Capítulo 7

El agente de transferencia de correo qmail

Introducción a los MTA

Características

Funcionamiento

Instalación y configuración

Webmail

Capítulo 7 El agente de transferencia de correo qmail

"La resistencia de una cadena la determina el eslabón más débil".

Un sistema seguro, como el que se está configurando en este proyecto, no puede flaquear por ningún lado. Por ello vamos a usar en él qmail como MTA.

Su facilidad de instalación y configuración harán que, en muy poco tiempo, podamos tener funcionando el correo de un gran número de personas.

Vamos a ayudar a su administración, junto a la del sistema en general, introduciendo las cuentas de los usuarios en el directorio OpenLDAP, además de que se autentiquen contra él.

7.1 Introducción a qmail

El correo electrónico se ha convertido en una forma de comunicación muy usada y a la vez muy necesaria para la empresa. Hoy en día es raro encontrar una organización que no ofrezca una cuenta de correo a sus empleados.

Para gestionar este correo existe un software específico conocido como MTA (agente de transferencia de correo). Entre los MTA más conocidos figuran Sendmail, Postfix o qmail.

Hasta hace relativamente pocos años Sendmail era el MTA de Unix, sin discusión. Fue el primero y por ello se había usado desde siempre. Debido a que fue creado en una época en la que la memoria era un bien escaso, no dispone de una fácil configuración y gestión.

Al principio (y cuando hablamos de principio nos referimos a la década de los 80) primaba el espacio. No importaba demasiado la claridad ni la facilidad de la sintaxis. Sendmail debía funcionar con las reducidas prestaciones de las máquinas de entonces. Además tantos años en máquinas en producción han hecho que Sendmail esté muy parcheado. Cada agujero de seguridad que surgía se solucionaba mediante un parche.

Cuando Sendmail fue diseñado, Internet era un lugar mucho menos hostil de lo que es ahora. No existía la necesidad de una seguridad alta. No se tuvo en cuenta, por tanto, en el diseño ni en la codificación.

Como afirman muchos autores, no se puede conseguir una seguridad verdadera sino se rediseña el sistema, algo que no se ha hecho (y probablemente no se haga) con Sendmail.

Como alternativa a Sendmail surgen distintos MTAs: Postfix, exim o qmail. La oferta es variada. Cada cual tiene sus pros y sus contras, aunque en uno de los aspectos más importantes, la seguridad, hay uno que sobresale. Ningún fallo, en cuanto a seguridad se refiere, desde 1998 (fecha de creación). Como anécdota podemos señalar que el creador de qmail, Dan Bernstein, ofreció en su día una recompensa para todo aquel que encontrara un agujero de seguridad. El premio sigue estando desierto.

7.2 Características de qmail

qmail como agente de transferencia de correo que es, proporciona entrega y retransmisión de correo local y remoto. Podemos enumerar algunas de sus principales características.

7.2.1 Características principales

- Rendimiento. qmail puede realizar entrega de correo en paralelo. De hecho es capaz de entregar hasta 20 mensajes simultáneamente. Este es además su valor por defecto.

- Fiabilidad. Una vez que qmail acepta un mensaje garantiza que no se perderá. Además soporta un nuevo formato de buzón de correo que funciona bien incluso sobre NFS sin bloqueo.

- Simplicidad. qmail es más pequeño que cualquier otro MTA de características equivalentes. Es fácil y rápido de instalar ya que no necesita que se tomen muchas decisiones al principio.

- Seguridad. Ya hemos hablado de ella en qmail. Es un factor primordial y como tal se le confiere tanta importancia como merece. Se consigue básicamente mediante una separación muy clara entre dirección, ficheros y programas. Se minimiza el código que debe ejecutarse bajo permiso de root y se controla minuciosamente.

7.2.2 Otras características

- Construcción de mensajes. Se adecua a las RFC 822 y RFC 1123. Compatibilidad por tanto con los agentes de usuario actuales.

- Soporte completo para grupos de direcciones. Presenta comandos de tipo Sendmail que permite mantener la compatibilidad con los agentes de usuario actuales.

- Servicio SMTP y POP3 de acuerdo con la RFC 821 y RFC 1939, respectivamente.

- Gestión de cola

- Enrutamiento por dominio

- Entrega locales

- Retransmisión y listas de correo

7.3 Funcionamiento y uso de qmail

No hace falta entender cómo funciona qmail para instalarlo o usarlo. Sin embargo, el conocimiento de su funcionamiento va a permitirnos llegar a dominar qmail y poder sacarle el mayor partido.

Afortunadamente, el diseño simple y modular de qmail va a convertir en fácil la dura tarea de comprender un sistema tan complejo como un MTA.

Una división posible de los módulos de qmail consiste en separarlos en dos grupos. Aquellos que aceptan nuevos mensajes y los colocan en la cola, y los que toman los mensajes de la cola y los entregan. Esto es, estamos separando en dos partes: recepción y entrega.

La separación es completa. Cualquiera de estas dos funciones puede trabajar con la otra apagada. Veamos a continuación la organización a un alto nivel conceptual de qmail.

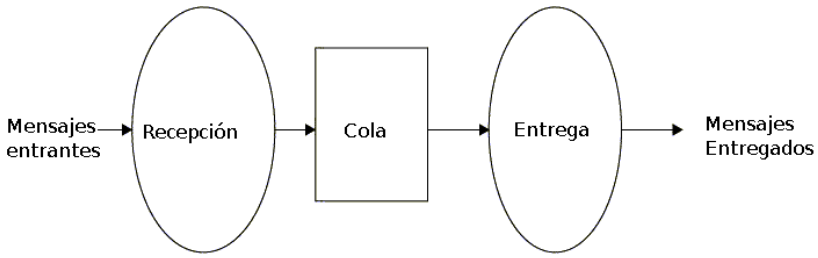

Figura 7.1 Esquema general qmail

7.3.1 Conceptos de recepción y entrega

7.3.1.1 Recepción

Los mensajes llegan a la cola por dos rutas principalmente.

- Inyección local: usando qmail-inject o sendmail

- Inyección desde red: usando qmail-smtpd, qmail-qmqpd o qmail-qmtpd.

