Linux WWW HOWTO

di Wayne Leister, n3mtr@qis.net v0.82, 19 Novembre 1997

Questo documento contiene informazioni sul settaggio e l'installazione di servizi WWW in Linux (sia server che client). Questo testo non vuole essere un manuale dettagliato, ma semplicemente una panoramica nonché un punto di partenza per successive ricerche. Traduzione di Silvio Porcellana <sporc at mbox.vol.it>.

Indice

1 Introduzione

Molte persone usano Linux perché sono alla ricerca di un buon sistema operativo che sia anche *Internet compatibile*. Inoltre, ci sono istituti, università, enti non-profit e piccole imprese che vogliono mettere on line dei siti Internet con un budget limitato. È qui che il WWW-HOWTO vuole inserirsi. Questo documento spiega come far funzionare dei client e server per la parte più consistente di Internet - il *World Wide Web*.

Tutti i prezzi si intendono in dollari americani. Questo documento presuppone che si stia utilizzando Linux su una piattaforma Intel: le istruzioni e la disponibilità dei prodotti possono variare da piattaforma a piattaforma. Sono presenti inoltre molti link per il download del software: quando possibile, è preferibile utilizzare i siti mirror per ottenere un download più veloce e ridurre il traffico verso il server principale.

Il governo americano probisce alle aziende statunitensi l'esportazione di algoritmi di crittografia che superino i 40 bit. Pertanto le compagnie americane offrono di solito due versioni del loro software: una per il mercato domestico solitamente a 128 bit e una solo per l'esportazione a 40 bit. Ciò si applica sia ai browser sia ai server che supportano le transazioni sicure, altrimenti note come Secure Socket Layer (SSL).

1.1 Copyright

> Per motivi legati alla validità della licenza le note di copyright devono essere mantenute in lingua originale.

1.2 Feedback

Ogni forma di feedback è gradita. Non ho la pretesa di essere un esperto. Alcune di queste informazioni sono state reperite da siti scritti male: ci possono essere errori o omissioni. Accertatevi comunque di avere l'ultima versione del documento prima di inviare correzioni: il problema potrebbe essere stato risolto nella release più recente (nella prossima sezione è spiegato dove trovare le ultime versioni del documento). Ogni commento va inviato a: `n3mtr@qis.net`.

1.3 Nuove versioni di questo documento

Le nuove versioni di questo documento possono essere scaricate in formato testo da Sunsite a <http://sunsite.unc.edu/pub/Linux/docs/HOWTO/WWW-HOWTO> e da quasi qualunque altro sito mirror. È possibile vedere l'ultima versione HTML sul web a <http://sunsite.unc.edu/LDP/HOWTO/WWW-HOWTO.html>. Ci sono inoltre versioni HTML presso Sunsite in archivi tar.

2 Installazione di software client per WWW

Il prossimo capitolo è dedicato all'installazione di web browser. Se il vostro browser favorito non è menzionato, sentitevi liberi di contattarmi per farmelo sapere. In questa versione del documento, solo pochi browser hanno la loro sezione, ma ho cercato di includerli tutti (quelli che ho trovato...) nella sezione Panoramica. Nel futuro, quei browser che meritano una loro sezione l'avranno.

La sezione Panoramica è pensata per aiutarvi a decidere quale browser utilizzare e per fornire informazioni di base su ogni singolo browser. La sezione Dettaglio serve invece per aiutare nell'installazione, settaggio e manutenzione del browser.

Personalmente, preferisco Netscape: è l'unico browser che si mantiene aggiornato sulle ultime novità in fatto di HTML, quali frame, Java, Javascript, style sheet, transazioni sicure e layer. Non c'è niente di peggio che provare a visitare un sito e scoprire che non lo si può vedere perché il prorio browser non supporta alcune nuove caratteristiche.

Comunque, utilizzo Lynx quando non ho voglia di lanciare il "mostro" X-Windows/Netscape.

2.1 Panoramica

5 (Navigator/Communicator)

Netscape Navigator è l'unico browser qui citato che sfrutti appieno le nuove caratteristiche di HTML avanzato. Alcune di queste sono i frame, Java, Javascript, update automatico e layer. Funge anche da client per la posta e per la lettura delle news. È però un divoratore di risorse, sia di CPU che di memoria. Inoltre, crea una cache disco per ogni utente, sprecando un sacco di spazio. Netscape è un prodotto commerciale: le imprese hanno un periodo di prova di 30 giorni, ma non c'è alcun limite per i privati. Vorrei incoraggiare comunque tutti a registrarsi per supportare Netscape nella sua lotta contro Microsoft (e cosa sono poi 40 miseri dollari?). La mia paura è che se Microsoft vince, saremo tutti forzati ad usare Internet Explorer su una piattaforma Windows :(

3 (Lynx)

Lynx è uno tra i più piccoli browser: è gratuito e il codice sorgente è disponibile sotto la Licenza GNU. È il re dei browser testuali, offrendo molte funzioni speciali pur avendo solo un'interfaccia a caratteri.

Kfm

Kfm fa parte del K Desktop Environment (KDE). Si tratta di un sistema che funziona sotto X-Windows e che offre molte caratteristiche quali drag and drop, suoni, un cestino nonché un aspetto omogeneo. Kfm è una file manager, ma può essere utiizzato anche come browser. Non bisogna farsi ingannare dal nome, visto che, pur essendo un prodotto molto giovane, è molto comodo per navigare sul Web. Supporta i frame, le tabelle, il download ftp, la visualizzazione di archivi tar e molto altro. La versione corrente è la 1.39 ed è gratuita: è possibile utilizzarlo anche senza KDE, ma necessita comunque delle librerie che accompagnano questo windows manager. Maggiori informazioni possono essere reperite presso `<http://www.kde.org>`.

4 (Emacs)

Emacs è un "coltellino svizzero". È un word processor, un client per la posta e un browser. È molto difficile da utilizzare all'inizio perché bisogna imparare tutti i comandi associati alla tastiera, ma la versione per X-Windows è più semplice dal momento che tutte le funzioni sono sui menu. Un'ulteriore controindicazione è che è sostanzialmente testuale, e dunque le immagini possono essere visualizzate solo nella versione per X-Windows. Anche Emacs è gratuito e il codice sorgente è disponibile sotto la Licenza GNU.

NCSA Mosaic

Mosaic è un browser per X-Windows sviluppato dal Centro Nazionale delle Applicazioni per Supercomputer (NCSA) dell'Università dell'Illinois: NCSA ha impiegato quattro anni per completare il progetto e adesso si è rivolta verso altre possibilità. La versione più recente è la 2.6, rilasciata il 7 luglio 1995, il cui codice sorgente è disponibile per un uso non commerciale. *Spyglass Inc.* <http://www.spyglass.com> ha tutti i diritti commerciali su Mosaic. Questo è un browser per X-Windows molto solido, ma è carente per quanto riguarda le nuove caratteristiche dell'HTML. Per maggiori informazioni visitate la home page di Mosaic presso <http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/>. Il software può essere scaricato presso <ftp://ftp.ncsa.uiuc.edu/Mosaic/Unix/binaries/2.6/Mosaic-linux-2.6.Z>.

Arena

Arena è un browser per X-Windows concepito quando si stava testando l'HTML 3.0. Pertanto supporta tutte le caratteristiche di questa versione, dai fogli di stile alle tabelle. Lo sviluppo è stato portato avanti da Yggdrasil Computing con l'obiettivo di giungere alla creazione di un browser grafico gratuito e pienamente aggiornato. Nonostante ciò, lo sviluppo si è fermato nel febbraio del 1997 con la versione 0.3.11, che ha implementato solo una parte degli standard dell'HTML 3.2. Il codice sorgente è disponibile sotto la Licenza GNU. Informazioni presso `<http://www.yggdrasil.com/Products/Arena/>`. Può essere scaricato da `<ftp://ftp.yggdrasil.com/pub/dist/web/arena/>`.

