Escola Universitària
Politècnica de Mataró

Ingeniería Técnica de Telecomunicación: Especialidad Telemática

Diseño e implementación de un *HotSpot-in-a-Box*

Pau Oliva Fora

OTOÑO 2003

Escola Universitària
Politècnica de Mataró

Ingeniería Técnica de Telecomunicación: Especialidad Telemática

Diseño e implementación de un *HotSpot-in-a-Box*

Pau Oliva Fora
Léonard Janer García

OTOÑO 2003

Escola Universitària
Politècnica de Mataró

Ingeniería Técnica de Telecomunicación: Especialidad Telemática

TRABAJO FIN DE CARRERA

TÍTULO: Diseño e implementación de un *HotSpot-in-a-Box*

AUTOR: Pau Oliva Fora
PROFESOR PONENTE: Léonard Janer García

CALIFICACIÓN

Defendido el trabajo en la convocatoria del día: de de 2004

Ha obtenido la calificación de

DIRECTOR EUPMT	**PROFESOR Delegado UPC**	**PROFESOR PONENTE**
.....	.....	.....

SECRETARIO

.....

Diseño e implementación de un *HotSpot-in-a-Box* – Pau Oliva

OTOÑO
2003

Este documento ha sido realizado con software libre, utilizando el editor *vim* (*vi improved*), el lenguaje de macros LaTeX, y el programa de manipulación de imágenes *GIMP*.

Dedicatoria

A mis padres, Ricard y Elena, por brindarme su apoyo y comprensión durante toda mi vida, por soportarme y aguantarme, por guiarme, por estar ahí cuando más falta me han hecho, por quererme y apoyarme siempre, por darme todo lo que he necesitado, por creer en mí, por sus consejos, por ayudarme, por ofrecerme su confianza, su amistad y su amor. Sin ellos todo mi esfuerzo hubiera sido en vano.

Agradecimientos

La realización de este proyecto no hubiera sido posible sin la existencia del software libre, por ello quiero dar las gracias a todos aquellos que de una u otra forma han dedicado parte de su tiempo colaborando o realizando proyectos de software libre.

En primer lugar quiero dar las gracias a Léonard Janer por la transmisión de sus conocimientos y por dedicarme su tiempo y su experiencia. Gracias por los comentarios y los ánimos que de él he recibido.

Merecen mi agradecimiento Alicia Ollé y Anindya Gosh, que han contribuido con su punto de vista como profesionales y me han hecho ver de forma mucho más clara cuales son las necesidades tecnológicas del sector.

A Esteve Espuña por compartir conmigo sus grandes ideas, por ofrecerme su visión particular de las cosas y por hacer de conejillo de indias en las numerosas pruebas que hemos realizado juntos.

También me gustaría agradecer a Vicente Araña las incontables horas de charla que hemos tenido siempre a altas horas de la madrugada.

Por último, y no por ello menos importante, gracias a mi compañera Laia Albaladejo por su cariño y su apoyo en la distancia que me ha permitido dedicar las horas necesarias para concluir mis estudios con este proyecto.

Resumen

Castellano

En este documento se recogen los elementos clave para realizar un *HotSpot-in-a-Box* basado en la integración de diferentes herramientas Open Source. También se detallan las características técnicas disponibles en la aplicación final que se han alcanzado combinando estas piezas, como la posibilidad de utilizar cualquiera de los tres estándares más comunes de la IEEE, 802.11 a/b/g, el soporte de distintos métodos de autenticación, encriptación, roaming, etc.

Se comentan las aplicaciones que se le puede dar al *HotSpot-in-a-Box* tanto en entornos corporativos como desde el punto de vista de usuario final y se pone especial énfasis en el hecho de haber realizado la totalidad del *HotSpot-in-a-Box* utilizando herramientas libres.

Para concluir se presenta un estudio del modelo de negocio con el correspondiente análisis económico de implantación.

Català

En aquest document es recullen els elements clau per a la realització d'un *HotSpot-in-a-Box* basat amb la integració de diferents eines Open Source. També es detallen les característiques tècniques disponibles a l'aplicació final que s'han aconseguit combinant aquestes peces, com la possibilitat d'utilitzar qualsevol dels tres estàndards mes comuns de la IEEE, 802.11 a/b/g, el suport de diferents mètodes d'autenticació, encriptació, roaming, etc.

Es comenten les aplicacions que se li poden donar al *HotSpot-in-a-Box*, tant en entorns corporatius com des del punt de vista de l'usuari final, i es posa especial èmfasi en el fet d'haver realitzat la totalitat del *HotSpot-in-a-Box* fent servir eines lliures.

Per concloure es presenta un estudi del model de negoci amb el corresponent anàlisi econòmic d'implantació.

English

This document shows the key elements for developing a *HotSpot-in-a-Box* using a solution based on Open Source tools integration. The end application is achieved by combining those elements, the procedure that leads to the final product is also covered in depth, as the possibility of using any of the three most common IEEE 802.11 a/b/g standards, support for several authentication methods, encryption, roaming, etc.

This document also states the applications that can be given to the *HotSpot-in-a-Box* in a corporate environment or under the point of view of the end user, putting special enfasis in the fact that the whole *HotSpot-in-a-Box* has been done using free software.

To finish, it is presented a study of the business model with the corresponding implementation economic analysis.

ÍNDICE GENERAL

Acta I

Dedicatoria VII

Agradecimientos IX

Resumen XI

Índice general XIII

Índice de figuras XVII

1. Introducción 1

2. Objetivos 3

3. Hardware 5

3.1. Hardware elegido para la implementación del proyecto 5

3.1.1. Nimble™ V5 5

3.1.2. US Robotics™ 2410 8
3.1.3. Dlink™ DWL-650G 9

4. Software 11
4.1. Descripción del software utilizado para la implementación 12
4.1.1. Linux 12
4.1.2. Wireless tools y Wireless Extensions 12
4.1.3. PCMCIA Card Services 12
4.1.4. HostAP 13
4.1.5. Multiband Atheros Driver for WiFi (MADWIFI) 13
4.1.6. Bridge-utils 14
4.1.7. Netfilter e iptables 14
4.1.8. Iproute2 y tc 15
4.1.9. ISC DHCP 17
4.1.10. ISC BIND 17
4.1.11. FreeRADIUS 18
4.1.12. Apache 19
4.1.13. NoCat 19
4.1.14. OpenSSL 20
4.1.15. OpenVPN y VTun 21

5. Características Funcionales 23
5.1. Estándares 802.11 soportados 23
5.1.1. IEEE 802.11b 24
5.1.2. IEEE 802.11a 24
5.1.3. IEEE 802.11g 24
5.2. Modos de funcionamiento 25
5.2.1. AP/Router 25
5.2.2. AP/Bridge 26
5.2.3. AP/Repeater 26
5.3. Métodos de autenticación implementados 26
5.3.1. Standard Web Access Method (Universal Browser Login) 26
5.3.2. MAC Authentication (MAC address ACL) 28

5.3.3. IEEE 802.1X Athentication 29
5.4. Métodos de Encriptación implementados 32
5.4.1. Broadcast and unicast Dynamic WEP Rekeying 32
5.5. Inter-Provider Roaming 33
5.5.1. RADIUS Accounting 34
5.5.2. RADIUS Multiple REALM support 35
5.6. Otras Funciones del *HotSpot-in-a-Box* 36
5.6.1. Wireless card autodetection and automatic AP setup 36
5.6.2. Layer 2 user isolation 37
5.6.3. 802.11f Inter-Access Point Protocol (IAPP) 37
5.6.4. White Pages support (Wallet Garden) 38

6. Aplicaciones 39
6.1. Plug-n-share 39

7. Estudio del modelo de negocio del *HotSpot-in-a-Box* 41
7.1. El fenómeno Open Source 42
7.2. Integración de software 43
7.3. Posibilidades de éxito del *HotSpot-in-a-Box* 43
7.4. Ventajas competitivas 44
7.5. Ventajas para el desarrollador 44
7.6. Oportunidad de negocio 45
7.7. Análisis económico 46

Conclusiones 49

A. Análisis financiero del software libre 51

B. GNU Free Documentation License 69

Bibliografía 79

Glosario 80

ÍNDICE DE FIGURAS

3.1. Nimble™ V5 6
3.2. Nimble V5 es mucho más pequeño que un PC de sobremesa, aproximadamente del tamaño de un libro. 7
3.3. US Robotics™ 2410 8
3.4. D-Link™ DWL-650G 9

5.1. Login Page 27
5.2. Ventana de sesión 28
5.3. Sesión completa de autenticación 802.1X mostrando los mensajes EAP y RADIUS 31
5.4. Modelo genérico de roaming 34

CAPÍTULO

1

Introducción

Con la llegada de la tecnología Wi-Fi se creó un nuevo concepto de proveedor de servicios de Internet, los WISP (*Wireless Internet Service Provider*) son empresas dedicadas a ofrecer conectividad inalámbrica a sus clientes que operan instalando hotspots en lugares como hoteles, restaurantes, cafeterías, parques, bibliotecas y aeropuertos, donde gente equipada con portátiles o PDA's capaces de conectar a redes Wi-Fi pasa el tiempo conectada a Internet sin hilos.

Estos WISP suelen ofrecer conectividad por un tiempo limitado a bajo coste, ofreciendo tarifas por volumen de datos transmitido o por tiempo de conexión.

El perfil típico de los clientes de un WISP suele ser un hombre de negocios, que viaja frecuentemente y necesita acceder a su correo electrónico o descargar y enviar documentos para realizar su trabajo. Podríamos comparar a un WISP con una operadora de telefonía móvil, aunque con una cobertura mucho más limitada. Es por esto que los WISP empezaron su carrera por obtener el mayor número de hotspots posibles, haciendo así más atractiva la oferta para el cliente, pudiendo éste conectar a Internet desde un mayor número de lugares.

Los WISP, al igual que las operadoras de telefonía móvil, firman acuerdos de roaming entre

si, que les permiten ofrecer el servicio a sus suscriptores desde un hotspot de la competencia, ampliando así también su región de cobertura.

En un hotspot común un WISP suele instalar uno o varios puntos de acceso, un *gateway* o *access point controller* y una línea de acceso a Internet mediante tecnología xDSL o cable.

El *gateway* o *access point controller* se encarga de impedir el acceso a Internet a los clientes conectados que no hayan sido previamente autenticados, para ello se les suele mostrar una página web donde deben introducir su nombre de usuario y contraseña, o se les ofrece una pasarela de pago para comprar un bono y empezar a utilizar el servicio.

CAPÍTULO

2

Objetivos

Existen en el mercado diversas soluciones que realizan la función de *gateway*, pero desde mi punto de vista las soluciones más económicas suelen ofrecer un número de características muy limitado y no proporcionan la calidad y rendimiento deseados; las soluciones que implementan un mayor número de características y ofrecen mejores resultados tienen precios desorbitados y licencias inaceptables.

De ahí surgió la idea de implementar un *gateway* o *access point controller* utilizando únicamente software open source, que pueda ofrecer un nivel de calidad similar al proporcionado por un *gateway* comercial. Por lo tanto el objetivo principal de este trabajo es demostrar que es posible realizar una implementación de bajo coste simplemente integrando componentes o elementos de software Open Source y ofrecer un resultado técnicamente comparable a una solución comercial.

Los drivers para Linux *HostAP* y *MADWIFI* permiten utilizar la tarjeta Wi-Fi en modo "Master" (funcionando como si fuera un punto de acceso). Gracias a esto es posible convertir un *host* en un AP (modo Host AP), haciendo posible también que el *gateway* o *access point*

controller pueda desempeñar al mismo tiempo la función de punto de acceso, reuniendo todo el hardware necesario para montar un hotspot en un sólo equipo, de aquí surgió el nombre "*HotSpot-in-a-Box*".

Como consecuencia de esto, a parte de cumplir las funciones de *gateway* el *HotSpot-in-a-Box* debe cumplir también la función de punto de acceso, ampliando así los objetivos del proyecto para poder implementar todas las características posibles que ofrece un punto de acceso comercial, utilizando tan sólo una tarjeta Wi-Fi.

CAPÍTULO

3

Hardware

En este capítulo se describe el hardware sobre el cual se ha implementado el proyecto. Esto no significa que sólo pueda usarse el hardware descrito a continuación, sino que éste hardware —al igual que el de muchos otros fabricantes— cumple con las especificaciones necesarias para que el *HotSpot-in-a-Box* funcione.

3.1. Hardware elegido para la implementación del proyecto

3.1.1. Nimble™ V5

Antes de elegir Nimble™ V5 se han evaluado las diferentes posibilidades existentes, desde un simple PC de sobremesa con un adaptador *Cardbus* hasta dispositivos especializados para *embedded computing* basados en arquitectura x86 como los fabricados por Soekris Engineering™ [1].

[1] http://www.soekris.com

Los requisitos mínimos de hardware para poder utilizar sin problemas el software sobre el que se implementará el *HotSpot-in-a-Box* son los siguientes:

- Procesador con arquitectura x86, preferiblemente i586 o superior
- 64Mb de memoria RAM
- Bus de datos Cardbus, PCI o mini-PCI
- Dispositivo de almacenamiento no volátil (Disco duro, CompactFlash...)
- Tarjeta de red

Las principales razones por las que se ha elegido Nimble™ V5 es por que cubre sobradamente estos requisitos: Dispone de un procesador i686 (VIA Nehemiah C3-2 a 733 Mhz), 256Mb de RAM, 30Gb de disco físico y un adaptador Cardbus. Estas prestaciones son equiparables a las de cualquier equipo de sobremesa moderno pero con el tamaño de un libro y un diseño realmente atractivo como se puede ver en la figura 3.1.

Figura 3.1: *Nimble™ V5*

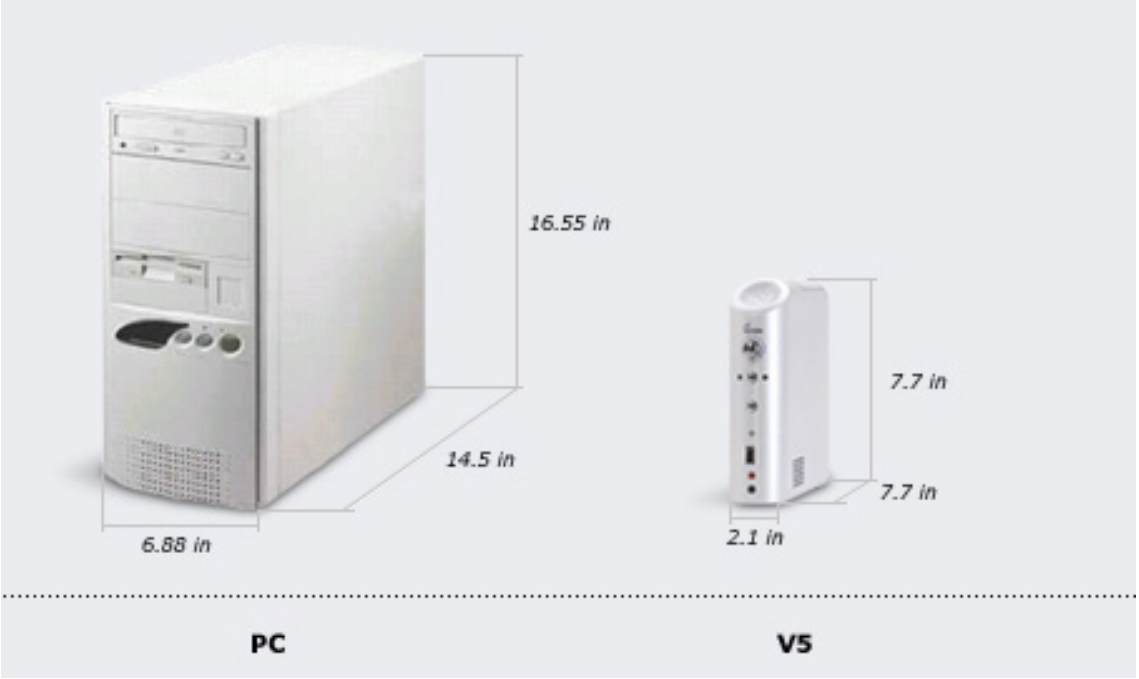


Figura 3.2: *Nimble V5 es mucho más pequeño que un PC de sobremesa, aproximadamente del tamaño de un libro.*

3.1.2. US Robotics™ 2410

Se ha elegido esta tarjeta PCMCIA por que lleva el *chipset* PRISM™ II y es una de las más económicas del mercado.

Figura 3.3: *US Robotics™ 2410*

El driver utilizado para la implementar la mayor parte de las funciones 802.11b del *HotSpot-in-a-Box* funciona sin problemas con cualquier tarjeta que disponga del chipset PRISM™ versión 2, 2.5 y 3.

El chipset PRISM™ II ha sido adoptado por muchos fabricantes de hardware para sus productos Wi-Fi. Este chipset, es totalmente compatible con 802.11 e incluye un módem de Secuencia Directa a 2,4 GHz, con todas las características comunes en los chipsets usados en WLAN (Roaming, WEP, IBSS Ad-Hoc...). El controlador de la capa de acceso al medio implementado por hardware se encarga de la mayor parte de las funcionalidades de 802.11 (en lugar de dejarlas para el driver), lo que simplifica el desarrollo de drivers y ayuda a aumentar el rendimiento.

Para obtener más información sobre este chipset el lector puede consultar la página web de Jean Tourrilhes [1].

3.1.3. Dlink™ DWL-650G

Este adaptador Cardbus es uno de los tantos disponibles en el mercado con el chipset Atheros™ AR5002 WLAN. El driver utilizado para implementar la mayor parte de las funciones 802.11a y 802.11g del *HotSpot-in-a-Box* funciona con cualquier tarjeta que disponga de este chipset, en cualquiera de sus variantes:

- AR5210 – IEEE 802.11a
- AR5211 – IEEE 802.11b y 802.11g
- AR5212 – IEEE 802.11a, 802.11b y 802.11g

Figura 3.4: *D-Link™ DWL-650G*

La familia de chipsets Atheros™ implementan los estándares IEEE 802.11a/b/g siendo capaces de trabajar a frecuencias tanto de 2,4 como de 5 Ghz, obteniendo velocidades de transmisión de hasta 54 Mbps.