Ambas rutas usan el módulo qmail-queue para inyectar los mensajes en la cola propiamente dicha. La siguiente figura muestra la organización de la función de recepción.

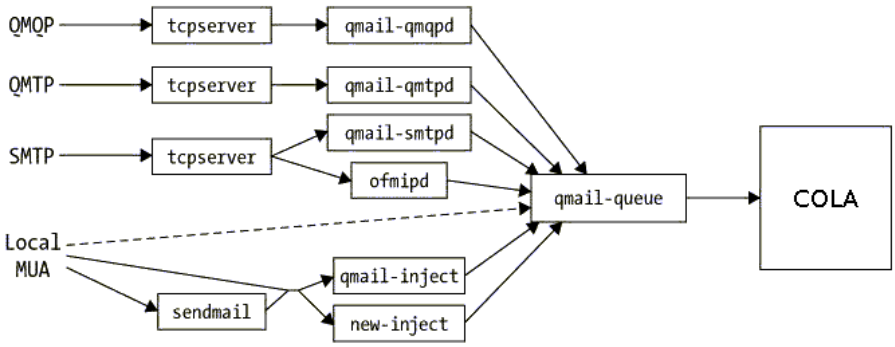

Figura 7.2 Recepción de mensajes.

7.3.1.2 Entrega

Los mensajes se entregan desde la cola siguiendo dos rutas:

- entrega local: usando para ello qmail-local, y,

- entrega remota: usando qmail-remote.

Ambas entregas se gestionan por qmail-send a través de qmail-lspawn y qmail-rspawn, respectivamente.

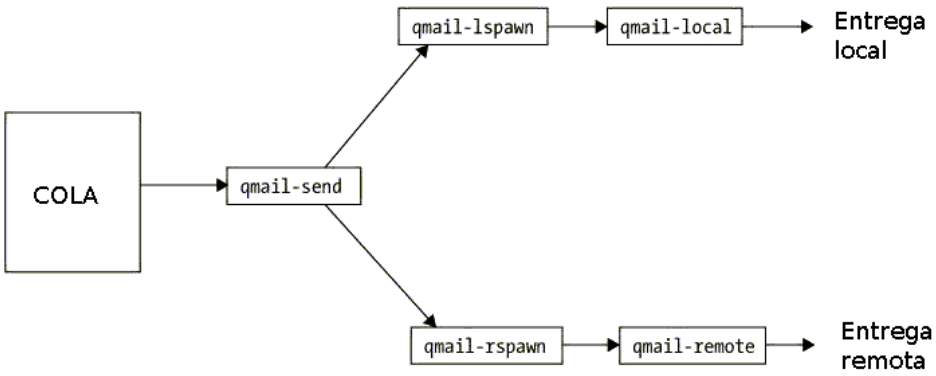

Figura 7.3 Entrega de mensajes

7.3.2 Módulos

7.3.2.1 Módulos de recepción

Sendmail

El comando Sendmail prácticamente no es más que un intermediario de qmail-inject. Acepta muchos de los argumentos y opciones de Sendmail, los traduce a los equivalentes en qmail-inject, ignora las opciones irrelevantes y finalmente ejecuta qmail-inject.

qmail-inject

La principal tarea de qmail-inject es la de asegurar que las cabeceras de los mensajes cumplen con la RFC 2822, antes de pasar a dichos mensajes hacia qmail-queue. Parte de su trabajo comprende:

- Comprobar direcciones. Para todas las direcciones que aparecen en los campos From, To, Cc y demás, qmail-inject asegura que están en formato usuario@FQDN.

- Si una dirección consiste en sólo la parte de usuario, añade @defaulthost. Con defaulthost nos referimos al fichero del mismo nombre que se encuentra en /var/qmail/control.
- Si una dirección consiste en sólo usuario@"nombre de máquina", añade a esta cadena .defaultdomain. Defaultdomain es otro de los ficheros de configuración que se encuentran en el directorio control.
- Si una dirección aparece como usuario@"nombre de máquina"+, qmail-inject reemplazaría el signo + por .plusdomain, otro de los ficheros del directorio control.

- comprobar destinatarios. Si no se especifican destinatarios por línea de comandos, qmail-inject los busca en los campos To, Cc, Bcc, Apparently-To, Resent-To, Resent-Cc y Resent-Bcc. Por supuesto, todos los cambos Bcc y Resent-Bcc se eliminan de las cabeceras.

El cumplimiento de la RFC 2822 requiere que todos los mensajes tengan un campo To o Cc. qmail-inject añade, si es necesario, un campo que contiene lo siguiente: Cc: recipient list not shown:; que es un grupo de direcciones vacío.

- campos requeridos. qmail-inject añade los siguientes campos, sino los encuentra.

– From: nombre del usuario que llamó a qmail-inject.
– Date: hora actual GMT.
– Message-id. En realidad este campo no debe aparecer necesariamente, pero es muy útil para seguirle el rastro a los mensajes. El valor de este campo es <timestamp.pid.qmail@FQDN>. El FQDN se suele construir en este caso mediante los archivos de configuración defaulthost y defaultdomain.

- otras características. qmail-inject también realiza lo siguiente:

– Las direcciones que aparecen listadas en los campos de la cabecera deben estar separadas por comas. Si qmail-inject encuentra que están separadas por espacios, inserta automáticamente comas entre ellas. Por ejemplo To: jose esperanza, será reescrito como To: jose, esperanza.
– Los campos Return-path y Content-length se eliminan.

qmail-smtpd

Los mensajes recibidos remotamente llegan a través de tres posibles módulos qmail-smtpd, qmail-qmtpd y qmail-qmpd, dependiendo del protocolo usado. En este caso estamos usando el protocolo SMTP.

qmail-smtpd lleva a cabo una sesión SMTP, acepta uno o más mensajes, y los pasa al módulo qmail-queue. qmail-smtpd no maneja las conexiones de red entrantes, sino que deben ser aceptadas por un servidor de red, tal como es tcpserver, xinted o inetd.

qmail-qmtpd

Este módulo realiza prácticamente las mismas funciones que qmail-smtpd. La diferencia está en que utiliza el protocolo QMTP en lugar de SMTP, lo que cambia las peticiones y respuestas.

qmail-qmpd

El comportamiento de qmail-qmpd es muy parecido al de qmail-qmtpd, excepto en que no realiza ningún control en la retransmisión. Se aceptan incondicionalmente todos los destinatarios.

qmail-queue

La tarea de este módulo es la de aceptar mensajes y colocarlos en la cola. Espera recibir las direcciones de los destinatarios perfectamente especificada. Además añade una línea al mensaje tal como mostramos:

```
Received: (qmail 17090 invoked from network); 10 Nov 2004 15:02:00 -0000
```

La descripción de los campos es sencilla. El número 17090 es el identificador de proceso de qmail-queue. La fecha hace referencia al momento en el que qmail-queue procesó el mensaje. Por último el método de invocación fue desde red (invoked by network), aunque también puede ser por el usuario alias (invoked by alias) o para una devolución (invoked for bounce).