Amaya

Amaya è il browser per X-Windows del W3C per HTML 3.2, di cui supporta tutti gli standard, nonché alcuni standard dell'HTML 4.0. Sono previsti tabelle, form, mappe client side, put publishing nonché i formati grafici GIF, JPEG e PNG. Si tratta sia di un browser che di uno strumento di authoring, giunto adesso alla versione 1.0 beta per il pubblico mentre la 1.1 beta per il test e lo sviluppo interno sarà rilasciata al più presto. Per informazioni, visitate il sito di Amaya a `<http://www.w3.org/Amaya/>`. Può essere scaricato presso `<ftp://ftp.w3.org/pub/Amaya-LINUX-ELF-1.0b.tar.gz>`.

Red Baron

Red Baron è un browser per X-Windows prodotto dalla Red Hat. È distribuito insieme alla distribuzione di Linux della Red Hat: non dispongo di molte informazioni, ma so che supporta i frame, i form e i SSL. Se usate Red Baron, per piacere aiutatemi a completare questa sezione. Per maggiori informazioni, visitate il sito Red Hat presso `<http://www.redhat.com>`

Chimera

Chimera è un browser X-Windows basilare: l'ultima versione, la 2.0 alfa 6, è stata rilasciata il 27 agosto del 1997 e supporta alcune caratteristiche dell'HTML 3.2. Per informazioni `<http://www.unlv.edu/chimera/>`. Chimera può essere scaricato presso `<ftp://ftp.cs.unlv.edu/pub/chimera-alpha/chimera-2.0a6.tar.gz>`.

Qweb

Qweb è un altro browser X-Windows molto semplice. Supporta tabelle, form nonché mappe server side. L'ultima versione è la 1.3. Per informazioni: `<http://sunsite.auc.dk/qweb/>` Il codice sorgente è disponibile presso `<http://sunsite.auc.dk/qweb/qweb-1.3.tar.gz>` I file binari, disponibili come RPM Red Hat, possono essere scaricati da `<http://sunsite.auc.dk/qweb/qweb-1.3-1.i386.rpm>`

Grail

Grail è un browser X-Windows sviluppato dalla Corporation for National Research Initiatives (CNRI). È scritto interamente in Python, un linguaggio ad oggetti interpretato. L'ultima versione è la 0.3 rilasciata il 7 maggio 1997: supporta form, bookmark, hystory, frame, tabelle e altre funzionalità previste dall'HTML 3.2.

Internet Explorer

Ci sono voci che annuncerebbero la Microsoft in procinto di portare Internet Explorer su svariate piattaforme Unix - tra cui forse Linux. Se è vero, ci stanno mettendo un sacco di tempo: se avete informazioni attendibili, fatevi vivi con un e-mail.

Secondo me, la maggior parte del software menzionato è praticamente inutilizzabile per una navigazione approfondita sul Web. Non è mia intenzione criticare gli autori - so che hanno lavorato molto seriamente sui loro progetti: il punto è che se tutte queste persone avessero lavorato su un unico prodotto, probabilmente sarebbe venuto fuori un browser gratuito che avrebbe potuto competere con Netscape o Internet Exlorer.

A mio parere, fra tutti i browser citati Netscape e Lynx sono i migliori. A seguire Kfm, Emacs-W3 e Mosaic.

3 Lynx

Lynx è uno dei più piccoli (l'eseguibile è di 600 K) e più veloci browser disponibili. Non necessita di molta banda né di grandi risorse hardware dal momento che lavora solo in modalità testo. Può funzionare su qualunque console, terminale o xterm. Non necessita di un sistema *X Windows* né di memoria addizionale.

3.1 Dove trovarlo

Sia la distribuzione Red Hat che la Slackware contengono Lynx. Pertanto non vi starò ad annoiare con i dettagli per compilare ed installare Lynx.

La versione più recente è la 2.7.1 e puo essere scaricata da `<http://www.slcc.edu/lynx/fote/>` o da quasi tutti i server FTP per Linux, come ad esempio `ftp://sunsite.unc.eduunder/pub/Linux/apps/www/broswers/` o i siti mirror.

Per maggiori informazioni su Lynx contattare:

Lynx Links

`<http://www.crl.com/~subir/lynx.html>`

Lynx Pages

`<http://lynx.browser.org>`

Lynx Help Pages

`<http://www.crl.com/~subir/lynx/lynx_help/lynx_help_main.html>` (questa è la pagina che si ottiene lanciando lynx –help o premendo ? in Lynx)

Nota: le pagine di help per Lynx sono state spostate di recente. Se avete una vecchia versione, dovrete cambiare lunx.cfg (in /usr/lib) e farlo puntare al nuovo indirizzo sopra menzionato.

Penso che la funzione più interessante di Lynx - nei confronti degli altri browser - sia la possibilità di scaricare in batch. È possibile scrivere uno script shell che scarica un documento, un file o qualunque altra cosa attraverso *http*, *FTP*, *gopher*, *WAIS*, *NNTP* o *file://* e lo salva su disco. Inoltre, è possibile riempire questionari e form in modalità batch semplicemente ridirigendo lo standard input e utilizzando l'opzione *-post_data*.

Per ulteriori funzioni speciali di Lynx, date un'occhiata ai file di help e alle pagine di manuale. Se utilizzate una caratteristica di Lynx che vorreste vedere aggiunta in questo documento, fatemelo sapere.

4 Emacs-W3

Ci sono diversi "gusti" di Emacs. I due più diffusi sono GNU Emacs e XEmacs. Gnu Emacs è ditribuito dalla Free Software Foundation ed è quello originale. È rivolto principalmente verso terminali testo, ma funziona anche sotto X-Windows. XEmacs (che una volta si chiamava Lucid Emacs) è una versione che gira solo sotto X-Windows sfruttandone molto le caratteristiche (menu migliori, ecc.)

4.1 Dove trovarlo

Sia la distribuzione Red Hat che la Slackware includono GNU Emacs.

La versione più recente è la 19.34. Non sembra esistere un sito web, mentre il sito FTP è `<ftp://ftp.gnu.ai.mit.edu/pub/gnu/>`.

L'ultima versione di XEmacs è la 20.2. Il sito FTP è a `<ftp://ftp.xemacs.org/pub/xemacs>`. Per maggiori informazioni si XEmacs, visitate `<http://www.xemacs.org>`.

Entrambi sono comunque disponibili dall'archivio Linux presso `ftp://sunsite.unc.eduunder/pub/Linux/apps/editors/emacs/`

Se avete GNU Emacs o XEmacs, probabilmente avete anche il browser W3.

Il modo W3 è un browser quasi completo scritto in Emacs Lisp. Principalmente tratta le informazioni in maniera testuale, ma può anche visualizzare immagini se utilizzato sotto X-Windows.

Per far funzionare XEmacs in modalità W3, bisogna andare nel menu 'apps' e selezionare 'browse the web'.

Io non uso Emacs, perciò se qualcuno vuole spiegare come entrare nel modo W3 lo aggiungerò a questo documento. La maggior parte di queste informazioni provengono dall'autore originale: se qualcuna è sbagliata, fatemelo sapere. Fatemi anche sapere se pensate che sia necessario aggiungere qualche altra informazione su Emacs.

5 Netscape Navigator/Communicator

5.1 Differenti versioni e opzioni.

Netscape Navigator è il re dei browser WWW. Può fare quasi tutto: d'altro canto, però, è uno dei programmi più affamati di memoria e risorse che io abbia mai visto.

Ci sono tre diverse versioni del programma:

Netscape Navigator include il browser, Netcaster (un client push) e un semplice programma di posta.

Netscape Communicator include il browser, l'editor, un programma di posta avanzato, un lettore di news, Netcaster e un'utility per le conferenze di gruppo.