Para más información se recomienda visitar la página web del fabricante [2] donde se pueden encontrar las especificaciones de todos los modelos.

[2] http://www.atheros.com

CAPÍTULO

4

Software

En este capítulo se describen los distintos elementos de software, que configurados de la forma adecuada componen la base para la implementación del *HotSpot-in-a-Box*, haciendo una breve descripción de cada pieza del puzle, detallando por que se ha elegido y que aporta al conjunto del proyecto.

Casi todo el software utilizado está disponible bajo licencias Open Source [1]. Open Source no significa solamente que el código fuente esté disponible; las condiciones de distribución del software Open Source cumplen una serie de requisitos:

La licencia permite la redistribución libre del software, sin imponer restricciones a ninguna parte para difundir o vender el software, también permiten la modificación del código fuente y la redistribución libre de los trabajos derivados del original, bajo las mismas condiciones impuestas por la licencia original. El software debe poderse distribuir en formato binario y el código fuente tiene que estar a disposición de todo el que lo quiera obtener.

[1] Se engloban bajo la definición de licencias Open Source todas aquellas licencias aceptadas por la *Open Source Initiative*: http://www.opensource.org/docs/definition_plain.php

4.1. Descripción del software utilizado para la implementación

4.1.1. Linux

Linux es un sistema operativo libre basado en Unix, fue creado originalmente por Linus Torvalds con la ayuda de desarrolladores de todo el mundo. El código fuente de Linux está disponible libremente bajo la licencia GNU General Public License.

La distribución de Linux sobre la que funciona el *HotSpot-in-a-Box* es Gentoo GNU/Linux, disponible en `http://www.gentoo.org`. Se ha elegido esta distribución por la versatilidad que ofrece a la hora de instalar y actualizar paquetes y la robustez que proporciona como sistema base.

La versión de kernel utilizada es la 2.4.22-ac4, que incluye el último parche de Alan Cox [2] disponible para la serie 2.4 del núcleo.

4.1.2. Wireless tools y Wireless Extensions

Las *wireless extensions* son una API genérica que proporciona una capa de abstracción para hacer posible que un driver envíe al espacio de usuario configuraciones y estadísticas específicas comunes a las redes inalámbricas.

Las *wireless tools* son una serie de utilidades que permiten manipular las wireless extensions.

El kernel 2.4.22-ac4 lleva incluida la versión 15 de *Wireless Extensions*, para obtener soporte de todos los rangos de frecuencias de 802.11a y 802.11g el núcleo del *HotSpot-in-a-Box* ha sido parcheado para incluir la versión 16.

Para obtener más información consultar el artículo *Wireless Tools for Linux* [2].

4.1.3. PCMCIA Card Services

Los *PCMCIA Card Services* proporcionan al sistema operativo soporte para este bus de datos, los módulos del kernel implementan la capa de acceso al hardware y el paquete pcmcia-cs-tools proporciona varias utilidades, como un servidor (card manager) capaz de responder a

[2] Alan Cox es un programador fuertemente involucrado en el desarrollo del núcleo de Linux

eventos como la inserción o extracción de una tarjeta para cargar el módulo —driver cliente— necesario.

En el *HotSpot-in-a-Box* se utiliza la versión 3.2.4 de pcmcia-cs-tools, disponibles bajo licencia GPL en `http://pcmcia-cs.sourceforge.net`. Estas utilidades, unidas a los scripts de inicio del sistema permiten detectar la tarjeta wi-fi insertada en el slot PCMCIA, cargando el módulo de kernel (driver) adecuado para configurar el punto de acceso.

4.1.4. HostAP

HostAP es un driver para tarjetas 802.11b basadas en el chipset Prism 2/2.5/3 desarrollado por Jouni Malinen y publicado bajo los terminos de la licencia GNU GPL. El código fuente está disponible en `http://hostap.epitest.fi/`. Este driver soporta el modo llamado Host AP y se encarga de las funciones de gestión (*management*) de IEEE 802.11 en el equipo, actuando como si éste fuera un punto de acceso. Esto no requiere ningún firmware especial en la tarjeta de red inalámbrica.

El firmware del chipset Prism2 se encarga (en modo Host AP) de tareas con tiempos críticos como el envío y reconocimiento de tramas de beacon, pero deja las otras tareas de gestión para el controlador (driver) el equipo.

En el *HotSpot-in-a-Box* este driver se usa en combinación con hostapd —un servidor que trabaja en espacio de usuario— configurado para soportar IEEE 802.1X, rekeying dinámico de WEP, accounting y autenticación 802.11 contra RADIUS entre otros. Estas características se encuentran detalladas en el capítulo 5.

4.1.5. Multiband Atheros Driver for WiFi (MADWIFI)

Madwifi es un driver para tarjetas 802.11a/b/g de tipo Cardbus, PCI o MiniPCI que contienen el chipset Atheros (ar5210, ar5211, ar5212).

Este driver permite que la tarjeta trabaje en modo Host AP, está portado a Linux a partir código original escrito para BSD por Sam Leffer y utiliza la API proporcionada por las *Wireless Extensions*.

El driver está dividido en tres módulos, ath_pci, ath_hal i wlan. El módulo “ath_pci” para dispositivos PCI/Cardbus, “wlan” contiene el soporte genérico del protocolo 802.11 y finalmente el módulo “ath_hal” contiene la capa de acceso al hardware (*hardware access layer*) y

controla la mayor parte de las operaciones específicas del chipset.

Los dos primeros módulos se distribuyen bajo licencias Open Source, en cambio el HAL se distribuye en forma binaria —bajo unos términos que permiten la redistribución— para cumplir con el reglamento establecido por la FCC [3].

La evolución de este driver es todavía incipiente, por esta razón el *HotSpot-in-a-Box* se ha desarrollado trabajando siempre con las últimas versiones de CVS. Se puede obtener el código fuente de madwifi y más documentación sobre éste en `http://sourceforge.net/projects/madwifi/`.

4.1.6. Bridge-utils

Las *bridge-utils* proporcionan al núcleo de Linux la implementación del protocolo IEEE 802.11d (ethernet bridging) y una interfaz de usuario para configurar las interfaces de red que formarán el bridge. Están disponibles en `http://bridge.sourceforge.net` y se distribuyen bajo licencia GPL. Su principal desarrollador es Lennert Buytenheck.

El *HotSpot-in-a-Box* utiliza estas utilidades para permitir la unión de una interfaz ethernet con una interfaz wireless.

Para ampliar información se puede consultar el "how-to" *Linux Ethernet Bridge* [3] de E. Béjar.

4.1.7. Netfilter e iptables

Netfilter e *iptables* forman el bloque que permite implementar filtrado, NAT/PAT y manipulación de paquetes de red en el núcleo de Linux. *Iptables* es una herramienta que normalmente trabaja en espacio de usuario y funciona con el framework proporcionado por *netfilter* en el núcleo.

Con *iptables* es posible diseñar una estructura de tablas para la definición de múltiples reglas (rulesets). Cada regla dentro de una tabla está formada por cierto número de clasificadores que permiten determinar si un paquete coincide con un cierto patrón especificado y una acción, que decidirá el destino del paquete.

---

[3] `http://ftp.fcc.gov/Bureaus/Engineering_Technology/Orders/2001/fcc01264.pdf`

El *HotSpot-in-a-Box* utiliza *netfilter/iptables* para realizar filtrado de paquetes, realizar NAT para compartir el acceso a Internet entre los clientes conectados y marcar ciertos paquetes para ayudar a iproute2 a realizar su tarea, tal como se verá en el apartado 4.1.8.

En la página oficial del proyecto, `http://www.netfilter.org`, se puede conseguir más información referente a *netfilter* e *iptables*, así como obtener el código fuente publicado bajo los términos de la licencia GPL.

4.1.8. Iproute2 y tc

La pila de protocolos de red es una de las áreas del núcleo de Linux que sufre más cambios y mejoras a medida que avanza el desarrollo de éste. Empezando en la versión 2.2 del núcleo, Alexey Kuznetsov introdujo un nuevo subsistema completo de encaminamiento IPv4 llamado *iproute2*, y un nuevo subsistema de modelado de tráfico (*traffic shapping*), controlado por la herramienta "tc" (*traffic control*).

Podemos decir pues que *iproute2* implementa la arquitectura utilizada por el sistema operativo GNU/Linux para proporcionar las capacidades de encaminamiento avanzadas de las que dispone. Para esto, en lugar de utilizar una única tabla de rutas, *iproute2* habilita múltiples tablas. Se decide que tabla de rutas se utilizará para un paquete determinado utilizando la base de datos de políticas de encaminamiento (*routing policy database*) que consiste en una lista de varias reglas. Cada regla tiene tres partes:

- *priority* La prioridad que indica el orden que se establecerá para atravesar la base de datos de políticas de encaminamiento.
- *match* Una expresión,para buscar los paquetes que coincidan y decidir a que paquetes se aplicará la acción. Esta expresión sirve para buscar coincidencias en los paquetes basándose en los siguientes campos:
  - IP de origen
  - IP de destino
  - valor del campo TOS (*Type of Service*)
  - interfaz de entrada
  - fwmark (*firewall mark*), una marca puesta en el paquete por *iptables*

- *action* La acción a realizar si un paquete coincide con la expresión indicada. Normalmente la acción apuntará a una de las tablas de encaminamiento.

El filtrado de paquetes basado en *iptables* también dispone de formas sofisticadas para buscar coincidencias en los paquetes, ofreciendo mayor flexibilidad la combinación de *iproute2* con *iptables*. Podemos seleccionar paquetes basándonos en sus flags TCP, número de puerto TCP/UDP, o incluso el estado de la conexión a la que pertenecen.

La interacción entre las reglas impuestas por *iptables* y las políticas de encaminamiento funciona tal como veremos a continuación.

Iptables asigna al paquete una marca (*fwmark*) de acuerdo a las reglas de filtrado de paquetes. Cuando el paquete tiene que ser encaminado y las políticas de encaminamiento tienen que tomar una decisión, se mira si hay alguna regla con la misma marca que contiene el paquete, entonces se realiza la acción apropiada, conectada con esta regla.

TC: Control de tráfico en Linux

Veamos ahora como se realiza el control de tráfico en el núcleo de Linux.

Una vez la pila de red dentro del núcleo ha tomado su decisión de encaminamiento se conoce hacia que dispositivo de red se tiene que enviar el paquete. Cada interfaz de red tiene cierta información adjunta a su estructura sobre como encolar los paquetes. Esta información es lo que los desarrolladores de Linux llaman disciplina de colas (*queueing discipline*) o *qdisc*.

Una disciplina de colas simple puede consistir en una única cola, donde todos los paquetes son almacenados con el mismo orden que llegan a la cola y que se vacía tan rápido como la capacidad que tiene el dispositivo de red para enviar.

Otras disciplinas de colas pueden usar filtros para distinguir entre clases diferentes de paquetes y procesar cada clase de una forma específica, por ejemplo, dando más prioridad a una clase sobre las otras.

Las clases y las disciplinas de colas están íntimamente atadas: la presencia de las clases y su semántica son propiedades fundamentales de la disciplina de cola. En contraste, los filtros pueden ser combinados arbitrariamente con disciplinas de colas y clases, en tanto en cuanto la disciplina de colas disponga de clases. Para incrementar todavía mas la flexibilidad que esto proporciona, cada clase puede utilizar otra disciplina de colas para encolar los paquetes. Esta disciplina de colas puede, por tanto, disponer de múltiples clases de nuevo, que a su vez pueden estar ligadas a sus propias disciplinas de colas, etc.

En el *HotSpot-in-a-Box* se decide hacia que tabla de encaminamiento se enviará un paquete en función de la marca puesta en el paquete mediante *iptables* a través del firewall que incorpora. En la configuración por defecto se han establecido 3 clases de usuarios distintas, que permiten controlar el ancho de banda disponible para cada usuario utilizando *tc*. La clase más baja dispone de un ancho de banda limitado a una cantidad fija y no puede obtener más de esa cantidad aunque el enlace esté libre (no haya tráfico). La segunda clase está también limitada a un ratio específico, aunque si no hay más tráfico en el enlace puede aprovechar la totalidad de éste. Por último la clase con más prioridad dispone de todo el ancho de banda del enlace, por encima de las otras dos clases.

Para ampliar conocimientos y conceptos sobre el encaminamiento avanzado disponible en Linux el lector puede consultar la documentación de Herald Welte [4]. Una aproximación mucho más practica a estos conceptos se puede encontrar en el *Linux Advanced Routing and Traffic Control HOWTO* [5].

4.1.9. ISC DHCP

El *Dynamic Host Configuration Protocol* (DHCP) es un protocolo que permite automatizar la configuración de equipos que utilizan TCP/IP. DHCP puede ser usado para asignar automáticamente direcciones IP y enviar otros parámetros de configuración TCP/IP como la mascara de red o la puerta de enlace.

Para cumplir este propósito en el *HotSpot-in-a-Box* se utiliza la versión 3 de la distribución de DHCP de *Internet Software Consortium* [4] (ISC), que se distribuye bajo una licencia Open Source en su misma página web.

4.1.10. ISC BIND

También distribuido por ISC bajo licencia Open Source podemos obtener BIND (*Berkeley Internet Name Domain*) versión 9, la implementación del protocolo DNS (*Domain Name System*) utilizada en el *HotSpot-in-a-Box* para traducir nombres a direcciones IP y viceversa.

En la configuración de BIND del *HotSpot-in-a-Box* se ha incluido un dominio que sólo es visible cuando se está conectado a éste y donde se han configurado varios subdominios para

[4] http://www.isc.org

permitir al usuario realizar acciones como cerrar su sesión simplemente escribiendo "*logout*" en la barra de direcciones de su navegador. Para que esto funcione independientemente del DNS que el cliente lleve configurado en su equipo [5] se ha añadido una regla de *iptables* que redirige todo el tráfico de DNS que el usuario realiza hacia el DNS instalado localmente en el *HotSpot-in-a-Box*.

4.1.11. FreeRADIUS

FreeRADIUS es el servidor RADIUS (*Remote Authentication Dial-In User Service*) Open Source utilizado más ampliamente, puede encontrarse en `http://www.freeradius.org`.

El protocolo RADIUS es un protocolo cliente/servidor que permite que servidores de acceso remotos se comuniquen con un servidor central para autentificar a los usuarios y autorizar su acceso al sistema o servicio para el cual han hecho la petición. RADIUS permite mantener perfiles de usuario en una base de datos central que comparten todos los servidores remotos. El hecho de tener un servidor central de autenticación proporciona mayor versatilidad para registrar el uso con propósitos de facturación o para hacer estadísticas.

FreeRADIUS es altamente configurable y ofrece alto rendimiento. Mientras que el servidor es conceptualmente similar a las variantes *Livingston*, tiene muchas más características y es mucho más configurable. Incluye módulos (*plug-in*) con soporte para MySQL, PostgreSQL, Oracle, IODBC, IBM DB2, MS-SQL, Sybase, LDAP, Kerberos, EAP, PAM, MS-CHAP y MPPE, autenticación *digest*, Python, X9.9, entre otros. Permite que el administrador modifique completamente el comportamiento para requisitos particulares de autenticación, accounting y almacenamiento de registros (*logging*).

En el *HotSpot-in-a-Box* se utiliza el servidor RADIUS para proporcionar un mecanismo centralizado para tomar decisiones de autenticación, tanto si se configura el *HotSpot-in-a-Box* para utilizar 802.1x tal como veremos en el capítulo siguiente, como si se permite la autenticación a cualquier cliente que desee conectarse a la red inalámbrica; en este último caso cuando el cliente abre su navegador se le presenta una página de acceso donde debe introducir su nombre de usuario y contraseña para la posterior validación de los datos contra el RADIUS.

[5] Aunque la configuración de red se realice mediante DHCP, es posible que un usuario haya especificado su servidor de DNS de forma estática, prevaleciendo éste sobre el valor obtenido por DHCP.

4.1.12. Apache

Apache es el servidor web Open Source utilizado en la mayor parte de los servidores de Internet, puede encontrarse en más información en `http://www.apache.org`.

En el *HotSpot-in-a-Box* se utiliza apache para servir el contenido de las páginas web que el usuario recibe para autenticarse antes de poder acceder a Internet.

mod_ssl

mod_ssl es un modulo que proporciona criptografía al servidor web *Apache* a través de los protocolos *Secure Sockets Layer* (SSL) y *Transport Layer Security* (TLS) con ayuda de la librería *OpenSSL* (véase 4.1.14). *mod_ssl* se distribuye bajo una licencia Open Source de tipo BSD y puede encontrarse en `http://www.modssl.org`.

En el *HotSpot-in-a-Box* se utiliza para proteger los datos del cliente (nombre de usuario y contraseña).

mod_rewrite

Gracias al módulo *mod_rewrite* —incluido en *Apache*— que permite reescribir las URLs recibidas en las peticiones realizadas por el cliente se ha implementado en el *HotSpot-in-a-Box* un mecanismo para poder ofrecer las páginas web que el cliente solicita incluso si éste tiene configurado un servidor proxy en las preferencias de su navegador. Esto se ha conseguido redirigiendo con *iptables* las peticiones realizadas a los puertos de proxy más comunes.

4.1.13. NoCat

NoCatAuth

NoCatAuth es la implementación Open Source de portal cautivo (*captive portal*) utilizada por el *HotSpot-in-a-Box*.

La idea de portal cautivo es la siguiente: Cuando un cliente intenta navegar por Internet automáticamente es redirigido a una página donde se le pide que se autentifique mediante usuario y contraseña. Esta información es validada contra un sistema de autenticación y si la información proporcionada es correcta entonces se le permite el acceso a Internet, modificando las reglas pertinentes en el firewall.