Para garantizar la fiabilidad de la recepción, la colocación de los mensajes en la cola se realiza en cuatro fases.

1) Se crea un fichero en el directorio /var/qmail/queue/pid tras el proceso de identificación de qmail-queue. El sistema de archivos asigna un número de nodo-i, y por tanto, garantiza que este número es único en todo el sistema de archivos. Se trata del identificador de proceso de qmail-queue para el mensaje.
2) El archivo pid/'pid' se renombra a mess/'split'/'nodo-i' y se escribe el mensaje a ese archivo. split es un número resultado de aplicar la función módulo a 'nodo-i' entre el parámetro de configuración conf-split.
3) Se crea el fichero intd/'nodo-i' y se escribe en él el envoltorio del mensaje.
4) Se crea un enlace al fichero anterior en todo/'nodo-i'

Justo en el momento en el que se produce en enlace anterior, el mensaje se introduce en la cola. Además escribe un byte en el fichero lock/trigger, que es una tubería que qmail-send examina. Cuando trigger contiene datos válidos, qmail-send comienza, limpia la tubería y escanea el directorio todo en busca de mensajes que enviar.

qmail-queue provoca su apagado si en el transcurso de 24 horas no ha logrado introducir el mensaje en la cola.

7.3.2.2 Módulos de entrega

qmail-send

qmail-send es el corazón de qmail. Procesa los mensajes de la cola y los entrega a los módulos qmail-lspawn y qmail-rspawn. Veamos las funciones de este módulo en orden desde que recoje un mensaje de la cola: preprocesado, entrega y limpieza.

- **preprocesado**. Al igual que en qmail-queue se realiza en varias fases.

1) Tras encontrar el fichero todo/'nodo-i', qmail-send borra los ficheros info/'split'/'nodo-i', local/'split'/'nodo-i' y remote/'split'/'nodo-i', si existen.
2) Se crea un nuevo fichero info/'split'/'nodo-i' que contiene la dirección del que envía.
3) Si el mensaje tiene remitentes locales, se añaden a local/'split'/'nodo-i'
4) Si el mensaje tiene remitentes remotos, éstos se añaden a remote/'split'/'nodo-i'.
5) Se borran intd/'nodo-i' y todo/'nodo-i'.

Tras la última fase el mensaje se considera preprocesado.

Los remitentes se consideran locales si sus dominios se encuentran listados en el fichero control/locals o control/virtualdomains. Si el remitente es virtual, la parte local se reescribe según se especifica en los dominios virtuales.

- **entrega**. Inicialmente todos los remitentes de local/'split'/'nodo-i' y remote/'split'/'nodo-i' se marcan como no hechos. Es qmail-send el que se encarga de entregarlos. Manda peticiones a qmail-lspawn y qmail-rspawn y cuando estos responden indicando entrega satisfatoria y error permanente, marca los mensajes como hechos para así no intentar entregarlos de nuevo.

Cuando todos los remitentes se han marcado como hechos, los ficheros usados se eliminan. Cuando un mensaje no se puede entregar se reintenta. El tiempo entre reintentos se determina mediante una fórmula cuadrática.

- **limpieza**. En esta fase de procesan los mensajes que hayan sido rechazados y se borran los ficheros usados de los directorios info y mess.

Los mensajes que se encuentran parcialmente en la cola a causa de un cuelgue del sistema, se pueden borrar de forma segura gracias a la política de fases que hemos llevado hasta el momento.

qmail-lspawn

Este módulo lee los comandos de qmail-send, llama a qmail-local para realizar las entregas, e informa a qmail-send de los resultados de la operación.

Antes de invocar a qmail-local, qmail-lspawn determina qué usuario local dispone de esa dirección, para así lanzar qmail-local con la identificación de usuario y grupo necesaria.

qmail-local

qmail-local acepta por la entrada estándar un mensaje, cuyos argumentos son: información de cabecera, localización para la entrega e instrucciones de entrega por defecto.

Antes de intentar una entrega, construye un campo, Delivered-To, basado en la información de entrega. Comprueba que no existe ningún campo llamado de la misma forma, y si lo hay, rechaza el mensaje y así previene de bucles.

qmail-rspawn

Este módulo lee las peticiones de entrega de qmail-send, invoca a qmail-remote para realizar la entrega de mensajes, e informa de los resultados a qmail-send.

qmail-remote

qmail-remote acepta un mensaje por la entrada estándar e información de cabecera como argumento. Tras intentar entregar el mensaje

remotamente vía SMTP, utiliza la salida estándar para informar de los resultados de sus operaciones.

La máquina remota se especifica también como un argumento. Puede ser tanto un FQDN como una dirección IP. Si se trata de un nombre de dominio, qmail-remote lo comprueba en el DNS y busca una entrada de tipo MX (Mail exchanger). Si la máquina remota se encuentra listada en el fichero control/smtproutes, qmail-remote usa la máquina que se especifica en este archivo.

7.4 Instalación y configuración

7.4.1 Obtener el software

El primer paso es obtener el software de la página oficial http://cr.yp.to. En esta podemos encontrar tanto el archivo qmail-1.03.tar.gz como los paquetes daemontools y uscpi-tcp, recomendados para la instalación.