Netscape Communicator Pro include tutto ciò contenuto nella suite del Communicator, più un calendario di gruppo, emulazione di terminale IBM e la possibilità di amministrazione remota (gli amministratori possono aggiornare migliaia di copie di Netscape senza lasciare la proprio scrivania).

Oltre alle tre versioni, ci sono altre due opzioni da scegliere.

La prima è fra installazione completa o di base. La completa include tutto, mentre quella di base comprende abbastanza per cominciare a navigare. È possibile scaricare i componenti addizionali quando si vuole (ad esempio supporto multimediale e Netcaster). Questi componenti possono essere installati attraverso l'utility di Netscape "Smart Update" (dopo l'installazione, andare in help->software updates). Al momento non è però disponibile la versione completa per Linux.

La seconda opzione è fra la versione per l'interno o per l'esportazione. Se vivete in Canada o negli Stati Uniti dovete scegliere la prima, che offre la più forte crittografazione a 128 bit per le transazioni sicure (SSL). La versione per l'esportazione è invece solo a 40 bit ed è l'unica ad essere autorizzata al di fuori di Canada e USA.

L'ultima versione di Netscape Navigator/Communicator/Communicator Pro è la 4.03. Ci sono due differenti versioni per Linux: uno per i vecchi kernel delle serie 1.2 e uno per i nuovi kernel 2.0. Se non avete ancora un kernel di questa serie, vi consiglio vivamente un upgrade: ci sono infatti molti miglioramenti.

Sono inoltre disponibili delle versioni beta, ma di solito durano solo un mese o giù di lì.

5.2 Dove trovarlo

Il miglior modo per ottenere il software Netscape è quello di scaricarlo dal loro sito web a <http://www.netscape.com/download/>. Ci sono dei menu che guidano alla scelta: la domanda sulla versione di Linux si riferisce al kernel (la maggior parte delle persone dovrebbe avere il 2.0). Se non si è sicuri della versione del proprio kernel, basta lanciare il comando 'cat /proc/version'. La versione per l'interno è disponibile solo attraverso il sito web.

Se invece volete la versione per l'esportazione, potete scaricarla direttamente dai server FTP della Netscape, che tra l'altro sono anche più aggiornati. Per esempio, quando ho scritto questo documento l'interfaccia web non aveva ancora la versione 4.03 non-beta per Linux, mentre il sito FTP sì. Ecco i link alle versioni esportabili per Linux:

Netscape Navigator 4.03 è a <ftp://ftp.netscape.com/pub/communicator/4.03/shipping/english/unix/linux20/navigator_standalone/navigator-v403-export.x86-unknown-linux2.0.tar.gz>

Netscape Communicator 4.03 per Linux 2.0 (kernel) è a <ftp://ftp.netscape.com/pub/communicator/4.03/shipping/english/unix/linux20/base_install/communicator-v403-export.x86-unknown-linux2.0.tar.gz>

Communicator Pro 4.03 per Linux non era ancora disponibile quando ho scritto questo documento.

Queste URL cambieranno quando usciranno nuove versioni: se i link non funzionano, potete trovare quelli giusti spulciando il sito FTP <ftp://ftp.netscape.com/pub/communicator/>.

Questi server sono a volte molto trafficati: è dunque preferibile aspettare le ore morte o scegliere un sito mirror. Comunque, preparatevi ad una lunga attesa dal momento che questi archivi sono grossi: Navigator è quasi 8 mega, mentre la versione base di Communicator 10.

5.3 Installazione

Questa sezione spiega come installare la versione quattro di Netscape Navigator, Communicator e Communicator Pro.

Per prima cosa, scompattate l'archivio in una directory temporanea. Poi, eseguite lo script `ns-install` (digitate `./ns-install`). In seguito, create un link simbolico dal file `/usr/local/netscape/netscape` a `/usr/local/bin/netscape` (digitate `ln -s /usr/local/netscape/netscape /usr/local/bin/netscape`). Infine, impostate la variabile d'ambiente `$MOZILLA_HOME` a `/usr/local/netscape`, di modo che Netscape possa trovare i propri file. Se utilizzate bash come shell, modificate `/etc/profile` aggiungendo le linee:

```
MOZILLA_HOME="/usr/local/netscape"
export MOZILLA_HOME
```

Dopo aver installato il software, esso si aggiornerà automaticamente con la "Smart Update". Basta lanciare Netscape come utente root e andare in help->software updates. È da qui, inoltre, che si installano gli altri componenti nel caso in cui abbiate la versione base.

Nota: lo "Smart Update" non rimuove le vecchie versione di Netscape, che vanno eliminate manualmente cancellando il file binario di Netscape e il file delle classi Java (per la versione 3).

6 Installare dei server WWW

Questa sezione contiene informazioni sui differenti server http e su strumenti addizionali quali linguaggi di scripting per programmazione CGI ecc. Esistono sul mercato dozzine di server web, ed io ho analizzato solo quelli pienamente funzionanti: dal momento che alcuni sono prodotti commerciali, non ho la possibilità di provarli. Una buona parte delle informazioni è stata trovata su svariati siti web e pertanto, in caso incontraste delle imperfezioni, fatemelo sapere.

Per una descrizione tecnica del meccanismo http, date un'occhiata agli RFC menzionati nel capitolo "Approfondimenti" di questo HOWTO.

Personalmente preferisco utilizzare il server Apache. Ha quasi tutte le caratteristiche di cui uno ha bisogno, e in più è gratis. Devo ammettere che questa sezione è sbilanciata nei confronti di Apache, ma è stata una mia scelta quella di concentrare i miei sforzi su questo server piuttosto che disperderli su tutti quanti. È probabile che mi occupi più approfonditamente degli altri in futuro.

6.1 Panoramica

Cern httpd

Questo fu il primo web server in assoluto: sviluppato dal Centro Europeo di Ricerche Nucleari (CERN), non è comunque più supportato. Sono noti alcuni gravi bug di questo server, nonché la sua lentezza e la "fame" di risorse. L'ultima versione è la 3.0, ma per maggiori informazioni è possibile rivolgersi alla home page del server http CERN presso `<http://www.w3.org/Daemon/Status.html>`. Si può scaricare presso `<ftp://sunsite.unc.edu/pub/Linux/apps/www/servers/httpd-3.0.term.tpz>` (non si tratta di un errore di battitura: l'estensione è veramente .tpz, ma probabilmente dovrebbe essere .tgz)

NCSA HTTPd

Il NCSA HTTPd server è il padre di Apache (lo sviluppo si divise in due server differenti): a causa di ciò, i file di setup sono molto simili. NCSA HTTPd è gratuito e il suo codice sorgente è disponibile per chiunque lo voglia. Non ho analizzato questo server nel documento, ma leggere la sezione su Apache può sicuramente aiutare. Questo server era molto diffuso, ma adesso è sempre più spesso soppiantato da Apache, che lo sostituisce in maniera perfetta (ha gli stessi file di configurazione) e ne risolve numerosi problemi. (fonte Nov. 1997 *Netcraft survey* `<http://www.netcraft.com/survey/>`). La versione piu recente è la 1.5.2a. Per maggiori informazioni, rivolgersi a `<http://hoohoo.ncsa.uiuc.edu>`.

7 (Apache)

Apache è il re di tutti i web server, e in più è distribuito in forma gratuita sia come binario che come codice sorgente. È molto modulare ed è pertanto molto semplice aggiungere caratteristiche e opzioni fra le tante disponibili: è inoltre molto diffuso, tanto che al momento attuale copre ben il 44% di tutti i domini web (50% se si contano anche i suoi derivati). Ci sono più di 695.000 server Apache in funzione (fonte Novembre 1997 *Netcraft survey* `<http://www.netcraft.com/survey/>`).

