7

Servidor de Correo: qmail 1.03

1- **Servicios de Mensajeria sobre TCP / IP**

2- **Qmail**

3- **Referencias**

4- **Anexos**

INDICE

1- Servicios de mensajeria sobre TCP/IP

a) Smtp
b) Formato del mensaje – rfc 822 + mime 1.0
c) Pop3
d) Seguimiento real de una sesion smtp.
e) Seguimiento real de una sesion pop3
f) Despliegue de un servidor de correo en internet.

2- Qmail

a) Historia
b) Licencia
c) Comparación con otros servidores de correo.
d) Estructura interna del mta – qmail
e) Instalación y configuración
    i. Obtención del código fuente
    ii. Compilación de qmail y sus dependencias
        1. Qmail
        2. Ucspi-tcp
        3. Daemoons tools
    iii. Estructura de directorios
    iv. Estructura de la cola del mta
    v. Ficheros de configuración
    vi. Iniciar qmail
    vii. Qmail-smtpd
        1. Retransmisión – mta (relay)
        2. Tipos de configuración de qmail para el relay.
        3. Script de inicio de qmail-smtpd
        4. Inicio de qmail-smtpd
        5. Comprobación
        6. Alias
        7. Maildir
        8. .qmail
        9. Script para añadir cuentas de correo.
    viii. Qmail-pop3
        1. Estructura
        2. Checkpassword
        3. Script de inicio de qmail-pop3
        4. Comprobacion
    ix. Prueba de envio recepción
g) Dominios virtual
h) Nombre de servidor multiples

3- Referencias

1 Servicios de Mensajeria sobre TCP / IP

Actores que intervienen sobre los sistemas de mensajeria:

1- **MUA(Message User Agent)** – Agente de Usuario, es el encargado de la composición del mensaje que queremos enviar y de su entrega al servidor de mensajería (MTA).

Ejemplos de MUA: Outlook, Evolution, Bat, etc..

2- **MTA (Message Transfer Agent)** – El MTA acepta mensajes de los MUA y de los otros MTA's, procesa su cabecera y lo reenvía hacia el MTA destinatario utilizando el protocolo SMTP.

Ejemplos de MTA: qmail-smtpd (SMTP) + el sistema de encolado y reenvio Gestión y control del encolado: (qmail-queue + qmail-send) + Sistema de entrega remota: (qmail-rspawn + qmail-remote) + Sistema de entrega local: (qmail-lspawn + qmail-local)

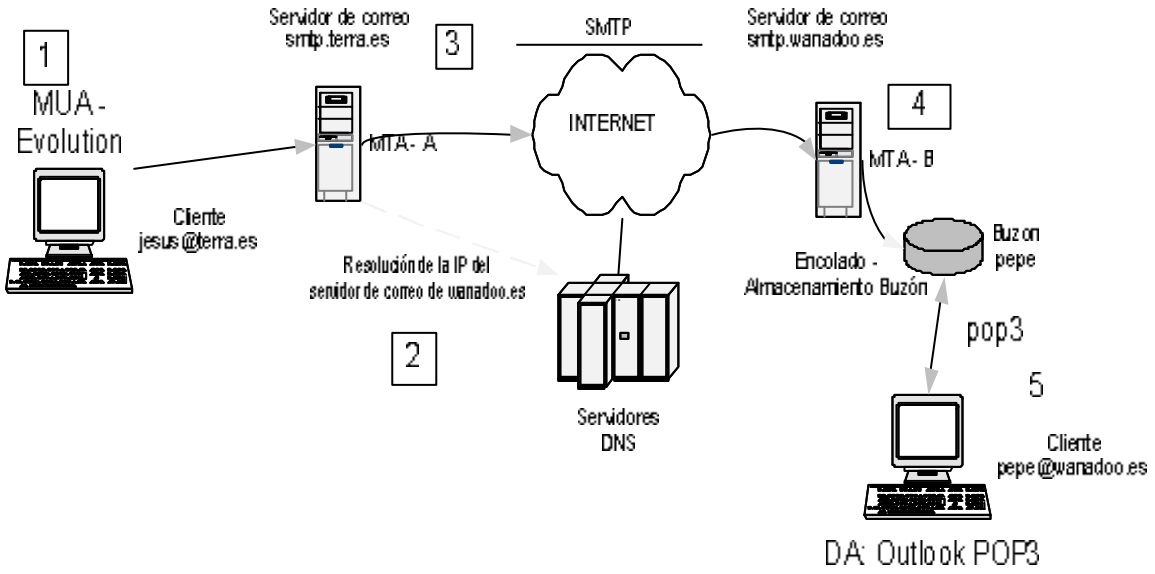
3- **DA (Delivery Agent)** Proporciona uno o varios mecanismos de recuperación de mensajes sobre usuarios autorizados.

Ejemplos de DA: POP3: qmail-pop3 , vm-pop3d , IMAP: Courier

Protocolos comunes:

SMTP: Para el envio de mensajes por MUA y para el intercambio de mensajes entre MTA's.

POP3: Para la gestión del buzón de correo.