En un primer momento vamos a usar el paquete netqmail-1.05 (descargado de www.qmail.org), que incluye además de los paquetes que hemos nombrado anteriormente, una serie de parches y de scripts. Los parches solucionan problemas surgidos a lo largo de estos años. Es muy conocido el parche que corrige el problema de definición de algunas variables en el fichero de cabeceras errno.h.

Los scripts nos ayudan a automatizar ciertos pasos, algunos de ellos muy tediosos, tanto en la instalación como en la configuración de qmail.

Tras esta instalación y mediante una configuración básica comprobaremos el correcto funcionamiento de qmail (cuyo desarrollo y conclusión se presentará en el capítulo de fase de pruebas). En esta instalación los usuarios de qmail se encuentran en /etc/passwd.

Posteriormente, mediante la configuración avanzada, en la que también se aplicarán diversos parches, integraremos los usuarios de qmail dentro del directorio OpenLDAP.

7.4.2 Requerimientos del sistema

qmail no presenta muchos problemas de instalación; es posible usarlo en la mayoría de distribuciones de Linux, aunque deben cumplir algunos requisitos.

Hardware

- Se necesita al menos 10 MB de espacio en disco para construir los ejecutables y unos 3 MB para binarios, documentación y ficheros de configuración.
- Un sistema de archivos "sano" para la cola de mensajes. La garantía de fiabilidad de qmail requiere que la cola resida en un sistema de archivos con semántica BSD FFS, aunque puede funcionar con cualquier otro. En nuestro caso usaremos ext3. Además debe disponer de suficiente espacio para alberga la cola.
- Conectividad adecuada a la red. qmail fue diseñado para sistemas "bien conectados". No sería muy conveniente usarlo como servidor para listas de correo de un equipo conectado mediante una línea de 28,8 Kbps. Hoy en día esto es no es problema.
- Acceso a un servidor DNS. Si qmail no contase con este servidor sólo podría mandar correo a sistemas remotos especificados explícitamente en el fichero de configuración smtproutes.

Software

- Compilador de C. Al igual que es resto de paquete que hemos instalado desde fuente, necesitamos un compilador de C. Como siempre contamos con GCC.
- Paquetes ucspi-tcp y daemontools. ucspi-tcp contiene el superservidor tcpserver. Sustituye a inetd, ya que el autor lo considera inseguro e incapaz de soportar cargas altas de trabajo. Es opcional, pero las recomendaciones de uso hacen que se instale casi siempre. Nosotros vamos a usarlo.

  daemontools es un conjunto de herramientas para gestionar los servicios Unix. En el caso de qmail gestiona tanto su funcionamiento como la información de logs. Los scripts que se proporcionan junto con Netqmail hacen uso de estas herramientas, así que su uso ahorrará bastante trabajo.
- Parches. Como ya hemos comentado se utilizarán diversos parches. Sin

ellos qmail no funcionaría. En un primer momento utilizaremos los parches que se distribuyen con Netqmail y para la integración en OpenLDAP usaremos el parche qmail-ldap descargado de www.nrg4u.com.

7.4.3 Pasos a seguir

Para instalar qmail vamos a partir de un sistema sin MTA, que aunque no es indispensable, va a facilitar mucho las cosas.

Por defecto en Red Hat se instala Sendmail, así que el primer paso será desinstalarlo.

```
rpm –e sendmail --no-deps
```

Usamos la opción --no-deps ya que seguramente estén instalados algunos clientes de correo que tengan a Sendmail como dependencia. No tenemos porqué desinstalarlos. De hecho incluso podemos utilizarlos, usando para ello las características de compatibilidad que presenta qmail con los clientes de Sendmail.

Para comenzar la instalación sólo debemos ejecutar el script qmail.sh como root en el directorio adecuado. Para ello:

```
su
./qmail.sh
```

No mostramos aquí el script debido a su extensión. Podemos encontrarlo completamente comentado en el apéndice.

Lo único que debemos tener en cuenta antes de ejecutar dicho script es la necesidad de qmail de encontrar nuestro nombre de máquina en el dns.

Aunque en estos días registrar un dominio es relativamente barato, hemos optado por uno gratuito (no necesitamos más para realizar pruebas).

7.4.4 Configuración básica

Todos los archivos de configuración de sistema de qmail residen en el directorio /var/qmail/control, a excepción de algunos archivos .qmail en alias. Esto es así por motivos de compatibilidad.

Los archivos mínimos necesarios son los siguientes:

- concurrencyincomming. Indica el número máximo de conexiones SMTP simultáneas que qmail acepta. Por defecto está a 0, aunque podemos incrementar este valor a 20 de forma que pueda realizar 20 conexiones en paralelo.
- defaultdelivery. Por defecto, para decidir de que forma se entregan los mensajes a los usuarios, se consulta el fichero .qmail de cada cuenta. Si éste no existe, qmail consulta este fichero de configuración para ver que otra opción sigue
- defaultdomain. Nombre del dominio por defecto. Se utiliza cuando mandamos un correo sin parte de la dirección. Si estando dentro del sistema un usuario le envía a otro un correo usando para ello sólo el nombre de usuario, qmail trata de rellenar la dirección con el valor defaultdomain. Si no lo indicamos explícitamente su valor es el mismo que el del fichero me.
- me. Contiene el FQDN del sistema.

Además podemos configurar otros aspectos mediante los siguientes ficheros.

- badmailfrom. Lista de direcciones de correo de la que no vamos a entregar correo. Cada línea de este fichero debe contener este esquema: @dirección, en donde indicamos la dirección rechazada.