NoCatAuth está escrito en Perl y C y se encuentra disponible libremente bajo la licencia GPL en http://www.nocat.net.

La versión de *NoCatAuth* incluida en el *HotSpot-in-a-Box* ha sido modificada para poder realizar la autenticación contra un servidor RADIUS. El parche de RADIUS escrito por Jacob Barrett se encuentra disponible en http://www.pogozone.net/projects/nocat/.

NoCatSplash

NoCatSplash es un port de *NoCatAuth* escrito completamente en C, aunque todavía se encuentra en un estado prematuro de desarrollo y no es del todo funcional —todavía no soporta ningún mecanismo de autenticación— permite mostrar una página de bienvenida al usuario cuando este se conecta por primera vez.

Se ha creído conveniente utilizar *NoCatSplash* en el *HotSpot-in-a-Box* para poder mostrar una página al usuario cuando este ya se ha autenticado mediante 802.1x para obtener acceso a la red, de esta manera es posible mostrar al usuario una página con el contenido que se desee, por ejemplo los términos de uso del servicio, información sobre la red, publicidad o simplemente darle la bienvenida.

Hay que añadir que la versión utilizada en el *HotSpot-in-a-Box* es la última versión de desarrollo, disponible en el CVS y que contenía un error en el código que hacía imposible pasar de la pantalla de bienvenida. Este problema no ha sido resuelto en el momento de escribir estas lineas, si el lector desea más información sobre el problema y la solución aplicada en el *HotSpot-in-a-Box* puede consultar mi correo enviado a la lista de correo de NoCat el 30 de Noviembre de 2003, disponible en http:/lists.nocat.net/pipermail/nocat/2003-November/004058.html.

4.1.14. OpenSSL

Implementación Open Source de los porotocolos SSL (*Secure Sockets Layer*) y TLS (*Transport Layer Security*) disponible en http://www.openssl.org.

OpenSSL se utiliza en el hotspot para proporcionar una API criptográfica a diversos programas como FreeRADIUS, Apache (mod_ssl), OpenVPN y VTun.

4.1.15. OpenVPN y VTun

Para proporcionar soporte VPN (*Virtual Private Network*) cliente y servidor se han incluido en el *HotSpot-in-a-Box* dos de los programas Open Source más utilizados actualmente con este proposito. Estos son *OpenVPN* y *VTun*, cuyas características se detallan a continuación.

OpenVPN

OpenVPN es un servidor fácil de usar, robusto y muy configurable que puede utilizarse para enlazar de forma segura dos o más redes privadas utilizando un túnel IP cifrado a través de un sólo puerto TCP o UDP. Para ello utiliza todas las facilidades de encriptación, autenticación y certificación que proporciona *OpenSSL*. Para más información visitar la página `http://openvpn.sourceforge.net`.

VTun

VTun permite crear túneles virtuales a través de redes TCP/IP, con filtros de ancho de banda, compresión y encriptación. También permite crear túneles sobre IP, PPP, SLIP y ethernet entre otros. Se puede obtener más información en `http://vtun.sourceforge.net`.

CAPÍTULO

5

Características Funcionales

En este capítulo se presentan las características del *HotSpot-in-a-Box*, mostrando la cantidad de soluciones que se pueden implementar utilizando software Open Source como el que hemos visto en el capítulo anterior, y haciendo hincapié en los protocolos que brindan soluciones de acceso seguras a las redes Wi-Fi.

5.1. Estándares 802.11 soportados

Dentro de la familia de protocolos IEEE 802.11, el *HotSpot-in-a-Box* está preparado para funcionar con 802.11a, 802.11b y 802.11g.

Si se utiliza 802.11b en el *HotSpot-in-a-Box* con una tarjeta con chipset PRISM se pueden obtener todas las características comentadas en este capítulo; desgraciadamente, el driver *MADWIFI* que ofrece soporte para Linux de los estándares 802.11a y 802.11g todavía no está tan evolucionado como *HostAP*, por lo tanto cuando se utilicen estos estándares en el *HotSpot-in-a-Box* no se podrán utilizar algunas de sus características, ya que estas son ofrecidas por el

driver *HostAP* que sólo sirve para dispositivos con chipset PRISM.

5.1.1. IEEE 802.11b

802.11b es por excelencia el protocolo de la revolución Wi-Fi, siendo actualmente el estándar *de facto* para las redes inalámbricas en los hotspots públicos como cafeterías, restaurantes, parques, bibliotecas, hoteles y aeropuertos, que contribuyen a aumentar así su popularidad.

Este estándar ofrece un rango de cobertura excelente a una velocidad razonable (11 Mbps). Funciona utilizando DSSS (*Direct Sequence Spread Spectrum*) a 2,4 GHz, y selecciona automáticamente la velocidad (1, 2, 5.5 o 11 Mbps) dependiendo de la potencia de la señal recibida.

Debido a que la velocidad que ofrece es mayor que la media obtenida por el acceso a Internet que disponen la mayoría de los usuarios, 802.11b todavía se puede considerar una buena elección para uso general, aunque existan otros protocolos que proporcionen velocidades mucho mayores.

5.1.2. IEEE 802.11a

802.11a ofrece más canales, mayor velocidad (54 Mbps) y menos interferencia que los otros protocolos, aunque no ha llegado a alcanzar su misma popularidad.

Este estándar trabaja en la banda de 5 GHz utilizando OFDM (*Orthogonal Frequency Division Multiplexing*) y ofrece un rango de cobertura más reducido ya que con la misma ganancia las señales a 5 GHz no son capaces de alcanzar las distancias conseguidas por los 2,4 GHz.

El hecho de que 802.11a no se haya popularizado tanto como 802.11b y 802.11g hace que el equipamiento de red sea más caro, y por lo tanto todavía menos común. No se recomienda utilizarlo en hotspots públicos, ya que la mayoría de usuarios disponen de dispositivos 802.11b y 802.11g que no son compatibles con este estándar.

5.1.3. IEEE 802.11g

802.11g utiliza la misma codificación OFDM que 802.11a en la banda de 2,4 GHz, también es capaz de volver a DSSS para mantener compatibilidad con los dispositivos 802.11b. Por lo tanto 802.11g permite conseguir velocidades de 54 Mbps en la banda de 2,4 GHz mientras se mantiene la compatibilidad con el equipamiento de red 802.11b, permitiendo por ejemplo la reutilización de antenas si se actualiza el hardware.

Aunque es una tecnología relativamente nueva y un poco más cara que 802.11b, sus características la hacen realmente atractiva y por ello es cada vez más popular, por lo tanto podríamos afirmar que 802.11g será la tecnología que reemplazará poco a poco a 802.11b en el futuro, convirtiéndose en el estándar más aceptado para las redes locales inalámbricas.

5.2. Modos de funcionamiento

5.2.1. AP/Router

IP routing with NAT/PAT and firewall filters

En este modo de funcionamiento *HotSpot-in-a-Box* realizará NAT [1] con las interfaces WLAN y LAN, utilizando una sola IP pública proporcionada por la interfaz WAN, donde se puede utilizar con cualquier tipo de conexión de banda ancha, DSL, cable módem, RDSI o RTB.

Cuando el *HotSpot-in-a-Box* se configura en modo AP/Router se utiliza la interfaz WLAN en modo AP para compartir la conexión a Internet proporcionada por la interfaz WAN. También es posible unirlo a un *hub* o *switch* mediante la interfaz LAN para compartir la conexión con PCs fijos.

Bajo el punto de vista de un WISP esto permite utilizar el *HotSpot-in-a-Box* como *gateway* para equipos instalados en un cibercafé, *Business Center*, o portátiles que no disponen de tarjeta inalámbrica conectados a la conexión de red ofrecida por un hotel en la habitación del cliente, controlando el acceso a Internet no sólo para los clientes wireless sino también para estos equipos conectados al *switch*.

En entornos SOHO [2], se puede configurar el *HotSpot-in-a-Box* para impedir el acceso ilegitimo a Internet desde la red inalámbrica, pero permitiendo que salgan a Internet sin pasar por la *landing page* al resto de los equipos conectados al *switch*. También es posible realizar PAT [3] para redirigir puertos externos hacia un servidor interno, permitiendo así ofrecer servicios de cara a Internet.

---

[1] *Network Adress Translation*

[2] *Small Office / Home Office*

[3] *Port Address Translation*

Al ser una solución de routing basada en Linux es posible aprovechar todas las capacidades que proporciona el sistema operativo para ofrecer soluciones de encaminamiento avanzadas.

5.2.2. AP/Bridge

802.1d Ethernet and Wireless Bridging

Utilizando este modo el *HotSpot-in-a-Box* une las interfaces wireless i ethernet para que el tráfico fluya entre ellas de manera libre y de forma transparente. Las estaciones asociadas al AP se pueden comunicar entonces con otras estaciones wireless y con los equipos conectados a la LAN y viceversa.

Dado que esta configuración permite el acceso legitimo a la red cableada a las estaciones inalámbricas es aconsejable utilizar un buen método de autenticación en la interfaz WLAN.

5.2.3. AP/Repeater

Wireless Distribution System (WDS)

Este modo de funcionamiento permite extender la zona de cobertura de la red inalámbrica enlazando el *HotSpot-in-a-Box* con otros APs trabajando en el mismo canal.

Para hacer esto se utiliza una cabecera 802.11 con cuatro direcciones MAC, dos direcciones MAC para el emisor y receptor intermedios y otras dos para el emisor y receptor originales de la trama.

5.3. Métodos de autenticación implementados

5.3.1. Standard Web Access Method (Universal Browser Login)

Este método funciona autenticando al usuario a nivel de aplicación, es el método que utilizan la mayoría de los hotspots existentes en la actualidad, ya que al no depender de las capacidades del hardware utilizado por el usuario final para conectar a la red wireless es compatible con cualquier tecnología utilizada para obtener el acceso al medio.

Cuando un usuario situado en un hotspot abre el navegador en su ordenador portátil o PDA, independientemente de la página de inicio que éste tenga configurado es redirigido a la página de login o *landing page* (véase figura 5.1) donde debe introducir su nombre de usuario y contraseña para poder acceder a Internet. Para proteger las credenciales del usuario, la página de login se muestra utilizando el protocolo SSL (https) que proporciona 128 bits de encriptación.

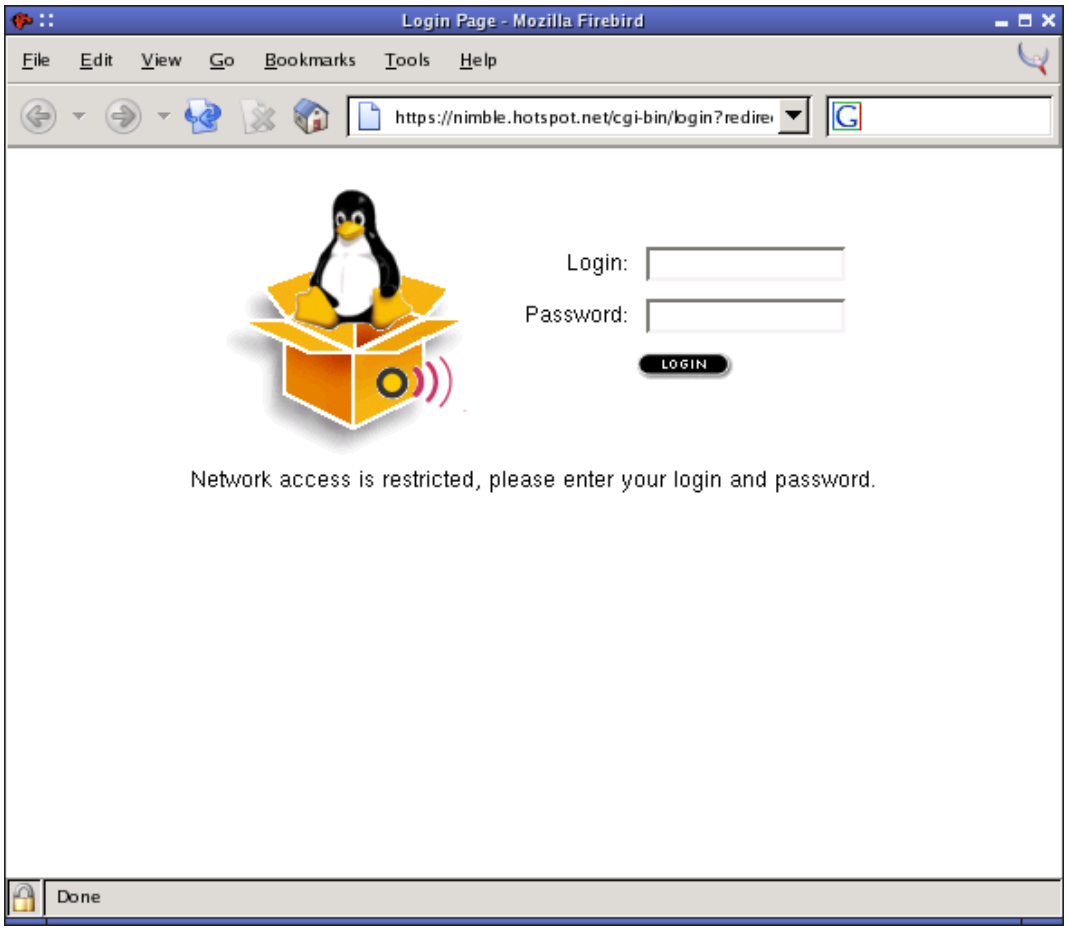


Figura 5.1: *Login Page*

Una vez el usuario se ha autenticado con éxito, se abre automáticamente una nueva ventana que le informa del estado de su sesión (véase figura 5.2), y es redirigido a la página que intentaba acceder inicialmente.

Desde la ventana de sesión el usuario puede ver el tiempo de conexión que le queda (*session timeout*) y también puede desconectar de Internet pulsando en el botón de “Logout”. En el

caso de que el usuario cerrara la ventana de sesión puede desconectarse escribiendo la palabra "logout" en la barra de direcciones del navegador. Si el usuario permanece conectado durante un tiempo sin generar tráfico será desconectado automáticamente al cabo del tiempo definido como *idle timeout.*

Figura 5.2: *Ventana de sesión*

5.3.2. MAC Authentication (MAC address ACL)

Cada interfaz de red dispone de una dirección física o MAC, que la identifica de forma única. Estas direcciones físicas son aplicadas a los dispositivos por los fabricantes. Es posible limitar el acceso a los usuarios filtrando por dirección MAC, para ello debemos especificar una lista de direcciones MAC y la política por defecto que se aplicará:

- Permitir el acceso, a menos que la MAC se encuentre en la lista
- Denegar el acceso, a menos que la MAC se encuentre en la lista

Hay que tener en cuenta que la autenticación basada en MAC no es una buena práctica, ya que es muy sencillo cambiar la dirección que se envía en las cabeceras 802.11 para suplantar la identidad de una dirección legitima.

5.3.3. IEEE 802.1X Athentication

IEEE 802.1X es un estándar de la IEEE que permite realizar autenticación y gestión de llaves con redes de área local IEEE 802, incluyendo Ethernet, Token Ring y FDDI.

Como IEEE 802.1X no trata el cifrado de datos, no es una alternativa a WEP, 3DES, AES o otro método de encriptación. IEEE 802.1X sólo especifica autenticación y gestión de llaves, no especifica ni como ni cuando los servicios de seguridad tienen que ser enviados utilizando las claves derivadas. De todos modos, puede ser usado para derivar autenticación y llaves de encriptación para utilizar con cualquier método de encriptación, y también puede ser usado para refrescar periódicamente estas llaves y re-autenticar asegurando así el uso de nuevas llaves en un corto periodo de tiempo.

IEEE 802.1X no es un sólo método de autenticación; utiliza *Extensible Authentication Protocol* (EAP) como *framework* de autenticación. Esto significa que los *switches* y puntos de acceso que funcionen con IEEE 802.1X pueden soportar una gran variedad de métodos de autenticación, incluyendo métodos basados en certificados, *smartcards*, *token cards*, *one-time passwords*, etc. De esta manera, los *switches* y puntos de acceso actúan sólo como una "pasarela" para EAP, se pueden añadir nuevos métodos de autenticación sin la necesidad de actualizar el *switch* o punto de acceso, simplemente añadiendo el software adecuado en la máquina cliente y en el *backend* de autenticación utilizado (normalmente el servidor RADIUS).

Como IEEE 802.1X no implica encapsulado (a diferencia de PPPOE o VPN) no añade *overhead* a los paquetes y se puede implementar en los *switches* y puntos de acceso sin producir impacto alguno en su carga. Esto significa que IEEE 802.1X puede habilitarse en el hardware existente actualizando el *firmware*, sin necesidad de comprar hardware nuevo. En los *hosts*, dado que IEEE 802.1X puede ser implementado por el driver de la tarjeta de red, el soporte de IEEE 802.1X se puede obtener simplemente utilizando un controlador de la tarjeta de red que permita el uso de este estándar.

IEEE 802.1X se integra correctamente con los estándares abiertos de autenticación, autorización y accounting (incluyendo RADIUS y LDAP). Los servidores RADIUS que soportan EAP se pueden utilizar para gestionar el acceso a las redes con IEEE 802.1X.

RADIUS-based ACL for IEEE 802.11 authentication

IEEE 802.1X define una serie de elementos llamados "*Supplicant*", "*Authenticator*", *Port Access Entity*" y "*Authentication Server*" cuyas definciones se pueden encontrar en la sección 1.1 (*Terminology*) del RFC 3580 [8] y se explican a continuación:

El *Supplicant* es un componente en la estación cliente que realiza la autenticación contra el servidor de autenticación (*Authentication Server*). El *HotSpot-in-a-Box* incluye un *Authenticator* que hace de pasarela para los paquetes que intercambian el *Supplicant* y el servidor de autenticación. Además de esto, tiene también una Entidad de Acceso al puerto (PAE) con la funcionalidad de *Authenticator* para controlar la autorización virtual del puerto, es decir, si se aceptan o no los paquetes de o hacia la estación cliente.