1- El originante (jesus@terra.es) compone el mensaje con su MUA(cliente de correo, en nuestro caso el Evolution para linux). En la cabecera es necesario indicar al menos el campo del destinatario (To). Una vez compuesto el mensaje, solicita su envío. El MUA añade algunos campos a la cabecera (Date, Message-Id, ..) e inyecta el mensaje en el sistema de correo (Servidor MTA del originante). Se puede realizar de 2 formas: ejecutando un programa de inyección (por conexión local al MTA ej, qmail-inject,sendmail) o bien mediante una conexión al SMTP al puerto 25 del servidor (conexión remota).

2- Si la inyección del MUA tiene éxito dentro del sistema de colas del MTA, el proceso de gestión de las colas de mensajes activa el envió de los mensajes que permanecen en cola y realiza las conexiones con bs MTA remotos de los dominios dónde ha de enviar los mensajes, extrayéndolos de la dirección de mail de destino. Ejemplo: pepe@wanadoo.es → Resolución de la dirección IP del MTA de wanadoo.

3- Una vez resuelta la IP, se reenvía el mensaje al MTA de destino (MTA-B de wanadoo.es)

4- El MTA-B de destino, elige lo que hacer con el mensaje recibido del MTA-A, si lo rechaza el MTA-B prepara un mensaje de devolución (bounce) a la dirección de retorno para notificar el problema. Si lo acepta, analiza la dirección para determinar si el mensaje debe entregarse a una dirección local o remota(por si se realiza un enrutamiento del mensaje a otro servidor SMTP <>).

5- Mediante un DA, podemos recuperar nuestro mensaje en el formato adecuado (ej: texto plano, HTML, XHTML, otros formatos según la potencia del DA).

1.1 Smtp: Simple Mail Transfer Protocol – RFC 821

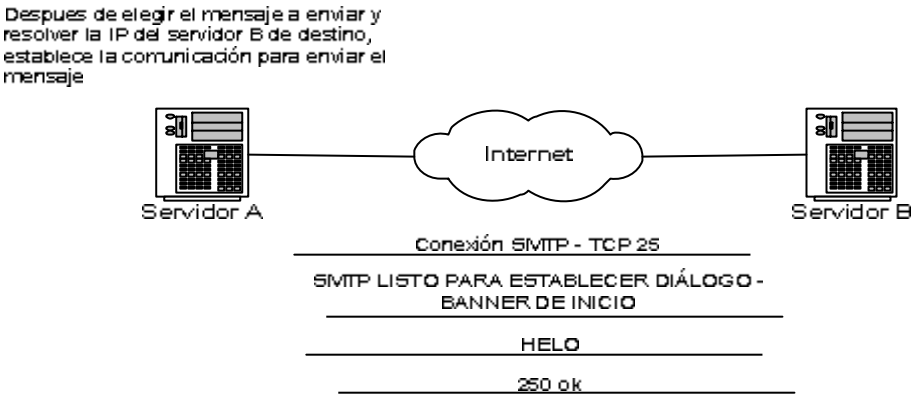
El correo electrónico entre hosts se gestiona mediante este protocolo descrito en la RFC 821 y equivalente al P3 de OSI. Se basa en el intercambio de comandos y respuestas formados por secuencias alfanuméricas codificadas en ASCII y delimitadas por CLRF

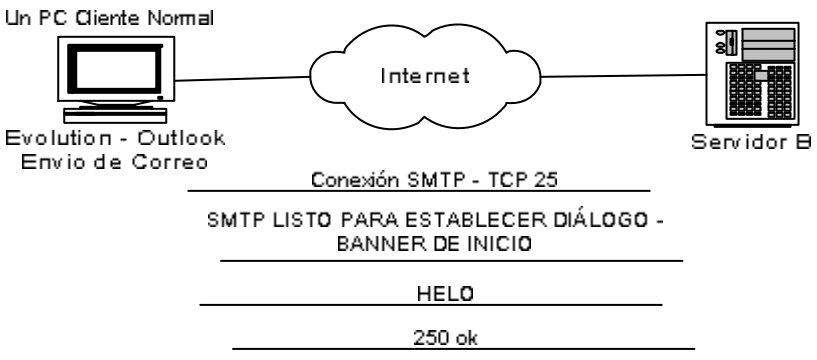
Se distinguen 3 fases en el diálogo:

1- **Establecimiento:**

a. Comunicación Servidor(A) – Servidor(B)
Servidor A: Conexión al puerto de escucha del SMTP (Por defecto el 25)

R: 220 <Servidor – B> Simple Mail Transfer Service Ready
C: HELO <Servidor – A >
R: 250 <Servidor – B >