La versione ufficiale di Apache non ha l'SSL, ma ci sono due derivati che risolvono il problema.

Stronghold è un prodotto commerciale basato su Apache che costa $995 nella versione base e $495 in quella economica (basata su una vecchia versione di Apache). Stronghold è il secondo server sicuro dopo Netscape (fonte *C2 net* `<http://www.c2.net/products/stronghold>` e *Netcraft survey* `<http://www.netcraft.com/survey/>`). Per maggiori informazioni visitate il sito di Stronghold a `<http://www.c2.net/products/stronghold/>`. È stato sviluppato fuori dagli Stati Uniti e pertanto è disponibile in tutto il mondo nella versione a 128 bit.

Apache-SSL è una implementazione gratuita di SSL ma non per un uso commerciale (RSA ha un brevetto americano sulla tecnologia SSL). È possibile utilizzarlo per scopi non commerciali negli Stati Uniti se ci si collega con la libreria RSAREF (gratuita). Per ulteriori informazioni `<http://www.algroup.co.uk/Apache-SSL/>`.

Netscape Fast Track Server

Fast Track è stato sviluppato da Netscape, ma la versione per Linux è distribuita da Caldera, sul cui sito web è possibile trovarlo come Fast Track per OpenLinux. Non so se funzioni solo su OpenLinux della Caldera o se giri su ogni distribuzione (scrivetemi se avete la risposta). I server Netscape contano per l'11,5% (percentuale in discesa) fra tutti i web server (fonte settembre 1997 `<http://www.netcraft.com/survey/>`). Il server costa $295, ma è incluso nella distribuzione Caldera di OpenLinux, che costa $399 ($199.50 per le scuole). La pagina web descrive una bella interfaccia da amministratore per un setup di soli 10 minuti. Il server supporta SSL a 40 bit, mentre per quello a 128 c'e bisogno del Netscape Enterprise Server: purtroppo, però, questo server non è ancora disponibile per Linux :(L'ultima versione di Fast Track è la 2.0 (la versione 3 è in fase beta, ma non è stata ancora portata su Linux). Per comprarne una copia, basta andare sul sito Caldera a `<http://www.caldera.com/products/netscape/netscape.html>` Per maggiori informazioni, la pagina di Fast Track è `<http://www.netscape.com/comprod/server_central/product/fast_track/>`

WN

WN ha molte caratteristiche che lo rendono interessante. Per prima cosa, è più piccolo dei server CERN, NCSA HTTPd e Apache. Ha inoltre molte funzionalità senza le quali ci sarebbe bisogno di programmazione CGI, quali ad esempio ricerca sul sito, includes avanzati dal lato del server: offre inoltre la possibilità di scaricare solo una parte di file con la sua opzione "ranges". È rilasciato sotto la Licenza Pubblica GNU. Le versione corrente è la 1.18.3: per maggiori informazioni rivolgersi a `<http://hopf.math.nwu.edu/>`.

AOLserver

AOLserver è prodotto da America On Line. Devo ammettere di essere rimasto sorpreso dalle potenzialità di un server web scritto da AOL. Oltre alle caratteristiche standard, è supportata infatti la connettività ai database: le pagine possono interrogare un database attraverso dei comandi SQL, e il database è accessibile attraverso la Open Database Connectivity (OBCD). Il server ha inoltre un motore di ricerca incorporato nonché il supporto per gli script TCL: se ciò non fosse abbastanza, è possibile aggiungere i propri moduli attraverso le API per il C. Quasi dimenticavo il supporto per le SSL a 40 bit: e tutto questo è gratuito!. Per maggiori informazioni, visitate il sito di AOLserver presso `<http://www.aolserver.com/server/>`

Zeus Server

Zeus Server è stato sviluppatod da Zeus Technology. Affermano di aver prodotto il server più veloce, almeno stando ai risultati del benchmark WebSpec96. È possibile inoltre controllare e configurare l'applicazione da qualsiasi browser, nonché limitare l'utilizzo del processore o della memoria da parte dei programmi CGI ed eseguirli in un contesto sicuro (qualunque cosa questo voglia dire...). Supporta infine un numero illimitato di server virtuali. Il prezzo per la versione standard è di $999, che diventano $1699 se si vuole l'SSL: la società comunque ha sede fuori dagli Stati Uniti e dunque la versione a 128 bit è disponibile in ogni parte del mondo. Per informazioni, visitate `<http://www.zeus.co.uk>`. Il sito americano è a `<http://www.zeus.com>`. Vi devo avvisare che, pur essendo molto convinti delle loro prestazioni in termini di velocità, non sono elencati fra i primi dieci server nei Netcraft Surveys.

CL-HTTP

CL-HTTP significa Common Lisp Hypermedia Server. Se siete dei programmatori in Lisp, questo è il server che fa per voi, visto che potete scrivere i vostri CGI in questo linguaggio. Questo server funziona con un setup dal web e supporta tutte le funzionalità standard di un server. CL-HTTP è gratuito e i sorgenti sono distribuiti pubblicamente. Il sito web è `<http://www.ai.mit.edu/projects/iiip/doc/cl-http/home-page.html>` (non si poteva fare un url un pò più lunga?)

Se avete degli scopi commerciali (sito web aziendale, ISP) vi raccomando fortemente Apache. Se invece avete bisogno di un setup facile facile a discapito delle funzionalità più avanzate, allora Zeus Server fa per voi: ho anche sentito dire che è facile configurare il server Netscape. Se le vostre esigenze sono per un uso prettamente "interno", allora potete godere di maggiore flessibilità. Comunque, a meno che voi siate alla ricerca di qualche cosa di specifico, vi suggerisco ancora di utilizzare uno dei tre che ho menzionato.

Questa è solo una lista parziale di tutti i server disponibili. Per un elenco più completo, visitate il sito Netcraft presso `<http://www.netcraft.com/survey/servers.html>` o Web Compare a `<http://webcompare.internet.com>`.

7 Apache

La versione corrente di Apache è la 1.2.4. La versione 1.3 è in fase beta. Il sito principale di Apache è a `<http://www.apache.org/>`. Un'altra buona fonte di informazioni è Apacheweek a `<http://www.apacheweek.com/>`. La documentazione Apache è abbastanza buona, perciò non mi addentrerò nei dettagli della configurazione: i doc sono infatti sia sul sito web che inclusi con i sorgenti (in formato HTML: ci sono anche dei file di testo ma quelli in HTML sono meglio). Inoltre, la documentazione dovrebbe migliorare appena prende il via l'Apache Documentation Project. Fino ad oggi gran parte dei manuali sono scritti dagli sviluppatori: non per criticarli, ma obiettivamente i loro testi sono un pò difficili da capire se non si padroneggia la terminologia.

7.1 Dove trovarlo

Apache è incluso nelle distribuzioni Red Hat, Slackware e OpenLinux: sebbene non siano magari le ultime versioni, sono comunque degli eseguibili molto affidabili. L'aspetto negativo, però, è che non si può cambiare la configurazione delle directory (che è totalmente diversa fra l'una e l'altra distribuzione nonché rispetto ai default Apache).

I codici sono disponibili presso il sito web Apache `<http://www.apache.org/dist/>`. I file binari si trovano nello stesso posto. È inoltre possibile scaricarli da SunSite a `<ftp://sunsite.unc.edu/pub/Linux/apps/www/servers/>`. Per chi ha Red Hat, l'RPM con i più recenti eseguibili è di solito nella directory contrib a `<ftp://ftp.redhat.com/pub/contrib/i386/>`.