IEEE 802.1X utiliza *Extensible Authentication Protocol* (EAP). Las tramas entre el *Supplicant* y el *Authenticator* se envian utilizando EAPoL (*EAP over LAN*) y el *Authenticator* envía estas tramas al servidor de autenticación (y de forma similar, envía los mensajes del servidor de autenticación al *Supplicant*). La configuración más común es utilizar un servidor de autenticación externo y encapsular las tramas EAP en las tramas utilizadas por este servidor. RADIUS permite hacer esto, aunque IEEE 802.1X permite utilizar también otros mecanismos. En la figura 5.3 se muestra una sesión de autenticación 802.1X utilizando EAP entre un *Supplicant* y un servidor RADIUS.

El *HotSpot-in-a-Box* utiliza el driver *HostAP* para realizar la autenticación 802.1X que incluye la funcionalidad de PAE en el propio driver. La funcionalidad de PAE es un mecanismo relativamente simple para denegar las tramas normales que vienen de un puerto no autorizado. PAE permite que las tramas 802.1X pasen entre el *Supplicant* y el *Authenticator* incluso por un puerto no autorizado.

HostAP, además del driver para el núcleo incluye un servidor —hostapd— que trabaja en espacio de usuario, este servidor realiza la función de *Authenticator* y recibe las tramas 802.1X (EAPoL) del *Supplicant*. El *Authenticator* necesita un servidor de autenticación externo, en el *HotSpot-in-a-Box* el servidor *FreeRADIUS* incluido en el sistema es el que realiza esta función, aunque puede configurarse para utilizar cualquier servidor RADIUS que soporte EAP. El *Aut-*

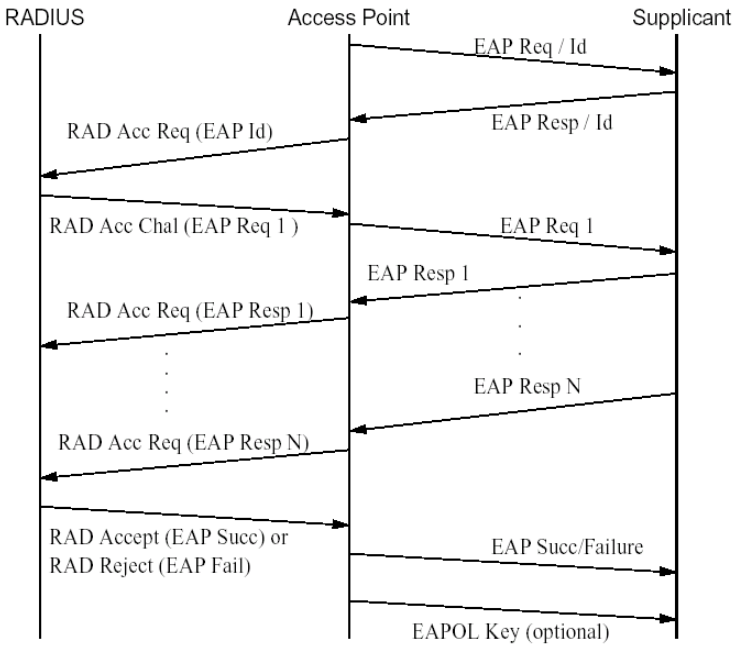


Figura 5.3: *Sesión completa de autenticación 802.1X mostrando los mensajes EAP y RADIUS*

henticator proporcionado por hostapd actúa como pasarela para las tramas que intercambian el servidor de autenticación (*FreeRADIUS*) y el *Supplicant* (la estación cliente), también controla la función de PAE proporcionada por el driver del kernel autorizando y desautorizando los puertos virtuales —conexión entre estación cliente y punto de acceso— basados en el estado de IEEE 802.1X.

Algoritmos de autenticación EAPoL

Los típos de autenticación que se utilizan de forma más común en los mensajes *Extensible Authentication Protocol over LAN* (EAPoL) incluyen EAP-MD5, EAP-TLS, EAP-TTLS, LEAP y PEAP entre otros. En el *HotSpot-in-a-Box* sólo se han implementado los que se describen a continuación.

EAP-MD5 (*Message Digest 5*) EAP-MD5 es un método de autenticación basado en la transmisión de un nombre de usuario y el hash MD5 de la contraseña. No se aconseja su uso en redes inalámbricas debido a las múltiples debilidades que presenta. Al no realizar autenti-

cación mútua, cualquiera puede suplantar la identidad del AP y devolver una respuesta de autenticación válida. Este método no proporciona un mecanismo para intercambiar nuevas llaves de forma segura, es decir, si se habilita EAP-MD5 no se podrá asignar la clave WEP de forma dinámica.

EAP-TLS (*Transport Layer Security*) EAP-TLS es un método de autenticación basado en certificados X.509. Proporciona autenticación mutua, por lo que requiere que el *Supplicant* y el servidor de autenticación dispongan de certificados, aunque esto lo hace mucho más resistente a ataques del tipo *man-in-the-middle*. Después de una autenticación realizada con éxito se establece un enlace seguro con TLS para enviar una llave de sesión unica desde el servidor de autenticación al *Authenticator*. Como este método precisa el uso de certificados X.509 se hace un poco más complejo de gestionar, sobre todo para el usuario (*Supplicant*).

EAP-TTLS (*Tunneled TLS*) EAP-TTLS utiliza un proceso de autenticación en dos fases. En la primera fase el servidor de autenticación es autenticado al *Supplicant* utilizando un certificado X.509. Utilizando TLS se establece un canal seguro a traves del cual el *Supplicant* puede ser autenticado al servidor de autenticación utilizando protocolos de autenticación heredados de PPP como PAP, CHAP y MS-CHAP. EAP-TTLS tiene la ventaja sobre EAP-TLS de que sólo necesita un certificado en el servidor de autenticación, esto hace que sea más sencillo de gestionar de cara al usuario final. EAP-TTLS tambien soporta ocultación de identidad, donde el *Authenticator* sólo conoce el nombre de usuario anónimo utilizado para establecer el canal TLS durante la primera fase, pero no conoce al usuario individual autenticado durante la segunda.

5.4. Métodos de Encriptación implementados

5.4.1. Broadcast and unicast Dynamic WEP Rekeying

Wired Equivalent Privacy (WEP) es un sistema de encriptación definido en el estándar 802.11 para prevenir el acceso a las redes inalámbricas por parte de intrusos y la captura del tráfico que los usuarios legítimos generan.

WEP está basado en el algoritmo de encriptación RC4, y utiliza llaves de 64 o 128 bits.

Han pasado ya cinco años desde que la IEEE hizo las especificaciones del protocolo WEP, desde entonces se han descubierto múltiples vulnerabilidades como las que se describen en el documento "(In)seguridad en redes 802.11b" [10], haciendo que hoy en día WEP sea insuficiente para cumplir el objetivo para el que fue diseñado originalmente.

En el *HotSpot-in-a-Box* se puede configurar WEP de forma estática, aunque no se recomienda su uso, para mejorar la seguridad proporcionada por WEP se ha añadido la opción de utilizar los mecanismos descritos a continuación.

- EAP/TLS genera una llave de sesión que puede ser usada para enviar las llaves WEP desde un punto de acceso a las estaciones autenticadas. El *Autenticator* se puede configurar para que seleccione automáticamente una llave broadcast aleatoria, compartida entre todas las estaciones autenticadas. Además es posible configurar llaves unicast, individuales para cada estación (es necesario que el driver de la tarjeta inalámbrica de la estación lo soporte).

- Las llaves WEP se pueden actualizar automáticamente utilizando *rekeying*. De esta forma se consigue mejorar la seguridad proporcionada por el protocolo WEP, ya que la misma llave WEP sólo será utilizada durante un corto periodo de tiempo, evitando así que se realicen ataques sobre ésta. En el *HotSpot-in-a-Box* se puede especificar el intervalo de *rekeying* en segundos si se utiliza EAP/TLS o EAP/TTLS.

5.5. Inter-Provider Roaming

El *Roaming* se define como la posibilidad de conectar con múltiples WISPs mientras se mantiene una suscripción con sólo uno. Es muy difícil que un sólo proveedor pueda construir una infraestructura que ofrezca acceso global a sus suscriptores, de ahí que el roaming entre proveedores sea esencial para poder expandir el número de sitios en los que el cliente pueda disfrutar de su acceso a Internet.

El comité WISPr de la *Wi-Fi Alliance* [4] creó a principios del año 2003 un documento [11] que recoge una serie de recomendaciones para implementar la tecnología que hace posible el roaming entre WISPs, de manera que sea posible facilitar la compatibilidad entre el mayor

[4] http://www.wi-fi.org

rango de productos Wi-Fi. RADIUS es el protocolo recomendado como *backend* para realizar las tareas de autenticación, autorización y accounting (AAA).

La figura 5.4 muestra el modelo genérico de roaming tal como lo describe el WISPr, incluyendo las funciones necesarias y los participantes.

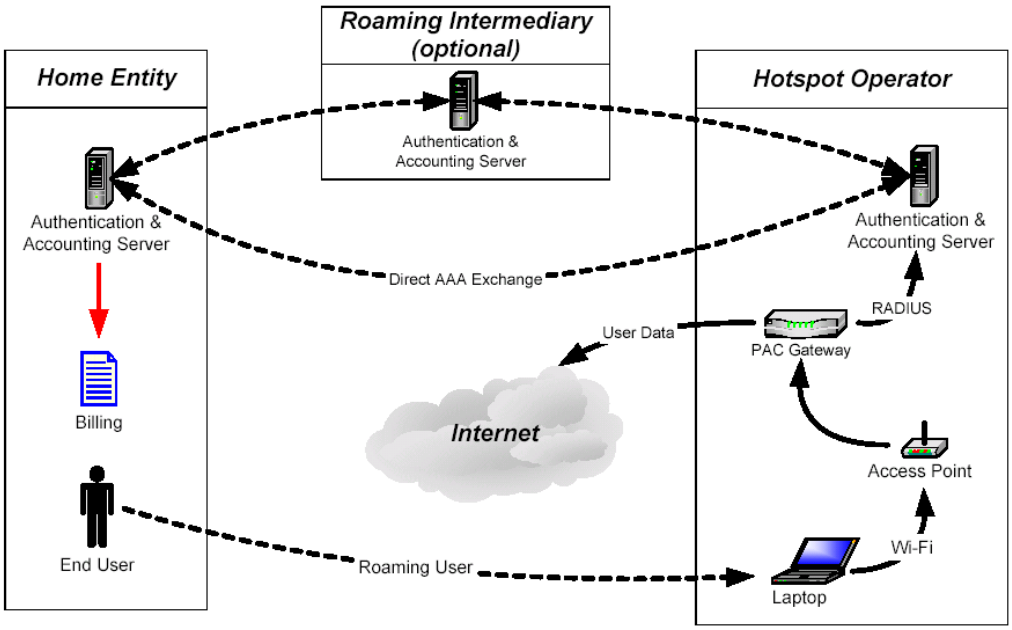


Figura 5.4: *Modelo genérico de roaming*

A continuación se detallan las características que debe soportar el servidor RADIUS de un WISP recomendadas por el WISPr, y que forman parte de la implementación realizada en el *HotSpot-in-a-Box*.

5.5.1. RADIUS Accounting

Los mensajes de accounting, descritos en el RFC 2866 [12], son esenciales para que los WISP puedan monitorizar y medir el uso que hacen de la red sus suscriptores, estos mensajes deben ser intercambiados por los servidores RADIUS de los WISPs que participen en el roaming en tiempo real para permitir el soporte de todo tipo de servicios basados en tarjetas prepago y suscripciones.

El servidor RADIUS (cliente) envía un paquete de inicio de accounting al RADIUS del WISP con el que tiene contratado el servicio el usuario en el momento en que éste inicia su sesión. Este paquete describe quien es el usuario y el tipo de servicio que éste recibe. De la misma manera, cuando el cliente finaliza la sesión el servidor RADIUS enviará un paquete de finalización de accounting, que, a parte de los datos mencionados anteriormente puede contener estadísticas como el tiempo de conexión del usuario o los octetos enviados y recibidos.

RADIUS Interim accounting updates

Los mensajes de interim accounting permiten minimizar el impacto producido por la perdida de un paquete de inicio o finalización de accounting. El servidor RADIUS del operador al que pertenece el cliente puede solicitar que el RADIUS del operador desde donde conecta el cliente envíe estos mensajes cada cierto intervalo de tiempo (El WISPr recomienda que éste no sea inferior a 10 minutos).

Esta solicitud de interim accounting será enviada por el servidor RADIUS después de autenticar al cliente, en el mensaje de *access-accept*.

5.5.2. RADIUS Multiple REALM support

El *REALM* sirve para identificar con que WISP tiene la suscripción un usuario. Cuando un usuario se identifica en un hotspot que pertenece al WISP con el que él ha contratado los servicios, por norma general no debe especificar el REALM, en cambio cuando éste quiere acceder a Internet desde un hotspot perteneciente a otro WISP (que tenga acuerdo de roaming con su operador) debe añadir el REALM a su nombre de usuario, normalmente esto se consigue añadiendo el nombre del WISP origen antes del nombre de usuario, separándolos por una barra (/) o después del nombre de usuario separándolos por una arroba (@).

5.6. Otras Funciones del *HotSpot-in-a-Box*

A parte de las características que se han descrito a lo largo de este capítulo el *HotSpot-in-a-Box* dispone de otras funciones que suelen ser comunes en dispositivos como éste que se encuentran actualmente en el mercado, estas son [5]:

- Servidor DNS
- Servidor DHCP
- SNMP Management
- Bandwitdh Management
- VPN server and client support

Además de estas funciones, el *HotSpot-in-a-Box* ofrece también otras características, que por ser específicas de entornos wireless se comentan con más detalle a continuación.

5.6.1. Wireless card autodetection and automatic AP setup

El *HotSpot-in-a-Box* detectará automáticamente la tarjeta wireless que se haya insertado en el slot PCMCIA siempre que ésta disponga de uno de los chipsets soportados y configurará automáticamente el punto de acceso.

Si es la primera vez que se utiliza los valores por defecto son los siguientes:

- **Modo de funcionamiento:** AP/Router
- **Interfaz WAN:** Obtener configuración de red por DHCP
- **Interfaz WLAN:** 192.168.2.254/255.255.255.0, SSID: hotspot
- **Interfaz LAN:** Deshabilitada
- **Autenticación:** Universal Browser Login
- **Cifrado:** Deshabilitado

[5] El software utilizado para implementar estas funciones se ha descrito con detalle en el capítulo 4.

5.6.2. Layer 2 user isolation

Esta es una de las funciones más demandadas por los WISP a los fabricantes de hardware, ya que permite hacer que dos clientes conectados al mismo punto de acceso no se vean entre sí, es por esto que se ha querido implementar esta función en el *HotSpot-in-a-Box*. *Cisco Systems* lo llama *Public Secure Packet Forwarding* (PSPF) y ofrece muy poca documentación [6] sobre la implementación de este protocolo.

Esta característica es interesante por que normalmente la gente utiliza equipos portátiles en hotspots públicos que después se conectan a las redes corporativas y entornos de trabajo de los clientes, si el usuario es incauto y deja sus unidades de red compartidas cualquier persona que se encuentre en el mismo hotspot publico podría acceder a sus archivos compartidos. Si se activa el aislamiento de capa 2, se bloquean las comunicaciones entre estaciones asociadas y los usuarios conectados al hotspot público no pueden verse entre sí, de modo que aunque el usuario tenga unidades compartidas nadie podrá acceder a estas.

La implementación realizada en el *HotSpot-in-a-Box* se basa en que trabajando en modo Infrastructura (Host AP modo Master), toda la comunicación entre estaciones cliente tiene que pasar a través del punto de acceso, el comportamiento normal del punto de acceso es hacer de "bridge" entre las estaciones asociadas, es decir, transmitir los paquetes que éstas se intercambian a nivel de enlace (capa 2) sin que estos pasen a capas más altas. Cuando se activa esta característica el *HotSpot-in-a-Box* instruye al driver *HostAP* para que no haga de "bridge" directamente entre estaciones asociadas, sino que pase los paquetes broadcast y multicast a la siguiente capa del kernel para que ésta los trate.

Para más información puede consultarse la función `bridge_packets()` en el código fuente de *HostAP*.

5.6.3. 802.11f Inter-Access Point Protocol (IAPP)

Inter-Access Point Protocol (IAPP) es un protocolo que todavía está en desarrollo entre IEEE 802.11f y el WorkGroup SEAMOBY de la IETF.

Este protocolo es utilizado por un punto de acceso para comunicarse con otro punto de acceso cuando ocurre algún evento determinado. La función de IAPP es facilitar la creación y

[6] `http://www.cisco.com/en/US/products/hw/wireless/ps458/prod_release_note09186a008007f7b8.html`

el mantenimiento del sistema de distribución que forman varios puntos de acceso conectados entre sí, soportar la movilidad transparente de estaciones 802.11 y forzar el requisito de una sola asociación para cada estación móvil.

Utilizando IAPP conseguimos que cuando una estación se autentique con un punto de acceso pueda cambiar a otro punto de acceso que forme parte del mismo sistema de distribución (mismo SSID) sin necesidad de volver a asociarse, reduciendo sustancialmente el tiempo de *handoff*.

El *HotSpot-in-a-Box* permite especificar la interfaz por la que los mensajes IAPP serán transmitidos, es interesante que esto se haga a través de la interfaz LAN si tenemos varios puntos de acceso conectados a un switch.

5.6.4. White Pages support (Wallet Garden)

Esta característica permite definir páginas de libre acceso, a las que cualquier cliente podrá acceder sin haberse autenticado previamente utilizando el *Universal Browser Login*.