2. **Intercambio:**

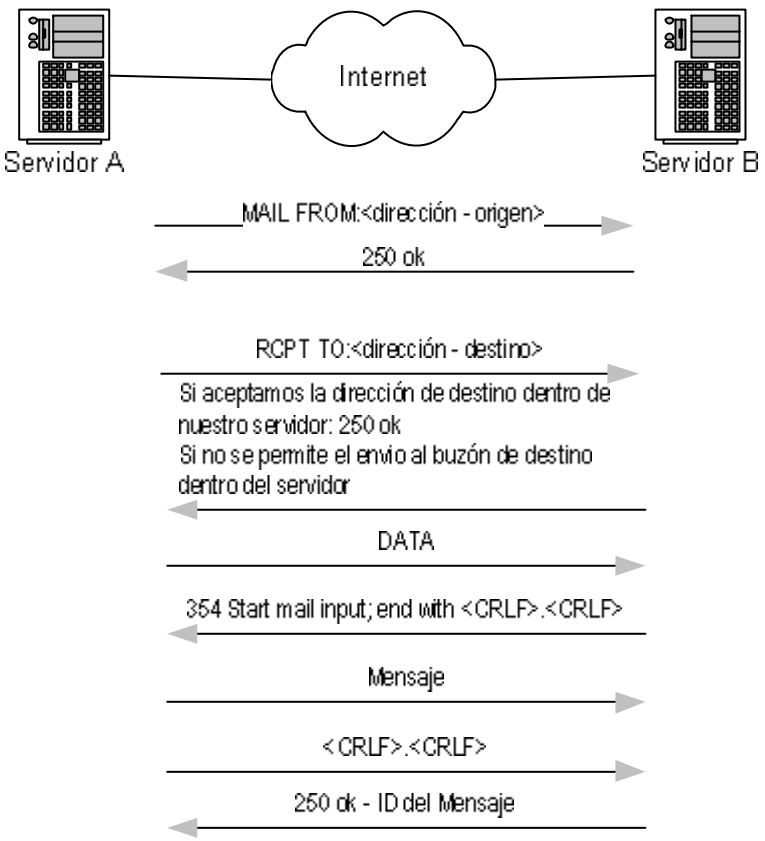
C: MAIL FROM: <direc- origen>
R: 250 OK

C: RCPT TO: <direc- destino>
R: 250 OK

C: DATA

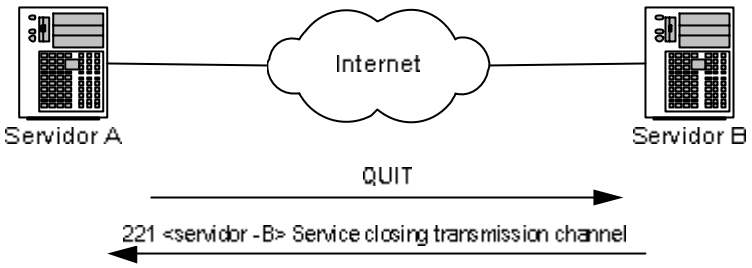
R: 354 Start mail input; end with <CRLF>.<CRLF>
C: Mensaje..
C: Mensaje..
C: Mensaje..
C: .

R: 250 OK



3. Liberación

C: QUIT
R: 221 <servidor –B> Service closing transmission channel



Otros comandos:

RSET:
Indicamos la anulación de la transacción de correo en curso.

SEND FROM: <Dirección de origen>
Envió de correo directamente al terminal, no al buzón.

SOML FROM: <Dirección de origen>
Se utiliza para enviar correo directamente al terminal, si no esta activo el terminal se envía al buzón.

SAML FROM: <Dirección de origen>
Se envía el correo al terminal y a el buzón a la vez.

VRFY <dirección-destino>: Se utiliza para verificar una dirección de usuario o lista de distribución. Posibles respuestas a este comando podrían ser:

C: VRFY Jesús
R: 250 Jesús Soto jesussoto@mifamilia.com
R: 251 User not local; Will forward to jesus@terra.es
R: 550 String does not match anything
R: 551 User not local; please try jesus@terra.es
R: 253 User ambiguous

EXPN <dirección>: Se utiliza para expander una lista, obteniendo la relación de direcciones de correo asociadas a una lista de distribución. Posibles respuestas:

C: EXPN <nombre lista>
R: 250- Usuario1 <direc-destino1>
R: 250- Usuario2 <direc-destino2>
R: 250- <direc-destino3>
R: ..

R: 550 Access Denied to You
ó
R: 550 That is a user name, not a mailing list.

HELP: Ayuda

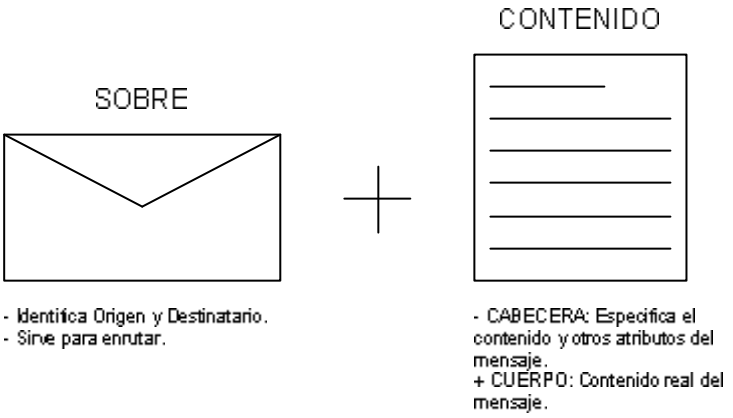
NOOP: No realiza ninguna acción, sirve para mantener viva la conexión.

TURN: Solicita intercambiar los papeles de emisor y receptor de comandos y respuestas. Si se acepta se responde con 250 OK, si se rechaza se responde con 502.

1.2 Formato del mensaje

RFC 822 :Especifica el formato de los mensajes de correo.
(Pág 7 RFC 822 – Descripción Dirección e-mail.)