Se il server sarà utilizzato per scopi commerciali, è molto meglio se vi scaricate i sorgenti dal sito Apache e lo compilate voi stessi. L'altra possibilità consiste nell'utilizzare gli eseguibili che vengono distribuiti con Slackware, Red Hat o OpenLinux, ma questi possono causare dei problemi di sicurezza: un eseguibile sconosciuto può avere delle backdoor per gli hacker, oppure una patch instabile potrebbe causare danni al sistema. Inoltre, scaricare i sorgenti vi dà più controllo sui moduli da compilare e sulle directory di default. Non è poì così difficile compilare Apache, e comunque sia non vi potete dire dei veri utilizzatori di Linux finché non compilerete da soli i vostri programmi ;)

7.2 Compilazione e installazione

Per prima cosa bisogna fare un untar dell'archivio in una directory temporanea. Poi, spostatevi nella directory src. Editate il file di configurazione per includere eventuali moduli speciali (la maggior parte dei più comuni è comunque già inclusa). Non c'è alcun bisogno di modificare le regole o i parametri del makefile per Linux. Eseguite poi lo script di configurazione (`./Configure`). Assicuratevi che dica "Linux platform" e che sia settato gcc come compilatore. È possibile editare adesso il file httpd.h per cambiare le directory di default. La directory home del server - dove vengono cioè mantenuti i file di configurazione - è settata (se non altrimenti specificato) a `/usr/local/etc/httpd/`, ma potreste volerla cambiare in un più semplice `/etc/httpd/`. La root del server (dove cioè vengono conservate le pagine HTML) è per default `/usr/local/etc/httpd/htdocs/`, ma io preferisco la directory `/home/httpd/html` (che tra l'altro è il default di Red Hat). Se avete intenzione di utilizzare su-exec (cfr. sotto, funzioni speciali) potrebbe essere utile cambiare anche quella directory. La root del server può essere cambiata anche nei file di configurazione, ma è una buona idea compilarcelo dentro nell'evenienza in cui Apache non sia in grado di trovare o leggere il file di configurazione: tutto il resto dovrebbe essere modificato nei file di config. Infine, lanciate make per compilare Apache.

Se avete problemi relativi alla mancanza di file include, assicuratevi delle seguenti cose: controllate di avere gli header del kernel (i file di include) installati per la vostra versione del kernel; assicuratevi inoltre di avere settati i seguenti link simbolici:

```
/usr/include/linux link a /usr/src/linux/include/linux
/usr/include/asm link a /usr/src/linux/include/asm
/usr/src/linux link alla directory del sorgenti di Linux (es. linux-2.0.30)
```

I links possono essere creati con `ln -s`, comando che funziona come cp con l'unica differenza che crea un link simbolico (`ln -s directory-sorgente link-di-destinazione`).

Quando make ha finito, ci dovrebbe essere un eseguibile chiamato http nella directory. Questo file deve essere spostato in una directory binaria: `/usr/sbin` o `/usr/local/sbin` potrebbero essere delle buone scelte.

Copiate poi le sub-directory di configurazione, di log e delle icone dai sorgenti alla directory home del server. Poi, bisogna rinominare 3 file nella sub-directory conf per liberarsi dell'estensione `-dist` (es `httpd.conf-dist` diventa `httpd.conf`).

Ci sono inoltre molti programmi di supporto inclusi con Apache. Si trovano nella directory `support` e devono essere copiati e installati separatamente. La maggior parte può essere creata usando il makefile in quella directory (che viene creato quando si lancia lo script principale `Configure`). Nessuno di questi programmi è indispensabile per usare Apache, ma alcuni facilitano il compito dell'amministratore.

7.3 Configurazione

Adesso dovreste avere quattro file nella sub-directory `conf` (sotto la directory home del server). `httpd.conf` configura il demone del server (numero di porta, utente, ecc.). Il `srm.conf` imposta la struttura di base dei documenti, gli handler speciali, ecc. Il `access.conf` si occupa invece della configurazione di base per l'accesso. Infine `mime.types` comunica al server quale tipo MIME inviare al browser a seconda delle diverse estensioni.

I file di configurazione hanno comunque una documentazione interna molto buona (sono pieni di commenti), sempre che si capisca il gergo. È consigliabile leggerli tutti almeno una volta prima di mettere su il server. Ogni argomento sulla configurazione è coperto esaurientemente anche nella documentazione di Apache.

Il file `mime.types` non è esattamente un file di configurazione. È infatti utilizzato dal server per tradurre le estensioni dei file in tipi MIME da inviare al browser e contiene già la maggior parte dei più comuni tipi di file cosicché poche persone hanno bisogno di modificarlo. Inoltre, col passare del tempo sempre nuovi tipi MIME saranno aggiunti e dunque la cosa migliore è scaricarsi un nuovo file (e magari anche una nuova versione del server).

Ricordatevi sempre, quando cambiate i file di configurazione, di riavviare Apache o di inviargli il segnale SIGHUP con `kill` per fare in modo che le modifiche abbiano effetto. Assicuratevi comunque di mandare il segnale al processo padre e non ai figli: il padre ha di solito l'indicatore di processo più basso, ricavabile anche dal file `httpd.pid` nella directory di log. Se accidentalmente vi capita di inviare il segnale ad un processo figlio, questo verrà terminato e il processo padre lo riavvierà automaticamente.

Non percorrerò qui tutti i passaggi nella configurazione di Apache: ho deciso invece di occuparmi di argomenti specifici, scelte da compiere e peculiarità del server.

Raccomando comunque a tutti gli utenti di leggere i suggerimenti sulla sicurezza nella documentazione Apache, disponibile anche sul sito Apache a `<http://www.apache.org/docs/mics/security_tips.html>`.

7.4 Ospitare siti web virtuali

Il Virtual Hosting si ha quando un solo computer ha più di un dominio. Il vecchio metodo consisteva nell'assegnare ad ogni host virtuale un differente indirizzo IP: col nuovo metodo, invece, si usa un solo indirizzo IP ma bisogna utilizzare un browser che supporti l'HTTP 1.1.

Il mio suggerimento per le applicazioni commerciali è di utilizzare ancora il vecchio metodo (più indirizzi IP) finché la maggior parte delle persone disporranno di un browser compatibile con lo standard HTTP 1.1 (basta aspettare un annetto o due): in questo modo, inoltre, si avrà una più completa illusione di virtual hosting. Non bisogna dimenticare poi che, mentre tutti e due i metodi consentono la posta virtuale (qualcuno me lo può confermare?), solo il virtual hosting basato su differenti indirizzi IP consente l'FTP virtuale.

Se si tratta invece di un server per un club o per una pagina personale, è forse meglio considerare l'hosting basato sugli IP condivisi: è più economico e si risparmia del prezioso spazio IP.

È poi possibile realizzare un mix di IP condiviso e non sullo stesso server: per informazioni, visitate Apacheweek a `<http://www.apacheweek.com/features/vhost>`.

7.4.1 Hosting basato sugli IP virtuali

Con questo metodo ogni host virtuale ha il suo indirizzo IP: determinando l'indirizzo a cui è stata inviata la richiesta, Apache e gli altri programmi possono decidere quale dominio servire. Questo è comunque uno spreco incredibile di spazio. Prendete per esempio il server su cui si trova il mio dominio virtuale: ci sono circa 35.000 account, e cioè 35.000 indirizzi. Eppure, è probabile che i server effettivamente funzionanti non siano nemmeno una cinquantina.

Per installare e far funzionare questo metodo si procede in due fasi distinte: prima si dice a Linux di accedere a più di un indirizzo IP e poi si setta Apache per servire gli host virtuali.

Il primo passo per configurare Linux a supportare più indirizzi IP è ricreare il kernel, operazione che funziona meglio con i kernel della serie 2.0 (o più recenti): è necessario includere il networking IP nonché il supporto dell'aliasing IP. Per informazioni sulla compilazione del kernel, date un'occhiata al *Kernel HowTo* `<http://sunsite.unc.edu/LDP/HOWTO/Kernel-HOWTO.html>`.