Aprovechando esta característica es posible tener toda la *landing page* en un servidor de Internet, definiendo este servidor como *white page*. El único requisito para la *landing page* externa es que los formularios de login tendrán que utilizar el *backend* de autenticación proporcionado por el *HotSpot-in-a-Box*, para ello deberá utilizarse la dirección privada del *HotSpot-in-a-Box* en el método POST de los formularios de acceso. De esta manera es mucho más sencillo llevar a cabo una modificación en la *landing page* si se tienen diferentes *HotSpot-in-a-Box* distribuidos en lugares distintos, ya que basta con modificar la *landing page* colgada en Internet y no hay que retocar una por una las páginas locales de cada *HotSpot-in-a-Box*.

CAPÍTULO

6

Aplicaciones

En este capítulo se definen una serie de aplicaciones adicionales que se le pueden dar al *HotSpot-in-a-Box*, estas aplicaciones no han sido implementadas como parte del proyecto, pero se incluyen aquí para ofrecer una visión global de que otras cosas se pueden llegar a hacer con una solución wireless que esté basada en software libre.

Al utilizar GNU/Linux como sistema base, el *HotSpot-in-a-Box* no se limita sólo a las funciones implementadas en él como parte del proyecto, es posible aprovechar otras aplicaciones libres para implementar soluciones a medida, como por ejemplo las que se exponen a continuación.

6.1. Plug-n-share

Es muy común en entornos de oficina disponer de un servidor que tenga varios recursos compartidos, como por ejemplo impresoras o unidades de almacenamiento para hacer backups, etc. Si instalamos un *HotSpot-in-a-Box* en una oficina, podemos aprovechar sus capacidades

para ofrecer estos servicios, y ahorrarnos así los costes de un servidor adicional. De este modo, es posible instalar en el *HotSpot-in-a-Box* el paquete SAMBA [1] y conectarle a través de un puerto USB una impresora o un disco duro externo. Podemos entonces configurar SAMBA para convertir al hotspot en un potente servidor de impresión para la oficina, o utilizar el disco duro para compartir ficheros entre los distintos clientes. Podríamos utilizar también el paquete CUPS [2] (*Common UNIX Printing System*) para compartir la impresora con otras máquinas UNIX. También es posible configurar en el *HotSpot-in-a-Box* un servidor NFS o un servidor FTP para compartir ficheros.

Del mismo modo que con las impresoras y discos duros, podemos aprovechar uno de los puertos USB del *HotSpot-in-a-Box* para instalar una cámara web y utilizar el hotspot como herramienta de televigilancia.

¡Las posibilidades son infinitas gracias al software libre!

[1] http://www.samba.org
[2] http://www.cups.org

CAPÍTULO

7

Estudio del modelo de negocio del *HotSpot-in-a-Box*

En este capítulo se pretende ofrecer una visión del *HotSpot-in-a-Box* desde el punto de vista de un modelo de necgocio relacionado con el software libre, considerando como marco de referencia el avance experimentado por estas tecnologías en los últimos años.

NOTA: Este capítulo está en su mayor parte basado en el excelente artículo “Modelos de negocio basados en software libre” [13] de Fernando Monera, presentado en el congreso hispaLinux en Septiembre del 2003 y en los contenidos del libro “La pastilla roja” [14] de Alfredo Romeo y Juantomás García. En el apéndice A se incluye una copia del capítulo 7 del libro que trata del análisis financiero del software libre, distribuido por su autor en la página `http://www.lapastillaroja.net` bajo licencia *Creative Commons* según la cual se autoriza la copia y distribución del mismo sin ánimo de lucro.

7.1. El fenómeno Open Source

Tal como define Linus Torvalds el fenómeno Open Source en su libro *Just for fun* [15] una de las formas de entender el fenómeno Open Source es pensar en como la ciencia era percibida por la religión hace unos cientos de años. La ciencia era originalmente vista como algo peligroso y subversivo, básicamente como las compañías de software propietario ven a veces al Open Source.

El modelo Open Source permite que cualquiera participe en el desarrollo de un proyecto o en su explotación comercial, Linux es el ejemplo más claro de esto.

El software libre ha experimentado un crecimiento muy importante en los últimos años. La la creciente competitividad, el aumento de madurez del mercado tecnológico, el propio avance del software libre y su probada eficacia y calidad junto con una situación económicamente débil han provocado el comienzo de la asunción de un cambio de paradigma en el desarrollo y distribución de software. Los clientes empiezan a no estar satisfechos con la adquisición de productos, sino que quieren una personalización, una adaptación a sus problemas reales.

De esta forma muchas empresas han pasado de un modelo basado en venta de producto a otro basado en venta de servicios y soporte asociados. Es en este nuevo modelo donde el software libre es casi imbatible.

Utilizando soluciones basadas en software libre no es necesario estar constantemente reinventando la rueda. Si nos paramos a pensar un poco es un modelo muchísimo mas lógico desde el punto de vista del desarrollo de software. Los nuevos esfuerzos parten de código ya existente y disponible.

7.2. Integración de software

La integración de componentes o elementos de software no es un concepto nuevo. Desde hace muchos años el concepto de reutilización de código ha sido crítico en el éxito de una empresa de desarrollo. Lo que ha hecho el software libre es llevar la integración de software a límites mucho más extensos.

El *HotSpot-in-a-Box* es un software basado en la integración de diferentes productos Open Source, por ello se ha desarrollado evitando en la medida de lo posible la modificación de las aplicaciones a integrar, permitiendo así evolucionar los distintos elementos de la solución final con el mínimo impacto sobre la solución alcanzada.

7.3. Posibilidades de éxito del *HotSpot-in-a-Box*

El *HotSpot-in-a-Box* está por detrás de grandes productos comerciales como los ofrecidos por *Nomadix*™ o *Colubris Networks*™ en muchos aspectos (escalabilidad, soporte al cliente final, características técnicas, etc).

Las posibilidades de sobrevivir compitiendo con estos productos es nula en un ámbito puramente comercial. Pero el *HotSpot-in-a-Box* es software libre, cualquiera puede instalarlo y utilizarlo, y cubre todas las necesidades de la mayoría de las implantaciones que puede requerir un WISP. Este hecho puede fomentar la creación de WISPs, ya que la inversión inicial en software es nula, también puede ayudar a obtener ingresos a empresas dedicadas a ofrecer servicios de soporte y consultoría mediante las peticiones de personalización o solicitudes de nuevas características que algunos WISP podrían demandar.

Las mejoras logradas en el software pueden ser entonces incorporadas al propio producto, de modo que pasan a estar disponibles para todo el mundo en la siguiente versión. Debido a este proceso el *HotSpot-in-a-Box* puede verse inmerso en un desarrollo muy rápido y alcanzar la funcionalidad de sus hermanos mayores. Todo esto gracias a ser un producto libre y abierto.

7.4. Ventajas competitivas

La utilización del *HotSpot-in-a-Box* en un WISP ofrece una serie de ventajas competitivas muy importantes como las que se detallan a continuación, pero que suponen un cambio de mentalidad sobre el modelo actual basado en la compra de licencias asumido completamente por el entorno.

- Capacidad de modificación del código: Un WISP que utilice el *HotSpot-in-a-Box* tiene la capacidad para adaptarlo a sus necesidades, arreglar fallos operativos o de seguridad, etc.
- Independencia del proveedor: La implantación de una solución basada en software abierto permite al WISP la elección del mejor proveedor de servicios. La disponibilidad del código fuente y la capacidad para modificarlo permite que el WISP no quede atado a un determinado proveedor.
- Seguridad: La propia "comunidad de usuarios" desarrolla constantemente el trabajo de auditoría de código.
- Garantías de permanencia: La utilización sistemática de estándares hace difícil que el *HotSpot-in-a-Box* pueda quedar sin soporte. En el caso de que esto suceda, la disponibilidad del código permite que un grupo de usuarios u otra empresa pueda tomar el liderazgo en el desarrollo.

7.5. Ventajas para el desarrollador

Desde el punto de vista del WISP como empresa, contribuir en el desarrollo del *HotSpot-in-a-Box* también tiene una serie de ventajas indudables, como son:

- Disponibilidad de una comunidad potencial enorme de programadores y probadores (*beta testers*) del software
- Posibilidad de respuesta rápida ante clientes por problemas en el código, consecuencia de la anterior
- Evolución "automática" de las distintas piezas de software que componen el *HotSpot-in-a-Box*, como consecuencia de ser un software basado en integración

- Tendencia a una calidad enorme en el código desarrollado, como consecuencia de la necesidad de trabajar en equipo con personas desconocidas
- Tendencia a la utilización de estándares, lo que permite la construcción de soluciones mucho más completas mediante integración de distintas aplicaciones

7.6. Oportunidad de negocio

El cómo hacer dinero con el software libre es uno de los aspectos que más intrigan a las personas que no conocen de cerca este tipo de software. Siempre que se habla de software libre la pregunta es inevitable: "Bueno, ¿y dónde está vuestro beneficio si regaláis las aplicaciones?"

Lo que en realidad ocurre es que ganar dinero con software libre implica un cambio de mentalidad, dejar de cobrar por un producto concreto y cobrar por el servicio.

La oportunidad de negocio se encuentra en ofrecer servicios alrededor de una herramienta concreta, en este caso el *HotSpot-in-a-Box* desarrollado como parte de este proyecto. Cualquier WISP puede acceder a una plataforma en la que sólo tiene que invertir en los costes de implantación y mantenimiento.

La implantación y soporte técnico implican la puesta en marcha de toda la infraestructura técnica necesaria para que la herramienta opere de una manera óptima. Desde la implantación hasta los servicios de mantenimiento una empresa podría ofrecer al WISP interesado en el *HotSpot-in-a-Box* todos los servicios técnicos necesarios para su utilización. Incluso, podría vender el hardware asociado a la plataforma y, sobre el mismo, obtener un beneficio (*widget frosting*). También podría ofrecer servicios de formación y consultoría o dar la posibilidad de incorporar un servicio de reventa del *HotSpot-in-a-Box* con *marca blanca* entre sus líneas de negocio, realizando una personalización del software para el integrador de hardware que lo solicitara. Las posibilidades de personalización que el software libre trae, brindan la posibilidad de ajustar las aplicaciones a las necesidades de las organizaciones.

7.7. Análisis económico

Pongamos el caso de una cadena de restaurantes, que quiere ofrecer acceso inalámbrico a Internet en 30 de sus restaurantes. Vamos a analizar las diferencias que hay entre una implantación utilizando el *HotSpot-in-a-Box* y otra utilizando un producto comercial. Para ambos casos se estima en 4 años el ciclo de vida útil de la implantación tecnológica.

Basaremos la solución comercial en la compra del producto y la contratación de personal para el mantenimiento, y la solución basada en software libre en la compra del producto y la subcontratación de los servicios y soporte asociados a una empresa externa.

Para la solución comercial, la cadena de restaurantes contratará dos administradores de red para que se encarguen del mantenimiento, la gestión y la resolución de incidencias, se asume un salario medio de 18.000 euros/año con un incremento en el sueldo del 10 % anual. A esta solución tenemos que sumarle también los costes de formación de los administradores, consideramos que éstos deben acudir a un curso de formación de 20 horas con un coste de profesor de 50 euros/hora para las dos personas.

Para la solución basada en el *HotSpot-in-a-Box*, la cadena de restaurantes contratará el soporte de una empresa consultora de software libre. Esta empresa se encargará de la implantación y de ofrecer el soporte sobre la implantación realizada. El soporte consiste en comprobar y monitorizar que el rendimiento del sistema sea correcto, ofreciendo un paquete de soporte en función de las necesidades de la cadena de restaurantes, realizando un contrato de soporte por implantación y por paquetes de incidencias tanto a distancia como presenciales. Se estiman unos costes de soporte de 9.000 euros/año, se asume, asimismo, un incremento en el precio del servicio anual de un 10 %. Estos costes se calculan asumiendo que un trabajador de la consultora tiene un salario medio de 36.000 euros/año y puede cubrir el soporte de 10 clientes como la cadena de restaurantes, aplicamos al precio que le cuesta el empleado a la consultora un incremento del 150 % para que ésta pueda obtener también beneficios.

Para calcular los costes de instalación hemos asumido que el tiempo de *SetUp* necesario por restaurante es de 6 horas en el caso de la solución comercial y de 8 horas en el caso del *HotSpot-in-a-Box*, ya que este último resulta un poco más complejo de configurar. En el caso del producto comercial el restaurante contratará la instalación a una empresa externa, en cambio en el caso del *HotSpot-in-a-Box* la instalación será realizada por la consultora de software libre, la instalación se cuenta como coste adicional (contrato de implantación). En ambos casos el

coste del técnico que realiza la instalación es de 9 euros/hora.

El coste del *HotSpot-in-a-Box* se ha calculado en 600 euros (este precio sólo incluye el hardware), la solución comercial evaluada es un producto de *Colubris Networks*™ de características técnicas muy similares a las que ofrece el *HotSpot-in-a-Box* y que tiene un coste de 3162,50 euros.

Finalmente asumimos un precio de conectividad de 300 euros/mes que ofrecerá una velocidad de 2Mbps, de la misma manera, contaremos con un incremento en el precio de los servicios de conectividad de alrededor de un 10 % anual.

Resultados

	HotSpot-in-a-Box		**solución comercial**	
	Total	**%**	**Total**	**%**
Costes de Implantación	**20.160,00**	**25,63**	**97.495,00**	**34,66**
Hardware	18.000,00	22,89	94.875,00	33,72
Instalación & *SetUp*	2.160,00	2,74	1.620,00	0,57
Formación	0	0	1.000,00	0,35
AHORRO DE IMPLANTACIÓN	**77.335 euros (79,32 %)**			
Costes Años 0-4	**58.476,00**	**74,36**	**183.783,00**	**65,33**
Soporte	41.769,00	53,11	0	0
Sueldos y Salarios	0	0	167.076,00	59,39
Conectividad Internet	16.707,00	21,24	16.707,00	5,93
COSTES TOTALES	**78.636,00**	**100,00**	**281.278,00**	**100,00**
AHORRO	**202.642,00 (72,04)**			

Como podemos observar, existe una gran diferencia en el coste de implementación. La cadena de restaurantes, en el momento de la implantación de la tecnología, tendría que desembolsar 97.475 euros si optara por la solución comercial, mientras que con el *HotSpot-in-a-Box* sería de 20.160 euros.

Si la cadena de restaurantes quisiera cobrar la conexión a 3 euros la hora, ¿cuántas conexiones diarias tendría que tener para amortizar la inversión en un año? Utilizando el *HotSpot-in-a-Box* debería tener una media de 30,4 conexiones/día entre los 30 restaurantes, con la solución comercial debería tener 127 conexiones/día.

Supongamos ahora que la cadena de restaurantes quiere amortizar la inversión en un año y obtener beneficios durante los tres siguientes. Asumimos que entre los 30 restaurantes habrá una media de 15 conexiones de una hora cada día, por lo tanto la cadena deberá cobrar la hora de conexión a 6,1 euros si utiliza la solución basada en el *HotSpot-in-a-Box* o 25,4 euros si lo hace con la solución comercial.

Veamos ahora cuales serían los beneficios que obtendría la cadena cobrando a 6 euros la hora de conexión, asumiendo un aumento anual del 10 % en este precio, y con una media de 15 conexiones diarias, suponemos también que el número de conexiones aumentará un 10 % cada año.

	HotSpot-in-a-Box			**solución comercial**		
	Perdidas	**Ganancias**	**Balance**	**Perdidas**	**Ganancias**	**Balance**
Primer año	32.760,00	32.400,00	**-360,00**	137.095,00	32.400,00	**-104.695,00**
Segundo año	13.860,00	39.204,00	**25.344,00**	43.560,00	39.204,00	**-4.356,00**
Tercer año	15.246,00	47.436,84	**32.190,84**	47.916,00	47.436,84	**-479,16**
Cuarto año	16.770,60	57.398,58	**40.627,98**	52.707,60	57.399,58	**4.690,98**
TOTAL	**78.636,60**	**176.439,42**	**97.802,82**	**281.278,60**	**176.439,42**	**-104,839,18**

Podemos apreciar que para el caso de la cadena de restaurantes resulta mucho más rentable la solución basada en el *HotSpot-in-a-Box*, con estos parámetros la cadena tardaría 8 años en amortizar la inversión con la solución comercial, cuando la solución tecnológica ya se consideraría obsoleta.

Conclusiones

El objetivo inicial de este proyecto era implementar un *gateway* wireless y un *access point* en una misma máquina, utilizando únicamente herramientas libres. Es asombroso ver la cantidad de soluciones Open Source que existen para realizar todo tipo de tareas, gracias a la integración de estas herramientas se ha conseguido crear un producto con muchas posibilidades que permite utilizar cualquiera de los tres estándares más extendidos para comunicaciones wireless, IEEE 802.11 a/b/g.

Si se analizan los productos comerciales que existen para cumplir las funciones que realiza el *HotSpot-in-a-Box* uno se da cuenta de que todos tienen sus pros y sus contras, y habrá que buscar el que mas se ajuste a las necesidades concretas del entorno en el que se quiera implantar. A no ser que este entorno sea un caso muy típico, siempre nos encontraremos con el problema de que el producto A reune todas las características necesarias para ofrecernos la solución que necesitamos excepto una. Esta característica de la que carece el producto A la encontramos en el producto B, pero resulta que al producto B le faltan otras funciones que sí tenía el producto A. Por esta razón considero que el *HotSpot-in-a-Box* desarrollado como proyecto tiene una gran ventaja respecto a todas las soluciones comerciales: al tratarse de un producto abierto cualquiera puede implementar esa función que tanto necesita para que el producto se ajuste a las necesidades de su entorno de implantación.