SOBRE (envelope) + Contenido (Cabecera + Cuerpo)



Message-Id: Número único para referencia
In-Reply-to: Id. del mensaje que se contesta
References: Otros Id del mensaje
Keywords: Palabras claves de usuario
Subject: Asunto del que trata el mensaje
X-Usuario: Definibles por el usuario

LA CABECERA SE EXTIENDE CON MIME-Version 1.0 (RFC 2045,2046,2047, 2048):

- Posibilidad de enviar múltiples cuerpos dentro de un contenido.
- Añadir contenido de cualquier tipo, no solo texto. Archivos de aplicación, imágenes, etc..
- Utilizar en los mensajes de texto un juego de caracteres distinto del US-ASCII
- Utilizar en la cabecera de los mensajes un juego de caracteres distinto del US-ASCII

Campos de MIME añadidos a la RFC 822:

- Content-Type: tipo/subtipo
  - Tipo: text, image, audio, video, application, message, multipart.
  - Subtipo:
    - mixed (mensaje con partes independientes con tipos de codificación independientes).
    - Alternative: indica que el mensaje contiene distintos formatos de representación de la misma información.
    - Parallel: indica que el mensaje incluye subpartes que deben ser vistas juntas al mismo tiempo (audio/video)
    - Digest: indica que el mensaje contiene a su vez un conjunto de mensajes.

Ejemplo:

From: jesus@terra.es
To: lolo@wanadoo.es
MIME-version: 1.0
Content-Type: multipart/mixed, boundary=StartOfNextPart

--StartOfNextPart
Tio, te mando una foto!.

--StartOfNextPart
Content-Type: image/gif
Content-Transfer-Enconding: base64 → #modo de transferencia de octetos.

H23298ISAAS¿D`+sd¡'¡¡'¡' → octetos correspondientes a la foto. (Base64)

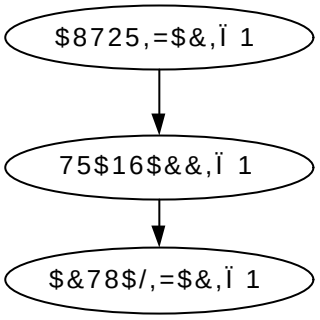
1.3 Pop3 - Post Office Protocol - Version 3

Este protocolo nos da la posibilidad de poder manipular un buzón de correo situado en un servidor remoto.

Equiparable al P7 de la propuesta OSI. Complementa a SMTP!.

Con POP3 descrito en la RFC 1939

HAY 3 FASES DENTRO DEL PROTOCOLO:



1- AUTORIZACIÓN

a. **USER name :** Identificador del usuario.
b. **PASS string :** Password para el usuario.

c. QUIT
d. APOP

2- TRANSACCIÓN

a. **STAT:** Nos da el estado del buzón.
Nos responde con +OK nn mm, siendo nn el número de mensajes y mm el tamaño total de los mensajes.

C: STAT
R: +OK 2 320

b. **LIST[msg]:** Nos da la estadistica del buzón.
Responde con +OK nn messages (mm octects) y a continuación la longitud de cada mensaje contenido.

C: LIST
R: +OK 2 messages (320 octets)
R: +OK 1 120
R: +OK 2 220
R:.

C: LIST 2
R: +OK 2 200

c. **RETR msg :** Recuperamos un mensaje del buzón, siempre que no este marcado para borrar. Finaliza con <CLRF>.<CLRF>

d. **DELE msg :** Marca un mensaje para eliminar. La eliminación se producirá cuando se pase a la fase de actualización realizada por el proceso correspondiente. En caso contrario no se producirá dicha eliminación.

e. **NOOP :** Permite mantener la conexión activa. No hace nada funcional.

f. **RSET :** Elimina las marcas de los mensajes borrados con el comando DELE.

g. **TOP msg n :** Recupera las n filas de 1 mensaje.

C: TOP 1 10
R: +OK
R: 10 primeras lineas del mensaje 1 incluyendo la cabecera.
R: .

h. **UIDL msg :** Solicita el identificador único del mensaje indicado.

3. ACTUALIZACIÓN

C:QUIT

R: +OK dewey POP3 server signing off.
ó
R: -ERR some deleted messages not removed.
ó
R: +OK dewey POP3 server signing off(maildrop empty).
Ó
R: +OK dewey POP3 server signing off(2 message left).

Proceso de envió de un mail

<Un servidor sin control MX(Ver resol. Dir inversas)>

1- NOS CONECTAMOS A EL SERVIDOR **SMTP** (Nuestro MTA)
#telnet <dir-ip> 25

HELO
250 mx.novalinux.net

MAIL FROM:root@mx.novalinux.net
250 ok

RCPT TO: jesussoto@mx.novalinux.net
250 ok

DATA
Subject:IMPORTANTE
Configurar un servidor de correo no es fácil
.
250 ok 1049243819 qp 1637

