_warning
# If the memory usage (as determined by mallinfo) exceeds
# value, Squid prints a WARNING with debug level 0 to get
# the administrators attention.
#
#Default:
# high_memory_warning 0

# TAG: store_dir_select_algorithm
# Set this to 'round-robin' as an alternative.
#
#Default:
# store_dir_select_algorithm least-load
```

```

# TAG: forward_log
# Note: This option is only available if Squid is rebuilt with the
#       -DWIP_FWD_LOG option
#
# Logs the server-side requests.
#
# This is currently work in progress.
#
#Default:
# none

# TAG: ie_refresh on|off
# Microsoft Internet Explorer up until version 5.5 Service
# Pack 1 has an issue with transparent proxies, wherein it
# is impossible to force a refresh. Turning this on provides
# a partial fix to the problem, by causing all IMS-REFRESH
# requests from older IE versions to check the origin server
# for fresh content. This reduces hit ratio by some amount
# (~10% in my experience), but allows users to actually get
# fresh content when they want it. Note that because Squid
# cannot tell if the user is using 5.5 or 5.5SP1, the behavior
# of 5.5 is unchanged from old versions of Squid (i.e. a
# forced refresh is impossible). Newer versions of IE will,
# hopefully, continue to have the new behavior and will be
# handled based on that assumption. This option defaults to
# the old Squid behavior, which is better for hit ratios but
# worse for clients using IE, if they need to be able to
# force fresh content.
#
#Default:
# ie_refresh off

```

8.2. PPTPD (para las VPN entrantes)

8.2.1. /etc/ppp/pptpd-options:

```

## SAMPLE ONLY
## CHANGE TO SUIT YOUR SYSTEM

## turn pptpd syslog debugging on
debug

## change 'servername' to whatever you specify as your server name in chap-secrets
name Merak

## change the domainname to your local domain
#domain mydomain.net

```

```
## these are reasonable defaults for WinXXXX clients
## for the security related settings
auth
+chap

#require-chap
#require-chapms
#require-chapms-v2
#+chap

##### ATTENTION #####
# These options are disabled because the stock Debian kernel as well as the
# pppd package do not support MPPE encryption. But it is recommended to patch
# your kernel and use a pppd with MPPE support if you use this package. Without
# these options, PPTP can not be considered to be safe.
##+chapms
##+chapms-v2
##mppe-40
##mppe-128
##mppe-stateless

## Fill in your addresses
ms-dns 192.168.0.1
# ms-wins 192.168.0.1

## Fill in your netmask
netmask 255.255.0.0

## some defaults
#ndefaultroute
lock
```

8.2.2. /etc/ppp/chap-secrets:

```
# Secrets for authentication using CHAP
# client server secret IP addresses
prueba * probando *
```

Capítulo 9

9 Créditos

Copyright

Copyright (c) 2003 Pablo Iranzo Gómez

Se le otorga permiso para copiar, distribuir y/o modificar este documento bajo los términos de la GNU General Public License Versión 2 o superior publicada por la Free Software Foundation.

Créditos

Este documento ha sido creado utilizando el editor L^AT_EX y ha sido compilado con L^AT_EX₂_ε bajo Debian GNU/Linux y luego convertido al formato que está viendo actualmente.

Por favor, si utiliza este manual, mándeme un email para saber hasta dónde llega y si es utilizado.