Dado el carácter académico del proyecto, cabe reseñar los aspectos negativos (punto que no se aplicaría en un proyecto real, probablemente). Por falta de tiempo no se ha realizado una interfaz de configuración via web, sino que se ha hecho a través de menús en modo consola. Este aspecto es quizás uno de los puntos más débiles del *HotSpot-in-a-Box*, ya que todos sus competidores comerciales cuentan con este tipo de interfaces de configuración, que son mucho más amigables y a las que hoy en día los administradores de red estamos más que acostumbrados. Por esta razón en el análisis económico se han considerado ocho horas de instalación y configuración por equipo en el caso del *HotSpot-in-a-Box* y seis horas en el caso de la solución comercial.

Finalmente, como futuras prolongaciones del proyecto me gustaría poder añadir al *HotSpot-in-a-Box* soporte de IEEE 802.11i. Durante el último año y medio, el *Task Group* IEEE 802.11i ha desarrollado con mucho cuidado una especificación que proporciona un modelo muy robusto de seguridad para redes wireless, con encriptación AES y autenticación 802.1X. En poco tiempo, y gracias al avance del driver *HostAP* se podría incluir este estándar en el *HotSpot-in-a-Box*.

APÉNDICE

A

Análisis financiero del software libre

7 ANÁLISIS FINANCIERO DEL SOFTWARE LIBRE[1]

La inversión en tecnología ha cambiado radicalmente en los últimos tres años. Se ha pasado de una época en la que el crecimiento medio del gasto tecnológico rondaba el 30%, a una situación donde cada una de las empresas necesitan analizar minuciosamente cada gasto en tecnología. A finales de los 90 y comienzos de esta década, las empresas entraron en una carrera febril por hacerse con tecnología. Parecía que quien no tuviera el último CRM del mercado, no podría competir en la nueva situación.

Durante toda la década de los 90, la inversión tecnológica realizada por parte de las empresas fue de dos dígitos. Las empresas invertían en tecnología, en la cual veían como el fin de todos sus problemas. Sin embargo, cuando la ralentización económica llegó, las empresas optaron por congelar las inversiones tecnológicas y concentrarse sólo en aquellas que realmente ofrecieran la mejor solución a un problema en concreto. Así, en menos de tres años, se ha pasado de una situación en la que la empresa estaba al servicio de la tecnología, a una en la que la invesión tecnológica sólo es considerada si, realmente, ofrece un retorno meridiano, reduce los costes de una manera significativa o, bien, no tienen más remedio por las políticas de actualizaciones de las licencias.

La madurez de determinados mercados donde el incremento en la cifra de negocios no es tema baladí, hace que las empresas centren sus objetivos anuales en una reducción de costes más que en un incremento de los ingresos de la empresa. Las diferentes posibilidades que se tienen en tecnología son estudiadas ampliamente por las empresas que pretenden reducir al máximo los costes no relacionados con la generación de beneficio.

Herramientas financieras de control tecnológico

El gasto/inversión en tecnología es normalmente controlado gracias a herramientas financieras. Como acabamos de indicar, los responsables de los gastos en tecnología necesitan controlar la inversión tecnológica de los proyectos que lleven a cabo. A pesar del contexto en el cual nos encontramos, no todas las empresas controlan esta inversión de una manera rigurosa. En EE.UU, por ejemplo, el control de esta inversión se hace de diferentes maneras. Según una encuesta realizada en EE.UU, sólo el 30% de las empresas realizaban un análisis TCO frente un 29% que utilizaban un análisis de Payback, mientras que sólo el 18% de los encuestados empleaban la fórmula del ROI para controlar la inversión. La gran mayoría se decantaba por análisis no-financieros, como cumplir el proyecto a tiempo y, en presupuesto, reducción de costes o bien el incremento en productividad[2].

Aunque las herramientas más usadas son el TCO, el ROI y el Payback, existen metodologías mucho más complejas para analizar la inversión tecnológica. Estas herramientas habitualmente se usan en análisis de proyectos más complejos, donde la inversión tecnológica afecta a

1. Este capítulo está basado en el Informe Financiero del Software Libre (Abril, 2003). [Documento en línea]. Realizado por Open:service. Análisis Financiero del Software Libre. Disponible en **http://www.lapastillaroja.net**
2. Cuando realizamos el Informe Financiero en abril de 2003, incluimos estos datos procedentes de una encuesta realizada a diferentes CIOs. Perdimos el link del cual provenían estos datos, por lo que no podemos incluir el origen de los mismos. Apelamos a la confianza del lector para que crea en la honestidad de los datos aportados.

diferentes partes de la empresa, requiriendo, por ende, de modelos financieros costosos y personalizados, pero necesarios para el nivel de inversión que se maneja. Entre ellos, encontramos modelos basados en análisis cualitativo (*Balance Scorecard*, *Information Economics*, *Port - folio Mangement*, *IT Scorecard*) o en análisis probabilístico (*Real Options Valuation* y *Applied Information Economics*)[3].

Nuestro análisis se centrará en el *Total Cost of Ownership,* ya que creemos es la herramienta financiera que más se acerca a reflejar el impacto de una tecnología sobre la cuenta de pérdidas y ganancias, así como el impacto en el *cash-flow* de la organización. El ROI está centrado en observar el retorno en la inversión que una implantación tecnológica tendrá en la cuenta de pérdidas y ganancias de una empresa. La gran mayoría de las implantaciones tecnológicas no permiten estimar de una manera fiable cuál será la rentabilidad que la organización obtendrá por el hecho de implantar una tecnología. Por su parte, el *Payback* de una empresa se centra en observar el tiempo medio en que se recupera una inversión realizada. Para determinadas implantaciones, este cálculo es algo subjetivo, de ahí que nos centremos en que un manager de institución puede entender y calcular en base a su conocimiento: el TCO.

Total Cost of Ownership. Definición y Componentes

El *Total Cost of Ownership* (TCO) se define como el coste total de propiedad de una tecnología concreta sobre su periodo de vida útil. El TCO nos ofrece un análisis de todos los costes que supone la implantación de una tecnología. Esta medida es clave para poder entender las implicaciones de la tecnología en nuestra organizacion, tanto desde el punto de vista financiero como el organizativo.

Aunque, por definición, el TCO es el coste total de propiedad, se cae normalmente en una contradicción al usarlo para calcular implementaciones de software propietario. El software propietario no nos concede la propiedad sobre la tecnología, sino que nos da permiso para su uso, por lo que las utilización del término TCO no sería la correcta. Por tanto, y aunque se emplee el TCO como el coste de implantar una tecnología, el nombre apropiado para el software propietario sería el del Coste Total de Arrendamiento.

El modelo del TCO fue inventado por Gartner, en el año 1987, para poder analizar y mostrar los costes envueltos con inversiones tecnológicas, concretamente en el área *desktop*. Desde entonces, los modelos de TCO se han popularizado y numerosas consultoras tecnológicas han creado su modelo particular de Total Cost Ownership (TCO): Compaq, Forrester o la misma Gartner utilizan metodologías diferentes para calcular el TCO. Aunque los componentes del TCO son los mismos, como veremos más adelante, estas consultoras lo utilizan de manera diferente.

Aunque el análisis de los componentes del TCO es común para hallar el coste de implantación de la tecnología, los casos empresariales donde se emplean son bastante específicos por naturaleza. En todos los análisis que conduzcamos, los costes son propios de una situación en cuestión, por lo que no es riguroso decir que el TCO de una tecnología concreta es menor que otra. Sólo el análisis de cada situación nos puede ofrecer un resultado objetivo.

3. Podemos encontrar más información sobre este tipo de herramientas financieras en la website CIO.com, en la sección *Tools & Techniques*. [Web en línea].
Disponible en Internet: **http://www.cio.com/research/itvalue/tools.html**

Componentes del TCO

Los componentes que conforman el TCO son todos aquellos costes que intervienen como consecuencia de la introducción de una nueva tecnología. En principio, podemos hablar de dos tipos de costes, los directos e indirectos. Los costes directos son aquellos costes, normalmente, conocidos y que implican una contraprestación económica. Por su parte, los costes indirectos incluyen todos los costes que no tienen una identificación económica conocida, es decir, que pueden aparecer en mayor o menor medida a lo largo de la vida de la tecnología, siendo, por lo general, bastante difíciles de identificar y calificar.

Costes directos

Centrándonos en los costes directos, podemos analizarlos en cinco grandes grupos: software, hardware, costes de soporte, costes de administración y desinstalación del mismo.

Los costes directos son los costes que componen el mayor tanto por ciento del coste total de propiedad.En función de la tecnología que estemos implantando, el peso específico de cada uno de ellos cambiará.Si analizamos el TCO en una instalación de diez PCs en una PYME o unidad de negocio, los costes de mantenimiento y administración serán mínimos, mientras que el mayor coste está repercutido en los costes de software y hardware.

Tanto en los costes de hardware como de software factorizamos el precio de compra / licencias, así como las actualizaciones que del mismo hagamos durante el periodo del análisis. En los costes de soporte incluiremos los costes de instalación, mantenimiento, resolución de problemas, herramientas de soporte, libros, etc., definiendo previamente si son internos (realizados por los técnicos de la empresa) o externos (contratados a una empresa exterior, *outsourcing*). Por último, los costes de administración se componen de gestión del proyecto, desarrollo del sistema, administración del sistema (externa o interna), así como otros costes (compra de herramientas de desarrollo, etc.).

Costes indirectos

Exceptuando los costes de formación impartida, los costes indirectos son extremadamente difíciles de calcular. Se componen de costes bastante intangibles como el *downtime* (si quisiéramos calcularlo, tendríamos que asumir el tiempo medio que el sistema no funcionará en los próximos tres a cinco años, dependiendo del tiempo de vida esperado, *life cycle*-) o el *Futz factor* (variable que, dependiendo de la permisividad de la empresa, puede llegar a suponer un coste significativo en la producción de una empresa. **(Ver tabla en página siguiente).**

Los costes indirectos son los que no están relacionados con la implantación de la tecnología, sino con la aplicación de esta tecnología.

- Autoformación de los usuarios.
- *Downtime* (tiempo que una tecnología no funciona, pantallazos azules que obligan a reiniciar el aquipo).
- *Futz Factor* (tiempo de pérdida en un puesto de trabajo por el uso de las nuevas tecnologías en beneficio propio:uso del correo electrónico, Internet, etc.).

COMPONENTES DEL TCO & COMPARACIÓN SW LIBRE VS. SW PROPIETARIO

Costes Directos		
Tipo de Coste	**Componente**	**Sw Libre vs. Sw Propietario**
Software	Coste de Adquisición / Licencias	Muy Superior
	Actualizaciones	Muy Superior
Hardware	Coste de Adquisición / Licencias	Igual-Superior
	Actualizaciones	Igual-Superior
Costes de Soporte	Instalación & Setup	Igual
	Mantenimiento	Superior
	Resolución de Problemas	Igual
	Otros (libros, etc.)	Igual
Costes de Personal	Gestión de Proyecto	Igual
	Ingeniería / Desarrollo Sistemas	Superior
	Administración de Sistemas	Superior
	Otros (compra de libros, etc.)	Igual
	Formación	Igual
Costes Indirectos		
Tipo de Coste	**Componente**	**Sw Libre vs. Sw Propietario**
Costes de Soporte	*Futz Factor*	Igual
	Aprendizaje Casual	Inferior 1ª Fase
Downtime	*Downtime*	Muy Superior
Otros	Negociación de licencias	Muy Superior
	Auditoría de licencias	Muy Superior

Fuente: Elaboración propia a partir del Mitre Report[4].

Costes directos

Licencias / Actualizaciones / Software

Aunque el 100% de las aplicaciones, sistemas y herramientas de Software Libre se encuentran libres para su descarga de la Red, en muchas ocasiones, las empresas con distribuciones propias ofrecen paquetes integrados de software por el cual cargan un precio, normalmente mínimo (80-95% de descuento) en comparación con sus equivalentes en software propietario. Por ejemplo, el editor gráfico the GIMP o Scribus pueden ser descargados desde multitud de sitios en Internet o bien acudir a un proveedor de servicios informáticos que lo ofrezca empaquetado conteniendo el CD, un manual de documentación y un paquete de soporte técnico por un precio razonable.

Por su parte, el software propietario que se instala tiene asociado un coste por licencia, por puesto de trabajo, por acceso, etc. que, en función del volumen de una empresa, puede suponer un alto tanto por ciento de los costes totales de la misma.

Por tanto, y partiendo desde el hecho que el software propietario basa su modelo de negocio en **licencias**, en este apartado siempre el Software Libre será **muy superior al software_propietario**. **(Ver tabla en página siguiente).**

4. A KENWOOD, KAROLYN. (Octubre, 2001). *A Business Case Study of Open Source Software.* The Mitre Coorporation. [Documento en línea]. Disponible en Internet: **http://www.mitre.org/work/tech_papers/tech_papers_01/ kenwood_software**

PRECIOS DE APLICACIONES LIBRES VS. SW PROPIETARIO

	Software Libre		Software Propietario	
Desktop	**Aplicación**	**Precio (euros)**	**Aplicación**	**Precio (euros)**
Sistema Operativo	Debian 3.0	0-100,00	Windows XP	150-300,00
Paquete Ofimático	OpenOffice.org	incluido	MS Office	300-650,00
Editor Gráfico	GIMP	incluido	Photoshop	1.200,00
Groupware	Evolution	incluido	MS Outlook	incluido
Servidor		**0-2.499,00**		
Servidor web	Apache	incluido	MS Internet Inf.	1.200,00
Servidor seguridad	IP tables	incluido	Microsoft ISA	1.492,00
Servidor E-Commerce	Squidd	incluido		
Servidor Bases Datos	MySQL	incluido	SQL Server	19.244,00
Servidor de Correo	Opengroupware	A descargar	Exchange Enterprise	1.400,00

Fuente: Elaboración propia a partir de datos obtenidos de Optize. (**www.optize.es**) (Marzo 2003).

Los nuevos sistemas de cobro implantados por empresas de software propietario están creando auténticos problemas a las empresas. El sistema de licencias que Microsoft comenzó a utilizar en julio de 2002, además de crear auténticos quebraderos de cabeza a los clientes para su entendimiento (*MultiAnual Licencia Open*, *Open Multilicencia*, *Open Suscripción*, *Select License*, *Enterprise Agreement*, *Enterprise Agreement Suscription*), ha supuesto, en la mayoría de las ocasiones, un incremento sustancial en el precio pagado por licencias. Esta política de licencias ha supuesto grandes críticas por parte de los clientes de la empresa estadounidense. Con incrementos superiores, en muchos casos, al 100% sobre los precios pagados previamente[5], los clientes han reaccionado lentamente ante la actualización de sus licencias.

Ya hemos comentado el tiempo que el responsable de tecnología tiene que invertir en el estudio de las numerosas licencias que empresas propietarias ofrecen, en algunos casos difíciles de entender dada la complejidad de las mismas. Desde las licencias de acceso por cliente, tarifas por procesador, licencias en alquiler, actualizaciones, Software Assurance, OpenMultilicencia, OpenSelect, etc., los empresarios necesitan invertir un tiempo importante en poder analizar la rentabilidad de la inversión, ya que las distintas opciones existentes ofrecen diferencias importantes. Esta inversión en tiempo no se reduce al periodo de su adquisición, sino que se alarga en el tiempo debido a la monitorización que necesitan las mismas, por lo que las empresas tendrán que tener esto muy en cuenta a la hora del cálculo del TCO.

Hardware

Se ha debatido en muchas ocasiones sobre las necesidades de hardware que presenta, por una parte, el Software Libre y, por otra, el software propietario. Por la propia modularidad que tiene GNU/Linux, así como la escasez relativa de las líneas de código, normalmente se necesitan máquinas con menos potencia para correr GNU/Linux[6]. Por otra parte, la tendencia en este sentido parece que continuará igual. Gracias a la constante utilización por parte de Windows

5. (21 de Mayo 2002). *Gartner Alerts Clients to Review Microsoft Software Licensing Agreements Now to Prepare for July 31, 2002 Deadline*. [Documento en línea]. Gartner Inc.
Disponible en Internet: **http://www3.gartner.com/ 5_about/press_releases/2002_05/pr20020521a.jsp**

6. KENWOOD, CAROLYN. (Julio, 2001). *A Business Case Study of Open Source Software*.
[Documento en línea]. MITRE REPORT. Disponible en Internet:
http://www.mitre.org/work/tech_papers/tech_papers_01/kenwood_software/index.html

de *rich data formats*[7], la necesidad de utilización de disco duro es bastante mayor que la presentada por sistemas operativos libres[8].

Costes de soporte

Los costes de soporte incluyen la instalación y el *setup*, el mantenimiento, resolución de proplemas y otros costes derivados del soporte (compra de libros, etc.) Con respecto a los costes de instalación y set-up de una implantación GNU/Linux frente a una instalación Windows, consideramos que la empresa puede elegir los servicios profesionales de consultoras que puedan implantar una base tecnológica necesaria, y que el tiempo de implantación de la misma es parecida para ambas plataformas.

Además, no sólo las empresas de servicios informáticos ofrecen soporte, sino que también las innumerables websites ofrecen información y soporte sobre la aplicación o el sistema operativo en cuestión. Cuando se implanta un servidor Apache, con una base de datos MySQL mediante el lenguaje de programación PHP, el administrador tiene a su disposición multitud de recursos en las websites de los diferentes proyectos, lo que disminuye los costes de soporte de la empresa, ya que las propias comunidades que hay detrás de cada proyecto ofrecen multitud de recursos y soporte, lo que conlleva a una reducción de los costes de este tipo.

Consciente de la importancia de este canal de soporte que el Software Libre ha establecido y domina, Microsoft ha creado recientemente sus propias comunidades de usuarios para poder ofrecer soporte de una manera eficiente.

Una de las grandes ventajas del Software Libre ha sido el **coste mínimo de mantenimiento** que presenta un sistema GNU/Linux si se ha configurado apropiadamente. En diferentes ocasiones, las comparaciones entre servidores GNU/Linux frente a Windows han demostrado el mantenimiento mínimo de los mismos.