La fase successiva è la configurazione di ogni interfaccia al boot: se state usando la distribuzione Red Hat potete farlo dal Control Panel avviando X-Windows come utente di root, cliccando sulla configurazione della rete nel pannello di controllo e specificando nella scheda delle interfacce la propria scheda di rete. È poi necessario cliccare su Alias (dovrebbe trovarsi al fondo dello schermo) e completare le informazioni: bisogna compiere queste informazioni per ogni host virtuale/indirizzo IP.

Se utilizzate invece delle altre distribuzioni, è probabile che dobbiate farlo a mano. Potete semplicemente mettere i comandi nel file `rc.local` che si trova in `/etc/rc.d` (in realtà dovrebbe essere tutto nelle cose collegate alla rete). Dovreste avere un comando `ifconfig` e `route` per ogni dispositivo. Viene così assegnato un sub-dispositivo ad ogni indirizzo alias: per esempio eth0 avrebbe gli alias eth0:0, eth0:1 ecc. Ecco un esempio di tutta la faccenda:

```
ifconfig eth0:0 192.168.1.57
route add -host 192.168.1.57 dev eth0:0
```

È anche possibile aggiungere un indirizzo broadcast e una netmask al comando ifconfig: se avete un sacco di alias, si potrebbe creare un ciclo for per facilitare le cose. Leggete comunque l'*IP alias mini howto* `<http://sunsite.unc.edu/LDP/HOWTO/mini/IP-Alias.html>`.

In seguito bisogna settare il domain name server (DNS) per coprire questi nuovi domini: se poi non avete ancora un dominio, allora vi dovete rivolgere alla *Internic* `<http://www.internic.net>` per registrarlo. Date comunque un'occhiata al DNS-HOWTO per creare il vostro DNS.

Infine, è necessario configurare Apache per fargli gestire correttamente i domini virtuali: per far ciò è necessario modificare il file `httpd.conf` verso la fine di cui viene comunque fornito un file di esempio. Tutti i comandi specifici per quell'host virtuale sono posti fra la tag di direttiva `virtualhost`, dove è possibile inserire quasi tutti i comandi. Di solito si setta una diversa directory radice per i documenti, una directory per gli script e per i file di log. Si possono aggiungere infiniti host virtuali semplicemente aggiungendo più tag `virtualhost`.

In rari casi potreste aver bisogno di lanciare server separati per specifici host: questo è necessario quando una direttiva è necessaria per un host virtuale ma non è fra quelle ammesse fra le tag per l'hosting virtuale. In questo caso si deve utilizzare la direttiva bindaddress: ogni server avrà un nome e dei file di configurazione diversi, ed ogni server risponderà ad un indirizzo IP, specificato dalla direttiva bindaddress. Questo è comunque un incredibile spreco di risorse.

7.4.2 Hosting basato sugli indirizzi IP condivisi

Si tratta di un nuovo metodo per l'hosting virtuale: viene utilizzato un solo indirizzo IP, riservandolo solo alle macchine reali (non a quelle virtuali). Nell'esempio precedente, dei 35.000 host virtuali solo 50 avrebbero un indirizzo IP (uno per ogni macchina). Tutto ciò è possibile sfruttando le caratteristiche dell'HTTP 1.1: è il browser che dice al server quale sito vuole. I problemi sorgono con quei browser che non supportano l'HTTP 1.1, perché questi leggeranno la pagina principale del server, che può comunque offrire una lista dei server virtuali a disposizione. Certo, questo rovina un po' l'intera illusione del hosting virtuale, e cioè quella di avere un proprio server.

Il settaggio è molto più semplice che nell'hosting basato sui diversi indirizzi IP. È sempre necessario avere un proprio dominio dalla Internic nonché configurare il proprio DNS, e Apache è settato nella stessa maniera di prima. Dal momento però che si usa sempre lo stesso indirizzo IP nella tag virtualhost, il server si accorge automaticamente che si sta utilizzando l'hosting virtuale basato sull'IP condiviso.

Ci sono molte soluzioni per i vecchi browser, e io adesso spiegherò quella che secondo me è la migliore. Per prima cosa è necessario trasformare la pagina principale in host virtuale (condiviso o no): in questo modo si libera la main page così da poterla utilizzare per una lista di link a tutti i virtual host presenti. Poi bisogna creare una backdoor per i vecchi browser utilizzando la direttiva `ServerPath` per ogni host virtuale all'interno della direttiva `virtualhost`. Per esempio, aggiungendo `ServerPath /mysite/` a www.mysite.com i vecchi browser saranno in grado di accedere al sito attraverso www.mysite.com/mysite/. Infine si mette la pagina di default sul sito principale che dica gentilmente di scaricarsi un nuovo browser e che punti a tutte le backdoor per i siti ospitati sul server: quando un vecchio browser accede al sito, saranno automaticamente inviati alla pagina principale e riceveranno un menu con tutti i link disponibili. I nuovi browser, invece, non vedranno mai la pagina principale e andranno direttamente agli host virtuali. Bisogna ricordare però che si devono mantenere i link relativi al proprio sito web dal momento che le pagine saranno accessibili da due URL differenti (www.mysite.com e www.mysite.com/mysite/).

Spero di non avervi confuso troppo, ma non è comunque una soluzione facile. Magari potreste considerare l'hosting virtuale basato sugli indirizzi IP. Comunque un trucchetto molto simile è spiegato sul sito Apache a `<http://www.apache.org/manual/host.html>`.

Se mai qualcuno avesse una interessante fonte di informazioni per l'hosting basato sugli IP condivisi, mi farbbe piacere saperlo. Sarebbe anche simpatico sapere quanti e quali browser supportano l'HTTP 1.1.

7.5 Script CGI

Ci sono due metodi diversi per consentire ai propri utenti di utilizzare script CGI: la prima è di considerare tutto ciò che finisce con `.cgi` uno script CGI, la seconda è quella di creare una directory per gli script (generalmente `cgi-bin`). Si possono anche utilizzare entrambi i metodi: in tutti e due i casi, comunque, gli script devono poter essere eseguiti da tutti (`chmod 711`). Nel concedere accesso agli script si apre una grande falla nella sicurezza: state bene attenti, drizzate le orecchie e cercate di minimizzare i rischi.

Personalmente preferisco il primo metodo, specialmente quando gli script sono complessi: questo metodo consente infatti di mettere gli script in ogni directory, e a me piace metterli dove risiedono anche le pagine web che li utilizzano. Per i siti con un sacco di script, è molto più elegante che avere una directory piena zeppa di script. Questo metodo è poi molto facile da settare: per prima cosa, bisogna togliere il commento all'handler `.cgi` alla fine del file `srm.conf`. Assicuratevi poi che tutte le vostre directory abbiano l'opzione `option ExecCGI` oppure `All` nel file `access.conf`, e il gioco è fatto.

Creare una directory apposita per gli script è però considerato più sicuro. Per farlo bisogna utilizzare la direttiva ScriptAlias nel file `srm.conf`, in cui il primo argomento è l'Alias ed il secondo la directory che sarà utilizzata quando qualcuno chiederà della directory `/cgi-bin/`: per esempio `ScriptAlias /cgi-bin/ /usr/httpd/cgi-bin/` abilita la directory `/usr/httpd/cgi-bin` ad eseguire degli script. Per motivi di sicurezza sarebbe poi meglio cambiare anche le proprietà delle directory togliendo i commenti alle linee `Options none, AllowOveride none` nel file `access.conf` fornito come esempio. Inoltre, non bisogna creare le directory per gli script come sub-directory delle cartelle che contengono la pagina web: per esempio, se la directory con le pagine è `/home/httpd/html/`, non mettete gli script in `/home/httpd/html/cgi-bin`, bensì in `/home/httpd/cgi-bin`.