2- NOS CONECTAMOS A NUESTRO SERVIDOR **POP**
#telnet <dir-ip> 110

+OK 1757.1049244732@jesus.novalinux.net

USER jesussoto
+OK

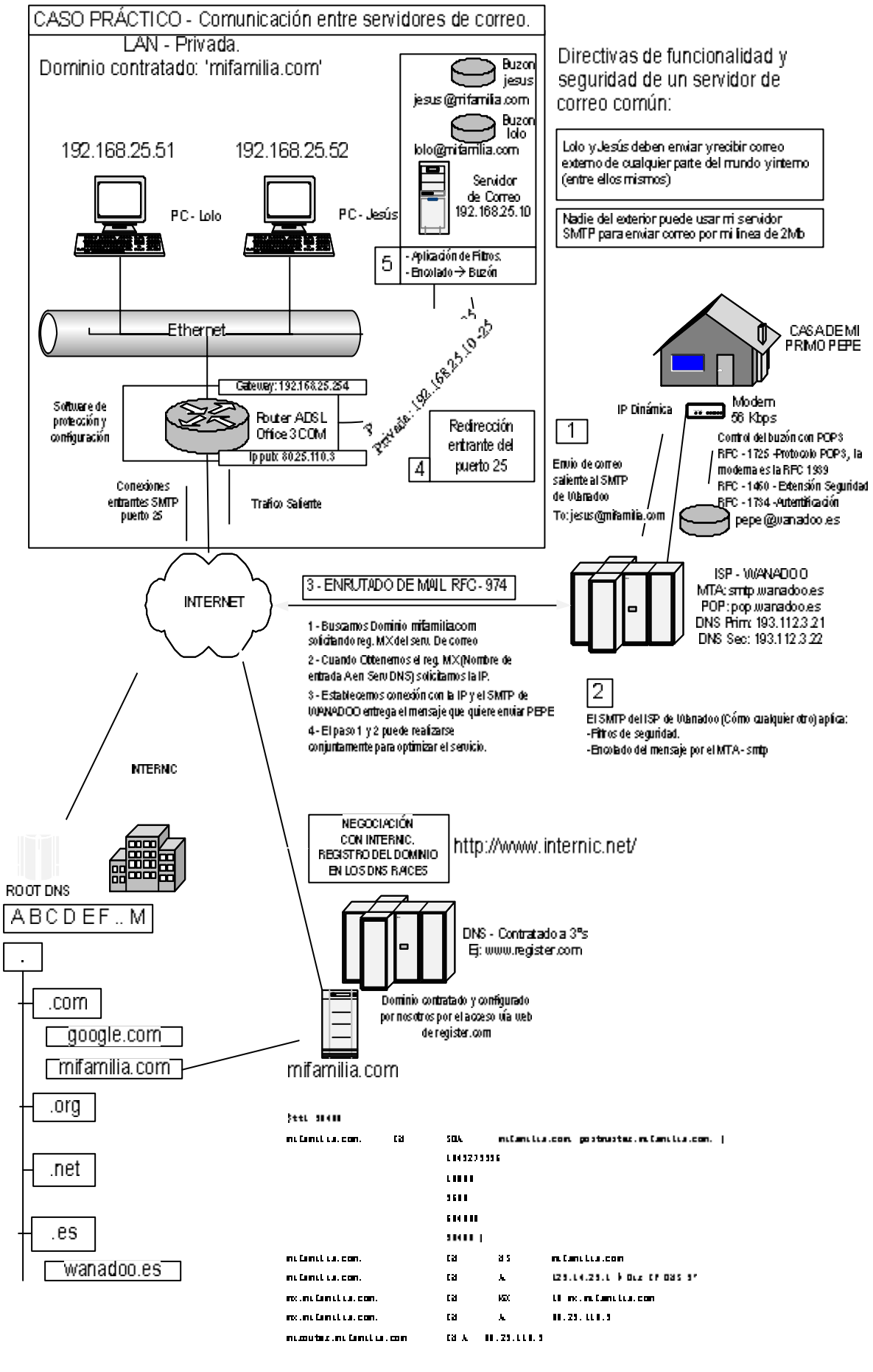
PASS mipass
+OK

LIST
1 250
+OK

TOP 1 10
+OK
Return-Path: <root@mx.novalinux.net>
Delivered-To: jesussoto@jesus.novalinux.net
Received: (qmail 1373 invoked by uid 1651); 1 Apr 2003 23:15:01 -
0000
Received: from localhost (127.0.0.1)
by localhost with SMTP; 1 Apr 2003 23:15:01 -0000
Subject:IMPORTANTE
Configurar un servidor de correo no es fácil

MTA (Mail Transfer Agent):

Son los servidores con la capacidad de proporcionar buzones a los usuarios reconocidos en el sistema y establecen conexiones con servidores remotos para el intercambio de mensajes.