- bouncefrom. Nombre de usuario con el que devolverán los mensajes no aceptados.
- bouncehost. Si un mensaje no se puede entregar permanentemente, qmail-send enviará un mensaje al remitente desde bouncefrom@bouncehost.
- concurrencylocal. Número máximo de intentos de entrega local. Está limitado en tiempo de compilación a 500.
- concurrencyremote. Número máximo de intentos de entrega remota. Está limitado en tiempo de compilación a 500.
- defaulthost. Nombre de máquina que se toma por defecto cuando no se indica ninguna.
- databytes. Máximo número de bytes permitidos para un mensaje.
- doublebouncehost. Máquina a la que se envía un mensaje doblemente devuelto.
- doublebounceto. Usuario que va a recibir los mensajes doblemente devueltos.
- envnoathost. Nombre de dominio que se usará para aquellas direcciones que no aparezcan con signo @.
- helohost. Nombre de máquina actual. Se usa exclusivamente para el saludo con el servidor SMTP remoto.
- idhost. Nombre de máquina para las identificaciones de mensaje. No tiene por qué coincidir con el valor del fichero me. De hecho se utiliza por si queremos que en los mensajes no aparezca el nombre real del servidor.
- localiphost. Nombre de máquina que se utiliza cuando se trata de direcciones ip internas.
- locals. Lista de nombres de dominios para los que la máquina actual recibe correo. Por defecto tiene el mismo contenido que el fichero me. Si éste no se encuentra en la lista, qmail-send no arrancará. Una dirección del tipo usuario@dominio se considera local si dominio se encuentra en locals.

- morercpthosts. Lista de nombres de dominios a los que se permiten enviar correo. Si existe rcpthosts, se añade al final de este fichero.
- percenthack. Lista de nombres de dominios donde se aplica el "truco del tanto por ciento" o percent hack. Si un dominio aparece en esta lista, cualquier dirección de la forma usuario%fqdn@dominio será reescrita como usuario@fqdn. En el usuario también puede aparecer %, así que percenthack puede ser aplicado repetidamente. qmail-send consulta primero percenthack antes de locals.
- qmqpservers. Lista de direcciones IP, una por línea, de los servidores QMQP en caso que usemos este protocolo.
- queuelifetime. Número de segundos que un mensaje puede permanecer en la cola. Por defecto es 604800 (una semana). Tras la expiración de este plazo, qmail-send tratará de enviarlo una vez más, pero cualquier fallo temporal lo considerará como permanente.
- rcpthosts. Dominios desde los que se permite enviar correo.
- smtpgreeting. Mensaje de bienvenida de smtp. La primera palabra de este mensaje debe ser el nombre de la máquina; si esto no ocurre qmail-smtpd no se ejecutatá.
- smtproutes. Rutas smtp artificiales. Se relacionan con la retransmisión de correo.
- timeoutconnect. Número máximo de segundos que qmail-remote esperará a un servidor smtp remoto que acepte la conexión. Por defecto es 60, aunque el kernel un límite superior de 75 segundos.
- timeoutremote. Número máximo de segundos que qmail-remote esperará para cada respuesta de un servidor smtp. Por defecto su valor es 1200.
- timeoutsmtpd. Número de segundos que qmail-smtpd esperará a cada búfer de datos, para cada conexión de un cliente remoto smtp.
- virtualdomains. Lista de usuarios y dominios virtuales, uno por línea.

7.4.5 Configuración avanzada. Integración.

El siguiente paso es integrar los usuarios de qmail. Tras esto, todos los usuarios que aparecen en el directorio OpenLDAP dispondrán de cuenta de correo. Esta cuenta seguirá el esquema:

usuario@dominio

De la misma forma, la clave para acceder a la cuenta será la misma que para el resto de servicios. Estas son algunas de las condiciones que imponíamos para conseguir la integración, y que ya están resueltas.

Para lograr que qmail obtenga los usuarios del directorio debemos aplicar un parche y volver a compilar el paquete. Además debemos realizar cierta configuración; hay que indicar, por ejemplo, donde se encuentra el directorio que se va a usar. Veamos los pasos.

1) En primer lugar debemos partir de una instalación limpia de qmail. Para ello vamos a usar el paquete netqmail que comentamos anteriormente. Debemos tener en cuenta que ya tenemos qmail instalado, así que nos ahorraremos ciertos pasos.

```
cd /usr/local/src
gunzip netqmail-1.05.tar.gz
tar xpf netqmail-1.05.tar.gz
cd netqmail-1.05
gunzip -c qmail-1.03.tar.gz | tar xf -
cd qmail-1.03
```

2) Aplicamos el parche qmail-ldap. Incluye además los parches referentes a errores e incompatibilidades, como por ejemplo el comentado fallo en errno.h.

```
patch -p1 < /usr/local/src/qmail-ldap-1.03-20041201.patch
```

3) Ajuste de algunos parámetros necesarios en el fichero Makefile. Para ello debemos editarlo y añadir la opción adecuada. Se trata de SHADOWLIBS. Eliminamos el comentario que aparece al principio de la

línea donde aparece dicha opción y la modificamos de esta forma:

```
SHADOWLIBS = -lcrypt
```

4) Compilación e instalación. Procedemos a construir el software.

```
make setup check
```

5) Podemos comprobar que el paso anterior se ha desarrollado satisfactoriamente si observamos que las últimas líneas de dicho proceso son éstas:

```
...
./install
./instcheck
```

Sólo queda añadir los ficheros de configuración en /var/qmail/control. Como mínimo deben ser los siguientes:

- ldapserver. Nombre de la máquina donde se encuentra el servidor OpenLDAP. Podemos incluir varios, uno por línea, a modo de redundancia.
- ldapbasedn. Debemos indicar el nodo base a partir del cual se comenzará a buscar.

7.5 Webmail

Una de las formas más habituales de acceder al correo es mediante web. Sólo se necesita un navegador y la dirección adecuada, sin necesidad de configurar ningún cliente. Por ello este sistema parece ser el más adecuado para uso en un centro de cálculo. Al cabo de cada día son muchos los estudiantes que se sientan delante de cada equipo. No es cómodo (ni seguro) que cada uno de ellos deba configurar un cliente para acceder a sus mensajes.

Además es independiente de la plataforma. Los usuarios pueden consultar el correo desde Linux o Windows.

El webmail permitirá a los usuarios mandar, recibir y gestionar sus correos. Pueden leer, copiar, corregir ortográficamente los mensajes. Básicamente pueden realizar las mismas funciones que podemos encontrarnos en un webmail comercial, como puede ser gmail o yahoo.

El webmail que se utilizará será SqWebMail. Ha sido elegido debido a su velocidad y a que es muy ligero, carga poco el servidor. Dispone además de las siguientes características:

- Accede directamente a los directorios Maildir, sin la necesidad de usar un servidor IMAP.
- La apariencia se puede modificar, en mayor parte sin cambiar el código fuente.
- Carpetas de correo jerárquicas e incluso compartidas.
- Posibilidad de configurar libretas de direcciones.
- No necesita para su ejecución ni Javascript ni cookies.
- Filtrado de correos mediante Maildrop.
- Calendario.
- Revisor ortográfico.