Se volete consentire ai vostri utenti di avere le loro directory di script, potete usare più comandi `ScriptAlias`, che dovrebbero essere all'interno della direttiva `virtualhost` per ogni host virtuale. Se c'è qualcuno che conosce un metodo più semplice per permettere a tutti gli utenti di avere una propria directory di script senza ricorrere ogni volta al comando ScriptAlias me lo faccia sapere.

7.6 Directory Web degli Utenti

Ci sono due metodi diversi per trattare le directory web degli utenti. Si può avere una sub-directory nella directory degli utenti (che di solito è `public_html`), oppure una directory specifica per le directory web. Con entrambi i metodi bisogna assicurarsi comunque che queste directory siano settate in lettura e scrittura nel file `access.conf`.

Il primo metodo è quello di default di Apache: quando arriva una richiesta per `/~bob/`, il server va a cercare la directory `public_html` nella cartella di bob. È possibile cambiare la directory con la direttiva `UserDir` nel file `srm.conf`: è comunque necessaria settarla in lettura ed esecuzione per tutti. Come si può vedere, questo metodo crea un potenziale rischio di sicurezza perché Apache può accedere alla directory solo se la directory home degli utenti è eseguibile per tutti.

Il secondo metodo è molto facile da settare: bisogna cambiare la direttiva `UserDir` nel file `srm.conf`. Questa direttiva può assumere svariati formati e quindi è una buona idea dare un'occhiata alla documentazione Apache per chiarimenti. Comunque, se volete che ogni utente abbia la propria directory in `/home/httpd/`, dovete usare `UserDir /home/httpd`. In questo caso, quando viene richiesta `/~bob/`, viene tradotta in `/home/httpd/bob/`. Se poi volete una sub-directory nella cartella di bob, allora dovrete usare `UserDir /home/httpd/*/html`: questo infatti porta il server a tradurre in `/home/httpd/bob/html/` e vi consente anche di avere una directory per gli script (ad esempio `/home/httpd/bob/cgi-bin/`).

7.7 Daemon mode vs. Inetd mode

Apache può essere eseguito in due modalità: come demone ("daemon") sempre in funzione (modalità che Apache chiama 'standalone') oppure lanciato dal super server inetd.

La modalità daemon è di gran lunga migliore rispetto a quella con inetd, ed è anche la modalità predefinita. L'unico caso in cui si potrebbe voler utilizzare quest'ultimo modo è in presenza di applicazioni poco frequenti: il test interno di alcuni script, l'Intranet di una piccola azienda. Il modo Inetd salva infatti memoria perché il server è caricato solo quando effettivamente richiesto, e solo il demone inetd risiede permanentemente in memoria.

Se non utilizzate Apache molto spesso, potreste utilizzarlo nella modalità daemon e lanciarlo solo quando vi serve, per poi ucciderlo una volta finito (assicuratevi di terminare il processo padre, e non uno dei figli).

Per impostare la modalità inetd è necessario modificare un po' di file. Per prima cosa, in `/etc/services` vedete se http è già presente: se non lo è, aggiungetelo:

```
http    80/tcp
```

Il posto migliore è dopo 79 (finger). Poi dovete modificare il file `/etc/inetd.conf` e aggiungerci le linee per Apache

```
http    stream  tcp     nowait  root    /usr/sbin/httpd httpd
```

Assicuratevi di cambiare il path se avete Apache in un altro posto: il secondo httpd non è un errore, dal momento che il demone inet lo richiede. Se non utilizzate questo demone, potreste voler togliere i commenti al resto delle linee nel file così da evitare l'attivazione di altri servizi (FTP, finger, telnet e molte altre cose che di solito funzionano con questo demone).

Se già avete in esecuzione il demone inet (`inetd`), allora dovete solo inviargli il segnale SIGHUP (con kill; consultate la documentazione su kill per informazioni) o riavviare il computer perché le modifiche abbiano effetto: se invece non è già in esecuzione, dovete avviare il demone manualmente o metterlo nei file di inizializzazione così da caricarlo all'avvio (la scelta migliore è il file `rc.local`).

7.8 Autorizzare i comandi put e delete

I nuovi strumenti di authoring HTML supportano un nuovo metodo per inviare le pagine web al server utilizzando l'http invece dell'FTP: alcuni di questi prodotti nemmeno lo supportano più l'FTP. Apache è in grado di supportare questo metodo, ma manca lo script per gestire la richiesta: tale script, però, potrebbe aprire un serio buco nella sicurezza e perciò riflettete bene prima di scriverne o installarne uno.

Se qualcuno è a conoscenza di uno script che funzioni bene me lo dica, e lo aggiungerò qui.

Per informazioni, consultate Apacheweek `<http://www.apacheweek.com/features/put>`.

7.9 Autenticazione dell'utente/Controllo d'accesso

Questa è una delle caratteristiche che preferisco: consente infatti di proteggere una directory o un file senza dover usare degli script CGI nonché consentire o negare l'accesso in base all'indirizzo IP o al dominio del client. Si tratta di un grande servizio per tenere lontani rompiscatole dalla message board o dal guest book (si ricava il loro indirizzo IP o il dominio dai file di log).

Per attivare l'autenticazione dell'utente la directory deve avere `AllowOverrides AuthConfig` settato nel file `access.conf`. Per consentire poi il controllo d'accesso (tramite dominio o indirizzo IP) bisogna poi impostare nella directory anche `AllowOverrides Limit`.

Per configurare la directory è necessario mettervi un file `.htaccess` Per l'autenticazione dell'utente si usa invece di solito un file `.htpasswd` oppure `.htgroup`: questi file possono essere condivisi tra vari file `.htaccess`.

Per ragioni di sicurezza raccomando vivamente a tutti di utilizzare le seguenti direttive nel file access.conf:

```
<files ~ "/\.ht">
order deny,allow
deny from all
</files>
```

Se non siete ammnistratori del sistema, potete anche mettere nel vostro file .htaccess se AllowOverride Limit è impostato per la vostra directory. Questa direttiva impedirà agli altri di andare a curiosare dentro i vostri file per il controllo dell'accesso (.htaccess, .htpasswd, etc).

Ci sono svariate opzioni e tipi di file che possono essere utilizzati col controllo dell'accesso: non è quindi negli scopi di questo documento descriverli. Per informazioni su come impostare la User Authentication date un'occhiata ad Apacheweek a `<http://www.apacheweek.com/features/userauth>` o le pagine NCSA a `<http://hoohoo.ncsa.uiuc.edu/docs-1.5/tutorials/user.html>`.

7.10 su-exec

La funzione su-exec esegue degli script CGI come se l'esecutore ne fosse anche il proprietario: normalmente viene eseguito come l'utente del web server (di solito 'nobody'). In questa maniera si consente agli utenti di accedere ai propri script CGI senza doverli rendere leggibili a tutti (cosa che sarebbe un rischio per la sicurezza). Se non si sta attenti, è però possibile aprire un buco più grande di quello che si vorrebbe coprire dal momento che su-exec esegue dei controlli prima di eseguire gli script, ma se questi controlli non sono impostati bene sono problemi.

Utilizzare su-exec non è un mestiere per principianti: se non sapete quello che state facendo, lasciate perdere. Potreste finire per consentire ai vostri utenti di acquisire accesso di root al vostro sistema, e non è quindi conveniente scherzarci. La codifica ed l'impostazione su-exec è difficile apposta, così che i principianti se ne tengano alla larga (tutto deve essere fatto a mano, niente file di make, niente script di installazione).