> *En Abril de 2002, PC Magazine, revista de informática, realiza un estudio comparativo entre los servidores de archivos e impresoras de Windows 2000 y GNU/Linux-Samba. El estudio concluyó, entre otras cosas, que el rendimiento de Samba era un 100% mayor que el de Windows 2000 y que los clientes gestionados por Samba llegan a ser hasta cuatro veces más que los soportados por Windows, siendo el mantenimiento de los mismos mínimo*[9].
>
> *Por su parte, el Robert Frances Group realizó, durante el primer semestre del 2002, una encuesta entre los directivos de tecnología de empresas Global 2000 para recoger diferentes datos sobre el uso de la tecnología en sus empresas, principalmente en las relacionadas con las arquitecturas servidores-clientes.*
>
> *Estos datos fueron posteriormente plasmados en un informe sobre el TCO de tres diferentes plataformas:Windows,GNU/Linux, Solaris (Total Cost of Ownership of GNU/Linux for the Enterprise), donde se analizaban cada una de las plataformas y el TCO de cada una de ellas.En el análisis del soporte el informe concluía que,cuando las empresas encuestadas*

7. Formatos ricos en datos, es decir, todos aquellos formatos como audio, vídeo, imágenes, etc.
8. GRYGUS, ANDREW. (17 de Julio, 2003). *2003 and Beyond.* [Documento en línea]. Disponible en Internet: **http://www.aaxnet.com/editor/edit029.html**
9. S HOWORTH, ROGER; STEVENS, ALAN; IT WEEK. (23 de Abril, 2002). *Samba run rigs around Win2000.* [Documento en línea]. ITWeek. Disponible en Internet: **http://www.itweek.co.uk/News/1131114**

se referían al número de servidores que un administrador de sistemas podría monitorizar, los administradores del sistema operativo GNU/Linux podían llegar a manejar de 40 a 60 servidores, frente a los de Microsoft, que lo hacían en torno a 10[10].

Costes de personal

La actual penetración de GNU/Linux está haciendo que el dominio de este sistema sea ya cada vez más extenso por parte de muchos administradores de sistemas. En la actualidad, una empresa que necesitara un administrador de sistemas GNU/Linux no tendría problema en contratar uno. Si acudimos al mayor portal de empleo de España, Infojobs.net, podemos ver el número de CVs enviados a puestos donde se necesita un administrador de sistemas. Como podemos ver, independientemente de la zona elegida, existen numerosos candidatos que puedan administrar, programar o crear aplicaciones de Software Libre, tanto en el área de servidores como en entorno *desktop.*

SUELDOS Y SALARIOS DE PERSONAL TÉCNICO GNU/LINUX

	Solicitud.	Sueldo Medio (euros)	Años Exp.	Ciudad
Administrador de Sistemas GNU/Linux	109	18.000 - 24.000	5	Madrid
Administrador de Sistemas GNU/Linux	123	12.000,00	1	Coslada (Madrid)
Técnico de Sistemas	186	12.000 - 18.000	1	La Coruña
Administrador de Sistemas GNU/Linux	123	18.000,00	1	Marbella (Málaga)
Ingeniero Programador GNU/Linux	45	12.000 - 18.000	-	Paterna (Valencia)
Administrador de Sistemas	82	12.000 - 18.000	1	Cartagena (Murcia)
Programador PHP	61	12.000 - 18.000	2	Vigo (Pontevedra)
Ingeniero de Sistemas GNU/Linux	139	18.000 - 30.000	1	Madrid
Técnico Informático	98	12.000 - 18.000	-	La Senia (Tarragona)
Programador PHP-MySQL	62	12.000 - 18.000	-	Bilbao
Técnico Soporte & Instalación	21	6.000 - 18.000	-	Córdoba
Administrador de Sistemas	39	12.000 - 18.000	3 a 5	Burjasot (Valencia)

Fuente: Infojobs (**www.infojobs.net**) (fecha de recopilación de datos: 17 de marzo de 2003).

Costes indirectos

Costes de soporte

Cuando hablamos de costes de soporte nos referimos a lo que le supone para una empresa las pérdidas de productividad por parte de los usuarios mediante el uso de la tecnología, bien sea por el desconocimiento de su uso, bien sea por una errónea utilización de la misma.

Los usuarios de las tecnologías en empresas normalmente se apoyan en los técnicos informáticos y en compañeros de trabajo para la resolución de problemas. Este hecho implica el conocimiento de la tecnología por parte de los usuarios de la empresa. La penetración del Software Libre en las empresas es todavía escasa como para comparar estos puntos. Hasta que se llegue a una penetración de mercado considerable, el Software Libre compara desfavorablemente ante el software propietario ante el aprendizaje casual. Las empresas siempre deberían tener en cuenta que la formación del usuario es una de las grandes claves para minimizar los costes derivados del aprendizaje casual.

10. Robert Frances Group. (Julio, 2002). *Total Cost of Ownership of GNU/Linux for the Enterprise*. [Documento en línea].
Disponible en Internet: **http://www-1.ibm.com/linux/RFG-LinuxTCO-vFINAL-Jul2002.pdf**

Por otra parte, el *Futz Factor* es un término acuñado por Gartner para referirse a las horas de producitividad perdidas por un usuario en el uso de la tecnología para fines personales. En la mayoría de los casos, los empleados utilizan tanto el correo electrónico como el navegador de Internet para fines propios, por lo que se entiende que es un coste que la empresa debe factorizar ante la implantación de una tecnología determinada.

Tanto tecnología libre como propietaria presentan el mismo factor de riesgo de verse afectados por este coste, el cual puede llegar a ocasionar grandes pérdidas para la empresa en términos de productividad, por lo que su comparación es neutra.

Downtime

El *downtime* es uno de los elementos que, no se suelen tener en cuenta a la hora de calcular el TCO, ya que es difícil calcularlo a priori. Éste se produce como consecuencia de un malfuncionamiento en la tecnología y las causas pueden ser diferentes. La más conocida y que más afecta a las empresas es la proliferación de virus. Un virus potente puede poner en jaque a los sistemas de una empresa con la consiguiente pérdida de productividad y, si son afectadas aplicaciones críticas, incluso de ingresos. No en vano, la empresa de seguridad británica Mi2g calcula en torno a veintidós mil millones de dólares las pérdidas producidas por virus a la economía mundial durante el año 2002[11].

Uno de los últimos casos conocidos es el del gusano Slammer, que atacó durante el pasado mes de enero de 2003 los servidores MS SQL Server de todo el mundo, consiguiendo ralentizar desde cajeros automáticos en EE.UU hasta sistemas de telefonía en Finlandia, pasando por malfuncionamiento de Internet en Japón y Corea del Sur. Este gusano es el primer código conocido que se puede clasificar como *Warhol Worm*[12]. Este tipo de códigos, a diferencia de sus antecesores, pueden infectar una red completamente en menos de quince minutos el gusano Slammer infectó al 90% de sus víctimas en menos de diez minutos por lo que los sistemas que no estén protegidos estarán irremediablemente condenados a ser infectados.

Aunque no podemos decir que el Software Libre esté libre de virus, los datos demuestran claramente cómo las empresas que presentan servidores basados en Windows reciben muchos más ataques que aquellas basadas en Software Libre.

La empresa estadounidense SecurityFocus, que monitoriza alrededor de 10.000 empresas en 150 países, comentaba que durante el 2001 los servidores IIS de Microsoft fueron atacados 17 millones de veces frente a 12.000 de los servidores Apache.

Recientemente se ha publicado un artículo, *Linux and the Knowledge Worker*[13], sobre el Ratio de Improductividad de Microsoft basado en el número de horas perdidas por parte de un *Knowledge Worker* durante un día de trabajo. Según el autor, durante la preparación de un documento de más de 100 páginas para su impresión, el autor invirtió más de tres horas de su tiempo en la reparación de errores sobre un tiempo total invertido de cinco y cuarenta y cinco minutos, o lo que arrija un Ratio de Improductividad de Microsoft de más de un 50%.

11. Extracto de *How to tackle Cybercrime Attacks* publicado en The Independent. [Documento en línea]. Mi2G Consulting .Disponible en Internet: **http://mi2g.com/cgi/mi2g/frameset.php?pageid=http%3A//mi2g.com/cgi/mi2g/press/180899.php**
12. WEAVER, NICHOLAS. (2001). *Warhol Worm. The Potential for very fast Internet Plagues*. [Documento en línea]. Disponible en Internet: **http://www.cs.berkeley.edu/~nweaver/warhol.html**
13. PETER, AARON. *Linux and the Knowledge Worker*. Desktop Linux. [Documento en línea]. Disponible en Internet: **http://www.desktoplinux.com/articles/AT8942921227.html**

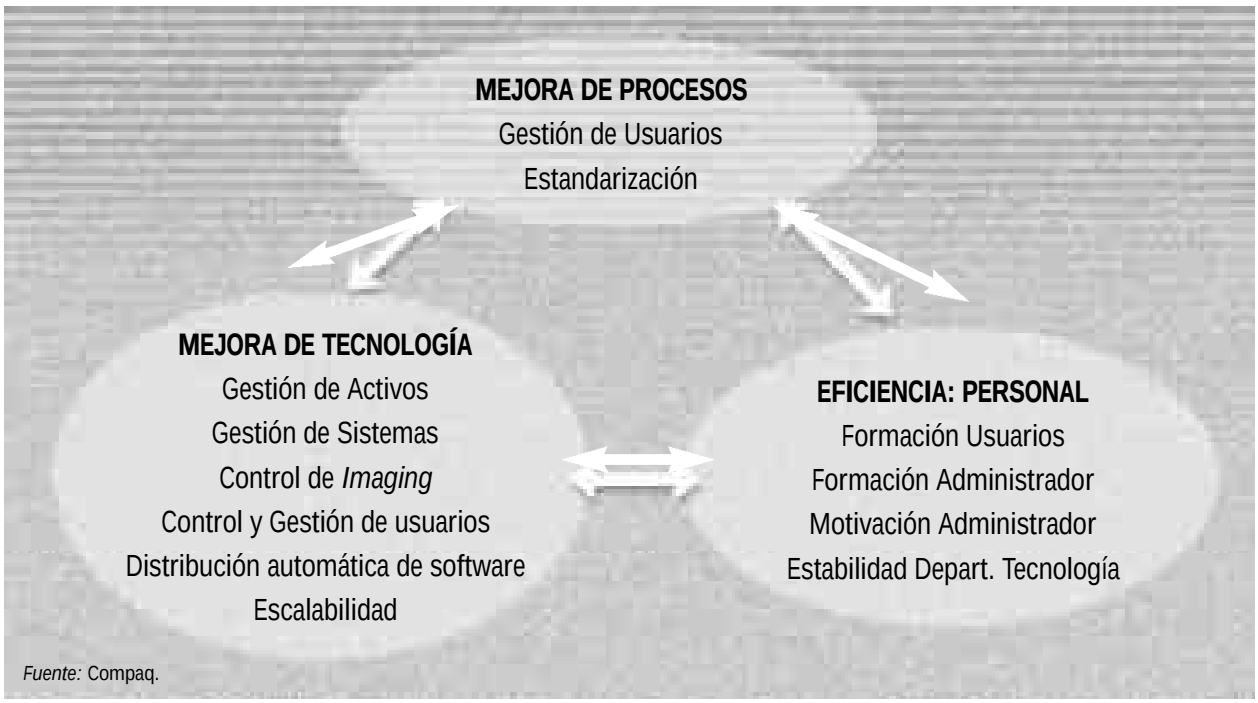
Por último, es necesario recalcar el estrés que los continuos malfuncionamientos en los equipos pueden llegar a derivar en situaciones dramáticas. El 5 de marzo de 2003, el propietario de un restaurante en Colorado disparó 4 veces a su ordenador portátil por las continuas veces que el ordenador se le colgaba[14].

Otros de los casos en los que podemos incurrir en *downtime* es por una incorrecta implantación de los sistemas. Para minimizar los costes derivados de este hecho, las empresas están comenzando a subcontratar el mantenimiento de parte o toda la infraestructura a la empresa de servicios informáticos. No en vano, la tendencia en *outsourcing* es creciente entre las empresas, especialmente en el aspecto tecnológico. Mediante la firma de *Service Level Agreements*, las partes implicadas pueden planificar los costes que puedan venir derivados de posibles malfuncionamientos en la tecnología.

Otra manera de mirar al Total Cost of Ownership

La empresa estadounidense Compaq, por su parte, introdujo su propio análisis sobre el TCO partiendo del modelo TCO desarrollado anterioremente, agrupando el análisis en tres grandes áreas: tecnología, procesos y eficiencia del personal.

Según Compaq, la reducción en el TCO de una determinada tecnología se maximiza cuando atacas las tres áreas conjuntamentente.



Fuente: Compaq.

Tecnología

La mejora en la tecnología viene determinada por la la eficacia de la nueva aplicación en cumplir con su misión. En función de la facilidad de su uso, la gestión de los activos, sistemas, usuarios,

14. (5 de Marzo, 2003). *Laptop blown away by enraged computer owner*. [Documento en línea]. USA TODAY. Disponible en Internet: **<http://www.usatoday.com/tech/news/2003-03-05-laptop-rage_x>**

de la distribución y la escalabilidad, la nueva tecnología deberá minimizar. Habrá que monitorizar estos componentes continuamente, ya que de ellos depende gran parte del TCO.

Procesos
Gracias a la introducción de la nueva tecnología, determinados procesos pueden ser racionalizados constituyendo un aumento en la eficiencia por parte del usuario de la aplicación. Así, los usuarios podrán centrarse más tiempo en las tareas propias de su organización. Gracias a la estandarización de la tecnología a usar por medio de toda la organización, los procesos se ven beneficiados.

Eficiencia de personal
La eficiencia del personal en el uso de la nueva tecnología es clave para que el TCO se vea reducido. Cuanta mayor sea la formación del personal en el uso de la tecnología en sí y el del administrador de la misma, menor será el coste de propiedad de la misma. Este aspecto es frecuentemente obviado por parte de las empresas, por lo que repercute enormemente en el TCO.

Análisis casos prácticos

Caso 1. 30 puestos de trabajo

Restricciones a tener en cuenta. Variables definidas
La empresa *Marketing a Distancia S.A* se plantea la apertura de una nueva unidad de negocio que tendrá como principal actividad la gestión de eventos y organizaciones. Este departamento es un centro de costes para la organización, por lo que el Director General de la empresa quiere conocer los costes envueltos de la tecnología a implantar para un periodo de tres años. Especialmente está interesado en conocer las diferencias que hay entre una implantación con Software Libre y una implantación con software propietario. Tras recibir la notificación, el Director de Informática realiza un análisis de los requerimientos tecnológicos que requiere la nueva unidad de negocio.

Variables a tener en cuenta
Una vez que el director informático es consciente de las necesidades que se tienen, comienza a elaborar un informe sobre los costes implícitos en la instalación de la tecnología necesaria:

Trabajadores
- – 25 Trabajadores a 20 euros/hora bruta
- – 5 Managers a 60 euros/hora bruta

Vida útil
- – Se estima en 4 años el ciclo de vida útil de la implantación tecnológica.

Tasa de descuento
- – Se calcula el coste de capital del 5% (valor actual neto).

Coste de red
- – Se calcula el coste de la red física en 100 euros por puesto de trabajo.

Aplicación: horas de desarrollo
- – Desarrollo de una aplicación a medida. Responsable informático calcula 500 horas.

Necesidades tecnológicas

Número de servidores:

– 1 Servidor por cada 40 puestos de trabajo en GNU/Linux.
– 1 Servidor por cada 20 puestos de trabajo en Windows.

Número de puestos de trabajo:

– 30 PCs

Aplicaciones necesarias:

– Desarrollo a medida: Aplicación de Base de Datos.
– Procesador de Textos / Hoja de cálculo / Editor de Presentaciones.
– Editor Profesional Gráfico.
– Gestión Información Personal & Correo Electrónico.
– Navegador de Internet.

Personal técnico necesario:

– Contratación de Personal: 1 Administrador de redes
– Contratación de Personal: Técnico de Sistemas

Contratación de soporte técnico:

– Se prevé una consultoría técnica por parte de una empresa exterior.

Precio Medio Hora Trabajador

– Se asume un precio medio bruto de la hora de cada uno de los trabajadores de 20 euros/hora, mientras que para los puestos ejecutivos se estima un precio de 60 euros/hora.

Especificaciones tecnológicas

– **Hardware: características y configuración**: se necesita la compra de 30 PCs con las siguientes características para cada uno de los puestos de trabajo: *Procesador Pentium III, Disco Duro 40 GB y Memoria 256 MB* como principales características. Asimismo, se requiere la compra de un servidor en el caso de la implantación de Software Libre y de dos unidades en el caso de la elección de software propietario (ver Informe Wheeler & Informe RFG Group[15]).

Aunque los responsables informáticos saben que el Software Libre corre en equipos de menor capacidad (ver **Análisis de Costes**), asumen que, independientemente del software que finalmente instale, el hardware será el mismo para ambas opciones.

– **Precios de Software**: es prácticamente imposible poder incluir un precio concreto para los precios de los productos de Microsoft. Dados los múltiples tipos de licencias que existen en la actualidad, no es fácil concretar unos precios porque dependerá del precio final que cargue el productor y/o canal. Para nuestro análisis hemos acudido a la tienda online informática Optize para la toma de datos. Sin embargo, seguramente estos precios podrían ser renegociados con un proveedor de servicios, por lo que los mismos podrían bajar, aunque no creemos de una manera sustancial (< 15%).