CASO PRÁCTICO - Comunicación entre servidores de correo.
LAN - Privada.
Dominio contratado: 'mifamilia.com'
Buzon jesus
jesus@mifamilia.com
Buzon lolo
lolo@mifamilia.com
Servidor de Correo 192.168.25.10
192.168.25.51
192.168.25.52
PC - Lolo
PC - Jesús
5
- Aplicación de Filtros.
- Encolado → Buzón
Ethernet
Gateway: 192.168.25.254
Software de protección y configuración
Router ADSL Office 3 COM
Ip puti: 80.25.110.3
Redirección entrante del puerto 25
4
Conexiones entrantes SMTP puerto 25
Trafico Saliente
Directivas de funcionalidad y seguridad de un servidor de correo común:
Lolo y Jesús deben enviar y recibir correo externo de cualquier parte del mundo y interno (entre ellos mismos)
Nadie del exterior puede usar mi servidor SMTP para enviar correo por mi línea de 2Mb
CASA DE MI PRIMO PEPE
IP Dinámica
Modem 56 Kbps
Control del buzón con POP3
RFC - 1725 -Protocolo POP3, la moderna es la RFC 1939
RFC - 1460 - Extensión Seguridad
RFC - 1734 -Autentificación
pepe@wanadoo.es
1
Envio de correo saliente al SMTP de Wanadoo
To: jesus@mifamilia.com
INTERNET
3 - ENRUTADO DE MAIL RFC - 974
1 - Buscamos Dominio mifamilia.com solicitando reg. MX del serv. De correo
2 - Cuando Obtenemos el reg. MX (Nombre de entrada A en Serv DNS) solicitamos la IP.
3 - Establecemos conexión con la IP y el SMTP de WANADOO entrega el mensaje que quiere enviar PEPE
4 - El paso 1 y 2 puede realizarse conjuntamente para optimizar el servicio.
ISP - WANADOO
MTA: smtp.wanadoo.es
POP: pop.wanadoo.es
DNS Prim: 193.112.3.21
DNS Sec: 193.112.3.22
2
El SMTP del ISP de Wanadoo (Cómo cualquier otro) aplica:
-Filtros de seguridad.
-Encolado del mensaje por el MTA - smtp
INTERNIC
NEGOCIACIÓN CON INTERNIC. REGISTRO DEL DOMINIO EN LOS DNS RAICES
http://www.internic.net/
ROOT DNS
A B C D E F .. M
.
.com
google.com
mifamilia.com
.org
.net
.es
wanadoo.es
DNS - Contratado a 3ºs
Ej: www.register.com
Dominio contratado y configurado por nosotros por el acceso vía web de register.com
mifamilia.com

2 QMAIL

Historia

qmail está desarrollado por Dan Bernstein (DJB), http://pobox.com/~djb/djb.html, un profesor de matemáticas ahora en la Universidad de Ilinois en Chicago. El Dr. Bernstein es asimismo bien conocido por su trabajo en el campo de la criptografía y por su juicio contra el gobierno de EE.UU. con respecto a la publicación de código fuente relativo a encriptación. Véase http://www.news.com/News/Item/0,4,36217,00.html?owv para más información sobre el proceso judicial.

El primer lanzamiento público de *qmail*, versión *beta* 0.70, tuvo lugar el 24 de enero de 1996. La primera versión gamma, *0.90*, se lanzó el 1 de agosto de 1996.

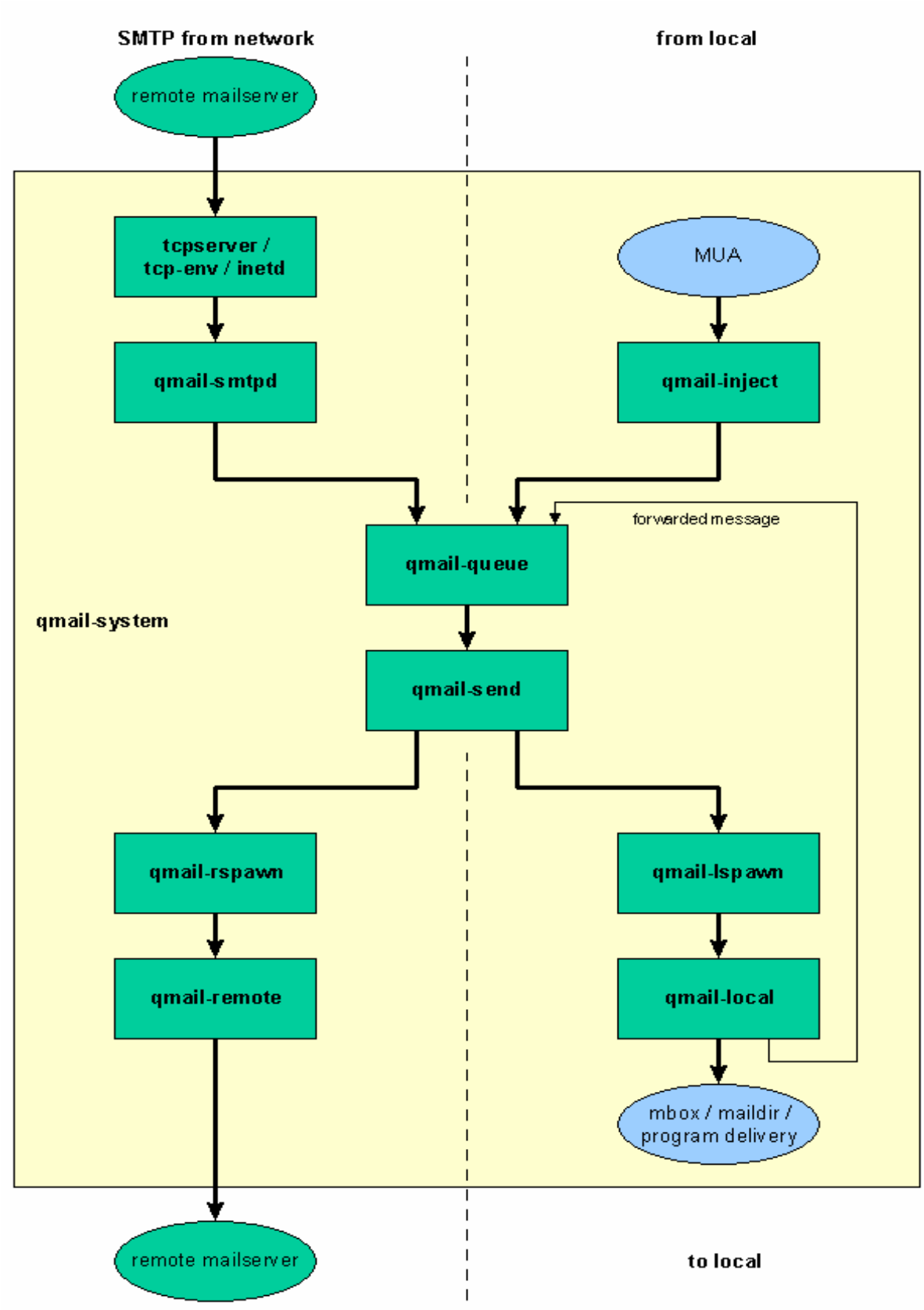
La versión 1.0, el primer lanzamiento general, se anunció el 20 de febrero de 1997. La versión actual, 1.03, se lanzó el 15 de Junio de 1998.