Il codice per su-exec si trova nella directory `support` dei sorgenti. Dapprima bisogna modificare il file `suexec.h` per il proprio sistema: poi bisogna compilare il codice per su-exec con il comando

```
gcc suexec.c -o suexec
```

In seguito si deve copiare l'eseguibile su-exec nella directory giusta, che Apache preimposta a `/usr/local/etc/httpd/sbin/`, ma che può comunque essere cambiata modificando `httpd.h` nella directory dei sorgenti e ricompilando Apache. Ricordatevi che Apache guarderà solo in questa directory, e non nel path. È poi necessario cambiare il proprietario del file e settarlo a root (`chown root suexec`) e settare il bit suid (`chmod 4711 suexec`). Infine, riavviate Apache e dovrebbe apparire sullo schermo che si sta utilizzando su-exec.

Gli script CGI dovrebbero essere in lettura per tutti come sempre: saranno eseguiti automaticamente da ogni utente come il proprietario del file. Se settate il bit SUID (set user id) sul CGI questo non verrà eseguito, così come non vengono eseguiti gli script posseduti dagli utenti di sistema (root, bin, ecc.). Per ulteriori informazioni sulle condizioni di sicurezza che devono essere rispettate, controllate la documentazione di su-exec: se poi avete problemi potete anche spulciare il file di log, che si chiama `cgi.log`.

Su-exec non funziona se lanciate Apache da inetd: questo problema sarà risolto nelle prossime versioni dal momento che non ci sarà più un modo inetd. Se vi piace giocare col codice sorgente, potreste voler modificare il file http_main.c per liberarvi delle linee in cui Apache annuncia che sta utilizzando su-exec (lo stampa erroneamente prima di qualsiasi output).

Assicuratevi infine di leggere la documentazione Apache riguardante su-exec: è inclusa con i sorgenti ed è anche disponibile sul sito di Apache a `<http://www.apache.org/docs/suexec.html>`

7.11 Imagemap

Apache è in grado di gestire le imagemap dal lato del server: si tratta di immagini su una pagina web che portano l'utente in diverse locazioni a seconda del punto su cui si clicca. Per abilitarle è per prima cosa necessario assicurarsi che il modulo per le imagemap sia installato (è uno dei moduli di default). Poi, togliete i commenti all'handler `.map` alla fine del file `srm.conf`: così facendo, tutti i file con estensione `.map` saranno dei file per le imagemap, dove cioè vengono settati i diversi link a seconda del punto in cui si clicca. Apache utilizza i file di mappa seguento lo standard NCSA, di cui vi presento un esempio:

```
<a href="/map/mapfile.map">
<img src="picture.gif" ISMAP>
</a>
```

In questo esempio `mapfile.map` è il mapfile mentre `picture.gif` è l'immagine su cui cliccare.

Esistono molti programmi che possono generare dei mapfile compatibili con lo standard NCSA, ma è anche possibile crearseli da soli. Per una discussione più approfondita sull'argomento, andate a `<http://www.apacheweek.com/features/imagemaps>`.

7.12 SSI/XSSI

Il Server Side Includes (SSI) aggiunge contenuti dinamici - inseriti nella pagina - a documenti web altrimenti statici e non interattivi: il server, infatti, processa gli SSI e passa i risultati alla pagina. In questo modo, si possono aggiungere intestazioni e note di chiusura, la data di ultima modifica, eseguire comandi di sistema o degli script CGI. Con i nuovi eXtended Server Side Includes (XSSI) si può fare ancora di più, aggiungendo condizioni di controllo (if...else): è quasi come avere un linguaggio di programmazione.

Processare tutti i file HTML per cercare comandi SSI sarebbe una perdita di tempo ed uno spreco di risorse: pertanto, è necessario distinguere le pagine che contengono HTML normale da quelle con i comandi SSI. Generalmente si ricorre per questi file a delle estensioni diverse: la più usata è `.shtml`.

Per abilitare SSI/XSSI assicuratevi per prima cosa che il modulo per gli include sia installato: poi, modificate `srm.conf` e togliete i commenti alle direttive `AddType` e `AddHandler` per i file `.shtml`. Infine dovete settare `Options Includes` per tutte le directory dove volete eseguire file SSI/XSSI modificando il file `access.conf`. Fatto ciò, tutti i file con l'estensione `.shtml` saranno processati per controllare se contengono comandi SSI/XSSI.

Un altro modo per abilitare le includes è quello di usare la direttiva `XBitHack`: se abilitata, controlla che il file sia eseguibile dall'utente generico e se per quella directory è settata `Options Includes`: nel caso in cui si verifichino entrambe le condizioni, allora il file viene trattato come SSI. Questo procedimento funziona solo per file con il tipo MIME text/html `.html .htm`): comunque, non è consigliabile usare questo metodo.

Esiste poi un rischio di sicurezza nell'assicurare a file SSI di eseguire comandi di sistema e script CGI: è pertanto possibile bloccare questa funzione con la direttiva `Option IncludesNOEXEC` invece che `Option Includes` nel file `access.conf`. Tutti gli altri comandi SSI continueranno a funzionare.

Per maggiori informazioni, leggete la documentazione Apache su mod_includes, distribuita con i sorgenti e disponibile a `<http://www.apache.org/docs/mod/mod_include.html>`.

Per una discussione più dettagliata su SSI/XSSI, andate su Apacheweek a `<http://www.apacheweek.com/features/ssi>`.

Informazioni sui comandi SSI si trovano anche alla NCSA a `<http://hoohoo.ncsa.uiuc.edu/docs/tutorials/includes.html>`.

Mentre per l'XSSI date un'occhiata a `<ftp://pageplus.com/pub/hsf/xssi/xssi-1.1.html>`.

7.13 Sistema modulare

Apache può essere ampliato per supportare quasi tutto attraverso i moduli, di cui ne esiste già un vasto numero in giro per il web. Solo i moduli di interesse più generale sono Distribuiti con Apache, ma link a tutti i moduli esistenti si possono trovare all'Apache Module Registry a `<http://www.zyzzyva.com/module_registry/>`.

Per informazioni su come scrivere il proprio modulo andate a `<http://www.zyzzyva.com/module_registry/reference/>`

8 Web Server Add-ons

Ops, questa parte non l'ho ancora scritta...

Prossimamente: mSQL, PHP/FI, cgiwrap, Fast-cgi, estensioni MS frontpage, e altro ancora.

9 FAQ

A dire il vero non ci sono state domande frequenti - per adesso...

10 Letture consigliate

10.1 Libri della O'Reilly & Associates

Secondo me O'Reilly & Associates è la migliore casa per libri tecnici del pianeta. Si concentrano principalmente su Internet, Unix e programmazione e cominciano sempre con un bel po' di esempi fino a giungere ad argomenti molto avanzati: se si legge solo metà del libro si raggiunge già una conoscenza di tutto rispetto. In più, riescono ad aggiungere un po' di humor ad argomenti altrimenti un po' pallosi.

Hanno degli ottimi libri su HTML, PERL, Programmazione CGI, Java, Javascript, C/C++, Sendmail, Linux e tanto altro. Gli argomenti più "caldi" vengono aggiornati circa ogni 6 mesi e così è altamente consigliato visitare il loro sito web *O'Reilly & Associates* `<http://www.ora.com/>` o recarsi nel più vicino negozio di libri.

State attenti alle imitazioni: se non ha O'Reilly & Associates sulla copertina, probabilmente è stato scritto da qualcun'altro!

10.2 Internet Request For Comments (RFC)

- RFC1866 scritta da T. Berners-Lee e D. Connolly, "Hypertext Markup Language - 2.0", 11/03/1995
- RFC1867 scritta da E. Nebel e L. Masinter, "Form-based File Upload in HTML", 11/07/1995
- RFC1942 scritta da D. Raggett, "HTML Tables", 05/15/1996

- RFC1945 di T. Berners-Lee, R. Fielding, H. Nielsen, "Hypertext Transfer Protocol – HTTP/1.0", 05/17/1996.
- RFC1630 di T. Berners-Lee, "Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in the World-Wide Web", 06/09/1994
- RFC1959 di T. Howes, M. Smith, "An LDAP URL Format", 06/19/1996