Los números incluidos se han obtenido de la licencia *Multilicencia A* de Microsoft para empresas con más de 5 puestos de trabajo y menos de 50. En ocasiones, Optize no ofrecía posibilidad

15. Robert Frances Group. (Julio, 2002). *Informe Total Cost of Ownership for GNU/Linux in the Enterprise*. [Documento en línea]. Disponible en Internet: **http://www-1.ibm.com/GNU/Linux/RFG-GNU/LinuxTCO-vFINAL-Jul2002.pdf**

a determinado software bajo esta modalidad, por lo que, en estos casos, hemos optado por el precio de *Licencias de Nuevo Usuario*.

Asimismo, no se ha elegido ninguno de los programas que ofrece Microsoft como *Software Assu - rance* destinados a sacar ventaja de todas las actualizaciones tecnológicas que lleven los productos de Microsoft. La inclusión de estos datos, inevitablemente, conllevaría un incremento en el TCO, por lo que se ha optado por la compra de las licencias sin estar adscritas a ningún programa de actualización.

Por parte del Software Libre se ha escogido un sistema operativo como RedHat, respaldado por una empresa con soporte comercial. Aunque, como se ha demostrado la estabilidad que ofrece Debian 3.0 sobre sistemas operativos libres comerciales es superior, en nuestro caso práctico la empresa opta por tener la seguridad de un producto empaquetado y soportado por una empresa, además de la Comunidad de usuarios. Gracias a la licencia que tiene RedHat, el mismo puede ser copiado y distribuido en más de un PC, por lo que no se necesita más de una compra de licencia para la instalación.

COMPONENTES DEL TCO & COMPARACIÓN SW LIBRE VS. SW PROPIETARIO

	Sw Libre		Sw Propietario	
	Aplicación	**Precio (euros)**	**Aplicación**	**Precio (euros)**
Desktop				
Sistema Operativo	Red Hat 8.0	302,50	Windows XP Professional	203,00
Paquete Ofimático	OpenOffice.org	incluido	Microsoft Office XP	375,00
Editor Gráfico	GIMP	incluido	Photoshop	1.200,00
Groupware	Evolution	incluido	MS Outlook	incluido
Paquete Antivirus	–	incluido	Symantec Antiv. SB Edit.	1.237,00
Servidor	–	incluido	Windows 2000 Server	4.373,00
Servidor Web	Apache	incluido	MS Internet Inf.	incluido
Servidor Seguridad	IP tables	incluido	MS ISA Server 2000	1.444,36
Servidor Bases de Datos	MySQL	incluido	MS SQL Enterprise 2000	21.335,00
Servidor de Correo	Opengroupware	Incluido	Exchange Server 2000	779,82
CALS	-	-	30 Clients Acc. Licenses	74,96

Fuente: Elaboración propia a partir de datos tomados en Optize (**www.optize.com**) (18 marzo 2003).

NÚMERO DE LICENCIAS NECESARIAS

Aplicación	**Nº Licencias**
Microsoft XP Full	30
Office XP Professional	30
Adobe Photoshop	3
Symantec Antivirus	30
Windows 2000 Advanced Server	2
SQL Server Enterprise	1
Exchange Enterprise	1
CALs Individuales	30
Microsoft ISA	1

Asimismo, se calcula que los *managers* de la empresa invertirán unas 15 horas en el estudio, análisis y decisión sobre las licencias propietarias.

– **Implantación & SetUp**: la implantación y el SetUp incluye la planificación del proyecto, así como el SetUp y el proyecto piloto de la Red. Hemos incluido los costes en que, normalmente, se incurre en la implantación de un sistema de este tipo: Planificación del Proyecto (Evaluación Hardware & Software, Plan de Comunicación, Desarrollo de Plan de Distribución), Desarrollo de Aplicación específica y SetUp.

Se ha incluido en este epígrafe el desarrollo de una aplicación en la que se invierten 500 horas de trabajo a un precio de 60 euros/hora. La misma se ha incluido tanto para desarrollo bajo Windows como para desarrollo bajo GNU/Linux. Otra de las variables que hemos tenido en cuenta ha sido un coste de 100 euros por cada uno de los puestos de trabajo para la instalación de la red física. Independientemente del tipo de software escogido, la implantación del sistema es similar para ambas plataformas.

Formación
Para calcular los costes de formación, hemos asumido que la gran mayoría de los usuarios conocen el manejo de programas propietarios y que no necesitan formación sobre los mismos. Sólo 5 de los 30 necesitarían una formación específica sobre los mismos. El administrador recibiría formación de 20 horas para avanzar en el funcionamiento de la red.

Si decidimos implantar Software Libre, toda la organización debería acudir a un curso de formación de 20 horas para el dominio de las aplicaciones libres, con un coste de profesor de 50 euros/hora en clases de 10 alumnos. Por su parte, el administrador recibiría 40 horas de formación sobre Administración de Sistemas GNU/Linux con un coste de hora por profesor de 100 euros.

Sueldos y Salarios
¿Cuántos administradores y profesional técnico se necesita para este sistema? En principio, tal y como hemos visto anteriormente, se requiere un administrador de sistemas que sea el que se encargue de la implantación de la red y del mantenimiento de los sistemas de la unidad de negocio. Con el soporte de una empresa profesional, un administrador de sistemas será quien dirija los mismos. Se asume un salario medio de 36.000 euros/año, independientemente del software escogido. Se asume un incremento en los sueldos del 10% anual.

Soporte
Asumimos unos costes de soporte de 10.000 euros/año para cada 50 PCs o 200 euros/ PC o, lo que es lo mismo, 5.000 euros/año. Se asume, asimismo, un incremento en el precio del servicio anual de un 10%, independientemente de la plataforma elegida.

Conectividad
Asumimos un precio de conectividad de 300 euros/mes que ofrecerán una velocidad de 2 Mbs más alojamiento de webs. Se asume, de la misma manera, un incremento en el precio de los servicios de conectividad de alrededor de un 10% anual.

Downtime
Hemos definido anteriormente, con respecto a los problemas existentes de *downtime*, los problemas asociados con la plataforma Microsoft sobre proliferación de virus y de *downtime*

en las empresas. Hemos visto cómo la empresa Dimension Data calculaba que el 67% de los trabajadores perdían, al menos, 1 hora a la semana o, lo que es lo mismo, alrededor de 4 horas al mes.

HORAS PERDIDAS AL MES DEBIDO AL DOWNTIME

Año 1		Año 2		Año 3		Año 4	
Sw Prop.	**Sw Libre**	**Sw Prop.**	**Sw Libre**	**Sw Prop.**	**Sw Libre**	**Sw Prop.**	**Sw Libre**
3	1	3	1	2	1	2	1

Dado el carácter conservador de nuestro informe, asumimos que los usuarios que trabajen con software propietario perderán una media de 3 horas/mes debido al *downtime* registrado, mientras que con el Software Libre asumimos que se perderá una media de una hora. Gracias al reforzamiento de las medidas de seguridad y de mejora en la estabilidad del sistema que está llevando a cabo Microsoft, estimamos que, a partir del tercer año, las horas de *downtime* perdidas mensuales decrecerá a 2 horas/mes, mientras que el Software Libre seguirá gozando de un entorno cuasi-libre de virus. Asimismo, asumimos para el Software Libre una pérdida de una hora en cuanto a operaciones de mantenimiento para cada uno de los cuatro años de vida útil de la tecnología.

RESULTADOS

COSTE TOTAL DE PROPIEDAD. CASO 1

	Sw Libre		Sw Propietario	
	Total	**%**	**Total**	**%**
Costes de Implantación	**68.630,00**	**21,66**	**126.219,00**	**30,54**
Implantacion & SetUp	36.100,00	11,39	35.810,00	8,66
Hardware	25.200,00	7,95	26.400,00	6,39
Licencias	330,00	0,10	61.509,00	14,88
Formación	7.000,00	2,21	2.500,00	0,60
AHORRO DE IMPLANTACIÓN		**57.589 euros (45,63%)**		
Costes Años 1-4	**248.244,00**	**78,34**	**287.057,00**	**69,46**
Soporte	21.474,00	6,78	21.474,00	5,20
Sueldos & Salarios	154.616,00	48,79	154.616,00	37,41
Conectividad Internet	30.923,00	9,76	30.923,00	7,48
Downtime	41.231,00	13,01	80.043,00	19,37
COSTES TOTALES	**316.875,00**	**100,00**	**413.276,00**	**100,00**
AHORRO		**96.401,00 euros (23,33%)**		

El análisis demuestra que las diferencias que se producen en la implementación entre una tecnología y otra radica, principalmente, en dos aspectos: coste de licencias y *downtime*, que llevan a una diferencia de coste en la tecnología de un 23,33% o lo que es un ahorro de 96.401 euros en un periodo de tres años. El coste de las licencias bajo el software propietario supone un 14,88% del TCO bajo este caso específico, mientras que el *downtime* en base a las variables introducidas es de un 19,37%.

Tan importante son las licencias como la seguridad y estabilidad del sistema en la tecnología. Un primer análisis del *downtime* potencial que se tiene en la actualidad con el software propietario, debido a los virus y la conocida baja estabilidad de los sistemas Windows, es un asunto al cual hay que prestar atención, ya que su repercusión en el TCO es bastante palpable.

Otro aspecto fundamental que se ha de tener en cuenta es la diferencia en el coste de implementación. La empresa, en el momento de la implantación de la tecnología, tendría que desembolsar 126.219 euros si optara por software propietario, mientras que con Software Libre sería de 68.630 euros, o una diferencia entre ambos de 57.589 euros, es decir, el 45,63%.

Caso 2. 100 puestos de trabajo

En nuestro caso 2 hemos procedido al cambio de algunas de las variables. Las especificaciones continúan pero, debido al número de nuevos PCs, el número de personal técnico se incrementa. En base al estudio del Robert Frances Group, asignamos un número determinado de servidores por cada uno de los administradores. Para un número de 3 y 5 servidores en plataformas GNU/Linux y Windows respectivamente, el número de administradores necesarios son los mismos en ambos casos.

Sin embargo,asumimos que, bajo una plataforma Windows, el número de técnicos necesarios se dobla debido a las condiciones apuntadas anteriormente. Las funciones que un técnico de Windows realiza son, por lo general, diferentes a las de un técnico de GNU/Linux, ya que las tareas del primero están dirigidas a la monitorización del sistema y arreglo de diversos malfuncionamientos en la implantación del sistema[16].

CAMBIO DE VARIABLES CON RESPECTO AL CASO PRÁCTICO ANTERIOR

	Sw Libre	Sw Propietario
Número de PCs	100	100
Número de Servidores	3	5
Número de Administradores	1	1
Número de Técnicos Necesarios	1	2
Cursos de Formación	10	2

Con estas nuevas especificaciones, el TCO cambia totalmente, como podemos ver en la **tabla de la siguiente página**.

En este caso, y cuando se ha producido una ampliación en la escala del departamento, vemos cómo el coste total se incrementa y la diferencia entre una implantación entre software propietario y Software Libre depende de cada uno de los casos. Los elementos más importantes son, por una parte, el coste de las licencias (ahorro de 125.000 euros en la implantación de un sistema GNU/Linux Vs. Windows) que hacen que la diferencia entre la implantación de un sistema y otro sea considerable, sobre todo, desde el punto de vista de inversión inicial para una empresa.

Para los cuatro años de vida útil de la tecnología, la diferencia entre una y otra se fundamenta, sobre todo, en los costes de personal, así como en el *downtime*. Con respecto a los primeros, nos hemos referido anteriormente a la diferencia entre las funciones de cada uno de los técnicos, mientras que el *downtime* continúa siendo una de las causas de mayor diferencia entre ambos sistemas.

16. Más información en Informe Wheeler: *Why Open Source / Free Software? Look at the Num - bers!*. [Documento en línea]. Disponible en Internet: **http://www.dwheeler.com/oss_fs_why.html**

COSTE TOTAL DE PROPIEDAD. CASO 2

	Sw Libre		Sw Propietario	
	Total	**%**	**Total**	**%**
Costes de Implantación	**134.030,00**	**22,65**	**249.314,00**	**27,98**
Implantacion & SetUp	36.100,00	6,10	35.810,00	4,02
Hardware	83.600,00	14,12	86.000,00	9,65
Licencias	330,00	0,06	125.004,00	14,03
Formación	14.000,00	2,37	2.500,00	0,28
AHORRO DE IMPLANTACIÓN		**115.285,00 euros (46,24%)**		
Costes Años 1-4	**488.759,00**	**78,48**	**672.801,00**	
Soporte	81.603,00	13,10	81.603,00	8,85
Sueldos & Salarios	231.925,00	37,24	309.232,00	33,54
Conectividad Internet	61.846,00	9,93	61.846,00	6,71
Downtime	113.385,00	18,21	220.120,00	23,87
COSTES TOTALES	**623.789,00**	**100,00**	**922.116,00**	**100,00**
AHORRO		**299.327,00 euros (32,46%)**		

APÉNDICE

B

GNU Free Documentation License

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The **"Document"**, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as **"you"**. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A **"Modified Version"** of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A **"Secondary Section"** is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The **"Invariant Sections"** are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The **"Cover Texts"** are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A **"Transparent"** copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called **"Opaque"**.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The **"Title Page"** means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License

requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section **"Entitled XYZ"** means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as **"Acknowledgements"**, **"Dedications"**, **"Endorsements"**, or **"History"**.) To **"Preserve the Title"** of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title

equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties–for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section

titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements".

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You

may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with...Texts." line with this:

> with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

BIBLIOGRAFÍA

[1] J. Tourrilhes. The devices, the drivers - 802.11b. http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Linux.Wireless.drivers.802.11b.html#Prism2, November 19 2003.

[2] J. Tourrilhes. Wireless Tools for Linux. http://www.hpl.hp.com/personal/Jean_Tourrilhes/Linux/Tools.html, October 27 2003.

[3] E. Béjar. Linux Ethernet Bridge. http://www.linkabu.net/linux/, November 8 2002.

[4] H. Welte. Advanced Linux Networking with iproute2 and tc. http://cvs.gnumonks.org/presentation/iproute2/, December 22 2000.

[5] B. Hubert. Linux advanced routing and traffic control HOWTO. http://www.lartc.org/howto/, October 29 2003.

[6] R. Flickenger. *Wireless Hacks*, pages 229–234. O'Reilly, September 2003. ISBN 0-596-00559-8.

[7] B. Aboba. The Unofficial 802.11 Security Web Page. http://www.drizzle.com/~aboba/IEEE/, January 6 2004.

[8] P. Congdon, B. Aboba, A.Smith, G. Zorn, and J. Roese. IEEE 802.1X Remote Authentication Dial In User Service (RADIUS). http://www.ietf.org/rfc/rfc3580.txt, September 2003.

[9] A. Mishra and W. Arbaugh. An Initial Security Analysis of the IEEE 802.1X Standard. http://www.cs.umd.edu/~waa/1x.pdf, February 2002.

[10] P. Oliva. (In)seguridad en redes 802.11b. http://pof.eslack.org/wireless/, March 2003.

[11] B. Anton, B. Bullock, and J. Short. Best Current Practices for Wireless Internet Service Provider (WISP) Roaming. http://www.wi-fi.org/opensection/downloads/WISPr_V1.0.pdf, February 2003.

[12] C. Rigney. RADIUS Accounting. http://www.ietf.org/rfc/rfc2866.txt, June 2000.

[13] F. Monera. Modelos de negocio basados en software libre. http://www.hispalinux.net/fotos/articulos/articulo_opensis.pdf, September 8 2003.

[14] A. Romeo and J. García. *La pastilla roja*. Edit Lin, November 2003. ISBN 84-932888-5-3.

[15] L. Torvalds and D. Diamond. *Just for fun: the story of an accidental revolutionary*, pages 225–235. HarperBusiness, June 2002. ISBN 0-06-662072-4.

GLOSARIO

API *Application Program Interface*. Una API es un método específico prescrito por el sistema operativo por el cual un programador que escribe una aplicación puede hacer peticiones al sistema operativo, página 12.

bridge Un bridge es un puente que sirve para unir dos interfaces de red para que el tráfico fluya entre ellas de manera libre y de forma transparente, página 14.

Cardbus Versión de 32-bits del bus de datos PCMCIA. Los Cardbus son de 32-bits y soportan *bus mastering* (un dispositivo del bus puede comunicarse con otros dispositivos en el mismo bus sin la intervención de la CPU), página 6.

chipset Chipset es un término genérico para un grupo de microprocesadores electrónicos (su significado literal viene del anglicismo *set of chips*), página 8.

CVS Concurrent Versioning System. Un sistema de control de código fuente que utilizan los desarrolladores para controlar los cambios en el código a través del tiempo realizados por múltiples personas, página 14.

firmware El firmware es el software que se encuentra incrustado en un dispositivo hardware, permitiendo la lectura y ejecución de éste, pero sin que pueda ser modificado por

el usuario final., página 13.

GNU GPL General Public License. Una licencia para la distribución de software libre que permite la copia, modificación y redistribución. Fue creada por la *Free Software Foundation* para proyectos como GNU, y ha sido aplicada también a Linux. Véase `http://www.gnu.org/copyleft/gpl.html`, página 12.

hotspot Los hotspots son lugares tales como aeropuertos, cafeterías, hoteles o parques que disponen de acceso a Internet a través de tecnología wireless, página 1.

mini-PCI Variante del bus PCI de tamaño reducido. Véase PCI, página 6.

PCI *Peripheral Component Interconnect*: Estándar que especifica una forma de conectar periféricos a la placa madre de un ordenador, página 6.

PCMCIA *Personal Computer Memory Card International Association*: Tarjetas utilizadas normalmente ordenadores portátiles, que permiten añadir dispositivos de red o memoria sin necesidad de apagar el equipo. También conocidas como *PC Cards*, página 6.

URL *Uniform Resource Locator* la dirección *World Wide Web* de un sitio de Internet. Contiene información sobre el método de acceso, el servidor a acceder y la ruta hacia el fichero a acceder, página 19.

WISP *Wireless Internet Service Provider*, proveedor de acceso a Internet inalámbrico, página 1.