– Encriptación de correo y firmas digitales con GnuPG.

7.5.1 Instalación y configuración

Para la instalación de SqWebMail debemos tener en cuenta que necesita un paquete extra para funcionar. Se trata de la biblioteca de autenticación Courier. Esta será la que se encargue de autenticar a los usuarios que accedan a su correo mediante SqWebMail.

La biblioteca Courier proporciona servicios de autenticación para otras aplicaciones Courier como puede ser el servidor Courier-IMAP. Su cometido es típico: toma un par usuario – clave, y determina si es válido. También es capaz de obtener, dado un usuario, el directorio home de la cuenta, el Maildir y la cuota.

No vamos a usar estas facilidades que nos ofrece la biblioteca. Sin embargo, ofrece la posibilidad de utilizar servicios externos, como pueden ser usar PAM o LDAP directamente.

Nosotros eligiremos la opción que hemos venido tomando desde un principio, que es la de usar el módulo pam_ldap. Es la forma más versátil y sencilla que estamos manejando.

Procedemos pues a instalar la biblioteca de autenticación Courier. No indicamos las dependencias de este paquete, ya que sólo requiere un compilador y una base de datos Berkeley DB. Aspectos resueltos si hemos conseguido instalar todos los programas que hemos tratado hasta este punto.

Debemos seguir los siguientes pasos:

1) Descargar el paquete adecuado. Se trata de courier-authlib-0.55.tar.bz2. Lo descargamos de la página www.courier-mta.org/authlib/.
2) Lo descomprimimos en el directorio de trabajo /usr/local/src
   ```
   tar -jxvf courier-authlib-0.55.tar.bz2
   ```
3) Construimos el software.
   ```
   ./configure –with-redhat
   gmake
   gmake install
   gmake install-migrate
   gmake install-configure
   ```
   En las instrucciones para la instalación se advierte que durante la ejecución del comando ./configure, puede que parezca que ha quedado en un bucle infinito. Esto no es más que una ilusión óptica. El paquete courier-authlib está compuesto por muchos módulos pequeños, y es por esto por lo que da la sensación de que se ha quedado en un bucle infinito.

   Como curiosidad podemos señalar que hemos tenido que añadir un argumento al comando ./configure. Si lo ejecutamos sin parámetros , el script se para ya que detecta que el sistema es redhat y sugiere que la instalación se realice desde rpm. Las razones que argumenta se basan en la facilidad. Tras comprobar que esto no es cierto, ya que exige que se instalen otros paquetes, nos decantamos por la opción seguida durante todo el proyecto; instalar desde fuente.
4) Por último ya sólo queda arrancar el demonio de autenticación.
   ```
   /usr/local/sbin/authdaemond start
   ```
   En la instalación se nos indica que debemos añadirlo a los scripts de inicio de sistema, si queremos que se inicie junto con el sistema.

Podemos proceder ya con la instalación de SqWebMail propiamente dicha. Los pasos a seguir son similares al paquete anterior. Los mostramos a continuación:

1) Descargar el paquete adecuado. Se trata de sqwebmail-5.0.1.tar.bz2. Lo descargamos de la página www.courier-mta.org/sqwebmail/.
2) Lo descomprimimos en el directorio de trabajo /usr/local/src
   ```
   tar -jxvf sqwebmail-5.0.1.tar.bz2
   ```
3) Construimos el software.
   ```
   ./configure
   gmake configure-check
   gmake
   gmake install-strip
   gmake install-configure
   ```
4) Por último sólo queda arrancar el servicio y añadir una tarea al cron. El servicio se arranca mediante el siguiente comando:
   ```
   /usr/lib/sqwebmail/libexec/sqwebmaild.rc start
   ```
   La tarea añadida al cron se encarga de limpiar la caché del webmail. Se va a realizar cada 20 minutos.
   ```
   */20 * * * * root /usr/lib/sqwebmail/share/sqwebmail/cleancache.pl
   ```

Tras esto debemos crear el fichero PAM referente al servicio. En concreto será /etc/pam.d/webmail, y su contenido lo mostramos a continuación.

```
#%PAM-1.0 Fichero /etc/pam.d/webmail
auth            required          pam_ldap.so
account         required          pam_ldap.so
```

Ya sólo queda ejecutar el webmail. Debemos tener en cuenta que es trata de un script cgi, y que por tanto necesita un servidor web para su funcionamiento. En la instalación que realizamos de red hat viene por defecto apache, que será el que utilizaremos para las pruebas.

Los scripts cgi deben ser encontrarse en el directorio /var/www/cgi-bin para su ejecución. SqWebMail copia automáticamente su script en esta localización, así que sólo tenemos que escribir la dirección en un navegador, y listo.