Licencia

El copyright de *qmail* pertenece a su autor, **Dan Bernstein**, y no se distribuye con una declaración de derechos del usuario. En http://pobox.com/~djb/softwarelaw.html,el autor resume lo que él considera que son los derechos del usuario bajo las leyes de copyright de los EE.UU. de Norteamérica. En http://www.es.qmail.org/documentacion/distro/dist.php3 concede el derecho de distribuir el código fuente de qmail. Las distribuciones binarias se permiten bajo los términos descritos aquí y en http://www.es.qmail.org/documentacion/autor/var-qmail.php3, aunque, a fecha de hoy, no existen tales intentos.

La letra pequeña es que **puede** usar *qmail* con cualquier finalidad, y puede redistribuir libremente distribuciones de código fuente de *qmail* pero sin modificaciones, puede certificar distribuciones binarias var-qmail, y puede redistribuir parches para *qmail*. Pero **no puede** distribuir código fuente de *qmail* modificado o distribuciones de binarios que no sean var-qmail.

Estructura interna de QMAIL



Descripción

1- Recepción de correo remoto desde otro servidor de correo. (Smtp from Network)

Ocurre cuando otro servidor de correo remoto entrega correo dentro del sistema QMAIL.

1.1- Se realiza el primer filtrado por TCPSERVER o TCP-ENV O INETD.

Estos procesos gestionan las conexiones TCP realizadas para no saturar un equipo. **tcpserver** es el proceso más adecuado para qmail, programa creado por el mismo autor de QMAIL el Dr. Bernstein

Cuando llega una conexión TCP por el puerto 25, tcpserver captura la conexión, la gestiona y pasa el control si es posible a **qmail-smtpd** (servidor SMTP) estableciendo el enlace entre los ordenadores que intervienen en la conexión.

Usamos el mejor: TCPSERVER (Paquete ucspi-tcp)

- **Espera conexiones entrantes, y transfiere el control a un servicio según su configuración por cada conexión.** El programa recibe unas variables de entorno manejadas por tcpserver (TCPLOCALIP, TCPLOCALHOST, TCPLOCALPORT, TCPREMOTEHOST, TCPREMOTEINFO, TCPREMOTEIP, TCPREMOTEPORT).

- Ofrece un límite de concurrencia, para proteger el sistema de sobrecarga limitando el número de procesos. Cuando se gestionan 40 conexiones simultaneas (por defecto), **tcpserver va posponiendo fluidamente la aceptación de nuevas conexiones.**

- **Proporciona controles de acceso TCP**, similares a host.allow del paquete tcp-wrappers/tcpd pero mucho más rápidos. Las reglas de control de acceso son compiladas en un formato precalculado con cdb de modo que puedan gestionar miles de máquinas distintas.

1.2 Servicio QMAIL-SMTPD

Acepta/Rechaza los mensajes recibidos vía SMTP. Se ejecuta bajo tcpserver.

Usa una serie de archivos de configuración para detectar de dónde provienen los mensajes para saber si ha de aceptarlos o rechazarlos.

1.3 Servicio QMAIL – QUEUE

Almacenamiento de mensajes encolados. Se albergan los mensajes en los directorios especiales de qmail para su posterior tratamiento.

1.4 Servicio QMAIL – SEND

Procesa los mensajes existentes en cola, enviándolos correspondientemente a los servicios de entrega LOCAL o REMOTA.

1.5 Servicio QMAIL - RSPAWN / QMAIL – REMOTE

Servicios de entrega REMOTA.

1.6 Servicio QMAIL – LSPAWN / QMAIL – LOCAL

Servicio de entrega LOCAL.

Inyección Local – MUA

En esta modalidad podemos entregar un mensaje directamente en nuestro equipo mediante un cliente (MUA).