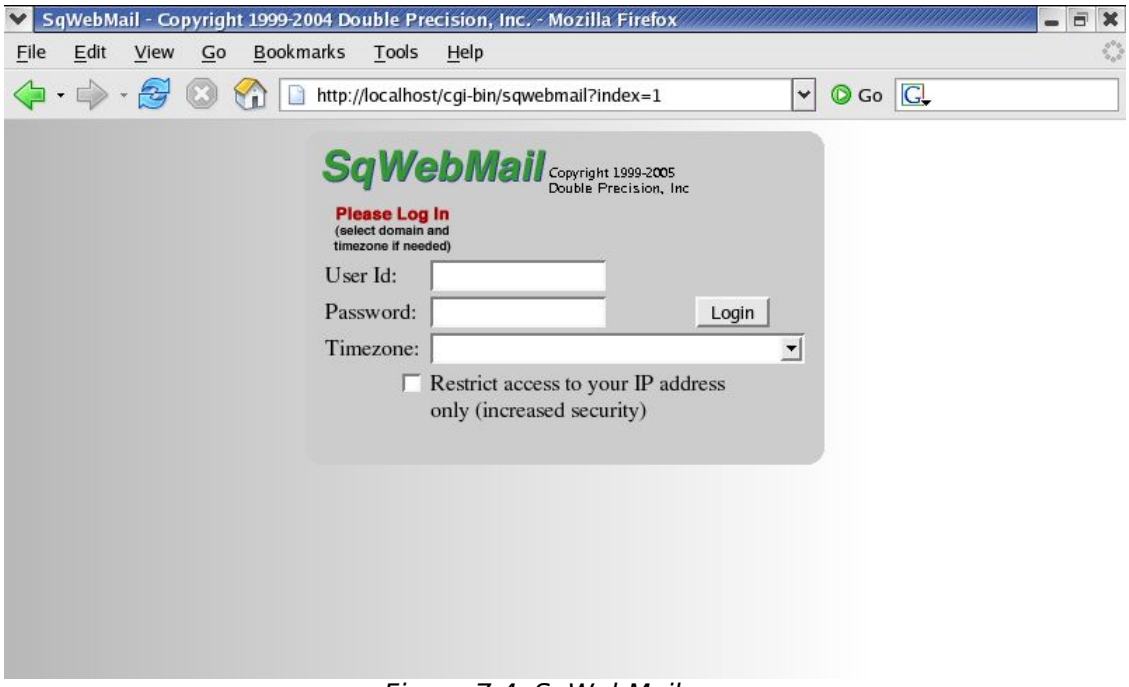

Figura 7.4 SqWebMail

Tal como ya hemos comentado al principio, las pantallas de SqWebMail se pueden personalizar. La primera pantalla es una de las que debemos hacerlo. Por lo menos debemos indicar el nombre de la organización que se trate y algún icono o logo identificativo.

Podemos comprobar si la autenticación se encuentra integrada. Intentamos acceder mediante uno de los usuarios creados en el directorio OpenLDAP, y que no se encuentran en el fichero /etc/passwd. Mostramos de paso la pantalla de bienvenida de SqWebMail al usuario.

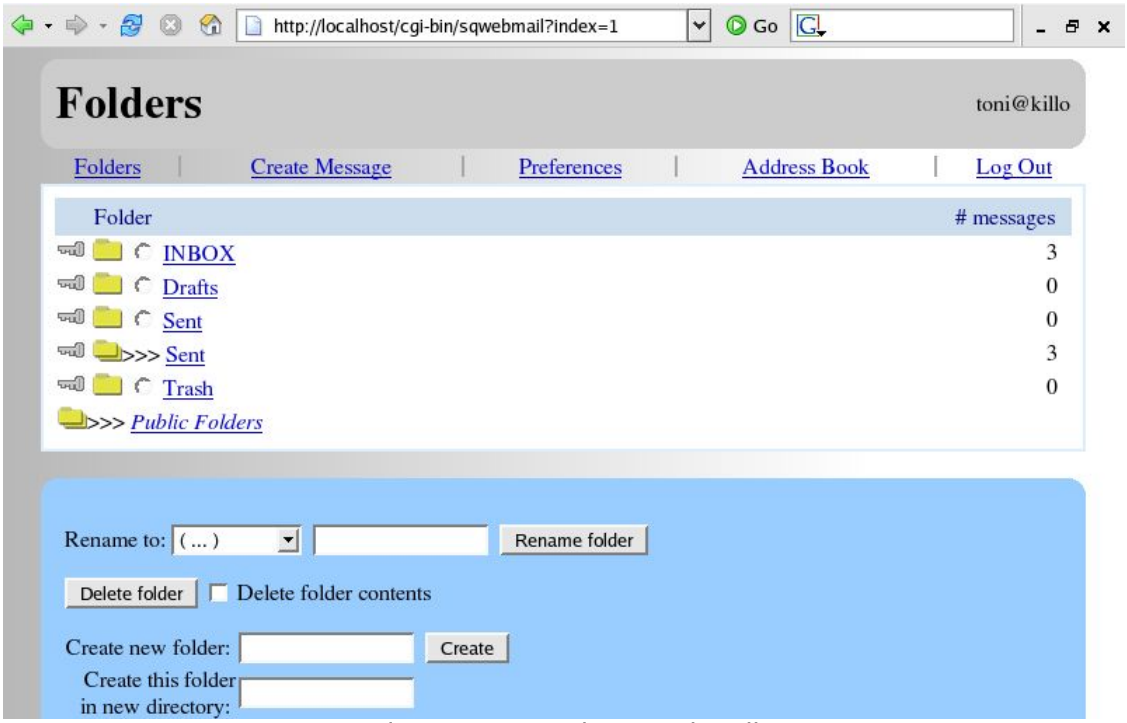

Figura 7.5 Uso de SqWebMail

7.5.2 Seguridad

La seguridad del sistema en general se ve comprometida con el correo. Al introducir el nombre de usuario y contraseña, éstos se transmiten sobre una conexión http, en texto plano. Podemos comprobar este hecho en la siguiente captura. Hemos usado para ello el sniffer ethereal.

Se puede ver, sin tener que buscar demasiado, que junto al nombre de usuario se encuentra el identificativo password seguida de otra palabra. Esta palabra "alskdj" es el password que le corresponde al usuario "toni".

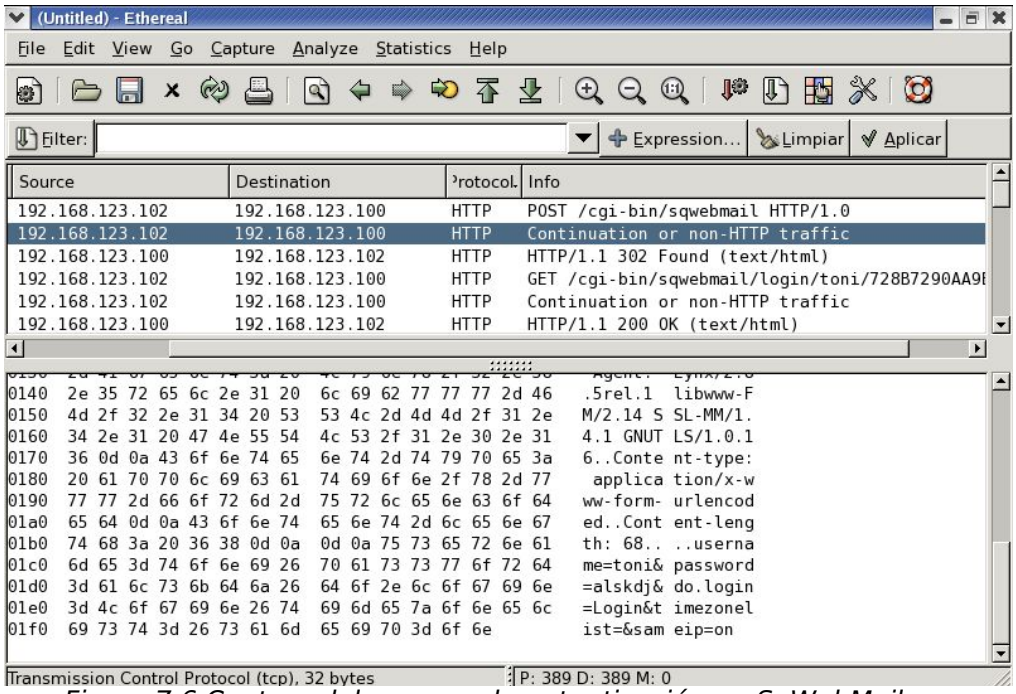

Figura 7.6 Captura del proceso de autenticación en SqWebMail

La solución que suelen tomar los webmails comerciales es usar una capa segura. Mediante https la comunicación entre servidor (SqWebMail) y cliente (nosotros) se encripta, y por tanto, se protege de cualquier interceptación. Es la opción que vamos a seguir.

Para que el servidor apache del que disponemos en Red Hat tenga la posibilidad de usar SSL, debemos instalar un nuevo paquete. Se trata de mod_ssl y permitirá que el servidor acepte conexiones seguras.

Los pasos a seguir son:

1) Descargar el paquete. El nombre completo del paquete que corresponde con nuestra distribución es mod_ssl-2.0.40-21.i386.rpm. Tenemos además la posibilidad de instalarlo directamente desde el menú Configuración del sistema, añadir/eliminar aplicaciones (está incluido en Red Hat 9). Se encuentra en el primer cd de la distribución.
2) Instalación. Para ello hacemos uso de la herramienta rpm

```
rpm -ivh mod_ssl-2.0.40-21.i386.rpm
```

3) Configuración. Para obtener la funcionalidad deseada sólo debemos crear un certificado digital para apache. Para ello:

```
cd /etc/http/conf
make server.key
```

Mostramos la ejecución de este último comando, que pedirá una palabra de paso y la confirmación de ésta.

```
Umask 77 ; \
/usr/bin/openssl genrsa -des3 1024 > server.key
Generating RSA private key, 1024 bit long modulus
.....................++++++
.................................++++++
e is 65537 (0x10001)
Enter pass phrase:
Verifying - Enter pass phrase:
```

4) Reinciamos el servidor apache.

```
/etc/init.d/http restart
```

Vamos ahora a comprobar lo dicho. Para ello accederemos mediante https y no http como lo hemos venido haciendo hasta ahora.

Figura 7.7 Conexión segura mediante https

Mediante el sniffer que venimos usando podemos comprobar como durante el acceso no se transmite información sensible. Lo único que vemos es información encriptada mediante TLS.

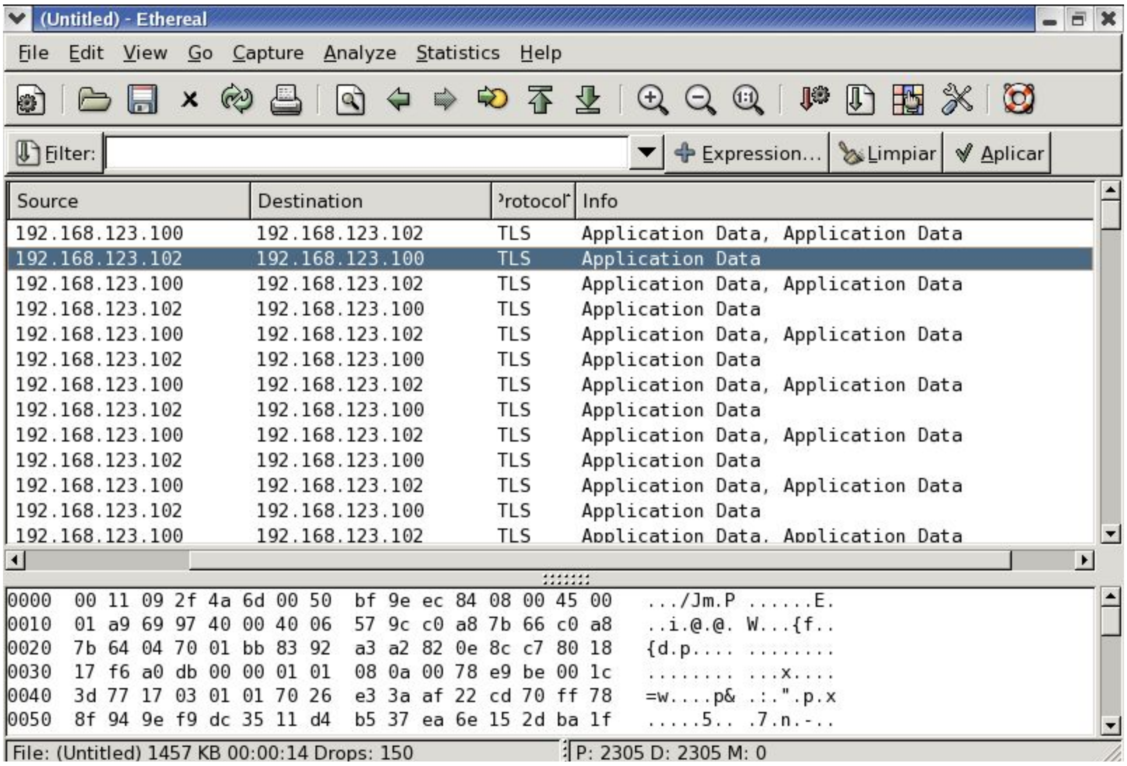

Figura 7.8 Tráfico encriptado entre cliente y servidor