Podemos usar qmail-inject que inserta un mensaje en la linea de comandos.

echo To: jesussoto@mx.upsam.net | /var/qmail/bin/qmail-inject

Proceso de instalación

Requisitos – (Dependen del número de usuarios a gestionar):

- Soporte Hardware - software para acceso a una Intranet ó a Internet.
- Sistema de desarrollo en C completo y en funcionamiento, con un compilador, los ficheros de cabecera del sistema, y las bibliotecas. Es necesario para compilar el código fuente de qmail → En linux gcc (En debian '**apt-get install gcc**')
- Un DNS configurado en **/etc/resolv.conf** para poder resolver direcciones de otros MTA's externos. Si se instala un servidor de correo corporativo, es recomendable tener un servidor propio de DNS.
- Espacio en disco suficiente para la cola de correo. Los sistemas pequeños para un único usuario precisan solamente un par de megabytes libres. Los servidores grandes es recomendable que sean escalables, el tamaño depende del número de usuarios que tenga que gestionar.

Un sistema operativo compatible. La mayor parte de las variantes de UNIX son válidas. Véase README en el árbol de código fuente para una lista de versiones compatibles conocidas. – LINUX ES COMPATIBLE CON QMAIL.

Obtención del código fuente

Bien, ya tiene un sistema que cumple con los requisitos, y preparado para instalar *qmail*. El primer paso es obtener el código fuente para *qmail* y otros complementos. Necesitará *qmail*, claro, y probablemente necesite también ucspi-tcp y daemontools:

- *qmail*, ftp://koobera.math.uic.edu/www/software/qmail-1.03.tar.gz
- ucspi-tcp, ftp://koobera.math.uic.edu/www/software/ucspi-tcp-0.84.tar.gz
- daemontools, ftp://koobera.math.uic.edu/www/daemontools/daemontools-0.61.tar.gz

1- DESCOMPRIMIR LOS FUENTES DESCARGADOS:

tar –xzvf qmail-1.0.3.tar.gz
tar –xzvf ucspi-tcp-0.84.tar.gz
tar –xzvf daemontools-0.61.tar.gz

2- NOS METEMOS EN EL DIRECTORIO DE LOS FUENTES DE QMAIL

cd /paquetes/qmail-1.0.3

3- AÑADIMOS LOS USUARIOS Y GRUPOS ESPECIALES QUE UTILIZA QMAIL

3.1 – COPIAMOS EL ARCHIVO INSTALL.ids

cp INSTALL.ids IDS

3.2 – EDITAMOS EL ARCHIVO 'IDS' y dejamos las lineas de script de los usuarios de QMAIL para **LINUX**

3.3 – DAR PERSIMOS DE EJECUCIÓN A IDS

chmod 700 IDS

3.4 – EJECUTAR EL ARCHIVO IDS

./IDS

```
groupadd nofiles
  useradd -g nofiles -d /var/qmail/alias alias
  useradd -g nofiles -d /var/qmail qmaild
  useradd -g nofiles -d /var/qmail qmaill
  useradd -g nofiles -d /var/qmail qmailp
  groupadd qmail
  useradd -g qmail -d /var/qmail qmailq
  useradd -g qmail -d /var/qmail qmailr
  useradd -g qmail -d /var/qmail qmails
```

3.5 – COMPILAR QMAIL

make setup check

3.6 – CONFIGURAR EL DOMINIO

make config-fast mx.upsam.net

Putting mx.upsam.net into control/me...

Putting upsam.net into control/defaultdomain...

Putting upsam.net into control/plusdomain...

Putting mx.upsam.net into control/locals...

Putting mx.upsam.net into control/rcpthosts...

Now qmail will refuse to accept SMTP messages except to mx.upsam.net.

Make sure to change rcpthosts if you add hosts to locals or virtualdomains!

3.7 – HOMOGENEIZAR LA ESTRUCTURA DE DIRECTORIOS CON ENLACES SIMBOLICOS.

ln -s /usr/man /var/qmail/man
ln -s /etc/qmail /var/qmail/control
ln -s /usr/sbin /var/qmail/bin

Instalación DE UCSPI-TCP

URL – Descarga: http://cr.yp.to/ucspi-tcp.html

URL – Documentación. http://cr.yp.to/ucspi-tcp.html

Anteriormente ya se han «desempaquetado» los .tar.gz de *qmail*, ucpsi-tcp y daemontools. Nos metemos en el directorio ucspi-tcp-0.84 (Paquete que contiene las herramientas de monitorización y gestión de conexiones entrantes)

cd /paquetes/ucspi-tcp-0.84

Si se modifican conf-cc y conf-ld en el paso de compilación de qmail también se necesitará hacer los mismos cambios en este directorio.

Luego ejecute:

make
make setup check

5 -INSTALACIÓN DAEMONTOOLS

URL – Descarga. http://cr.yp.to/daemontools/daemontools-0.76.tar.gz

URL – Documentación. http://cr.yp.to/daemontools.html

Nos introducimos dentro del directorio de administración:

cd /paquetes/admin.

Establecer la ubicación de destino. Por omisión, se instalan en /usr/local. Si quiere modificar esto, y usar otro directorio, edite ahora conf-home.

Compilar:

make

Ejecutar la Instalación:

make setup check

Estructura de ficheros

/var/qmail es la raíz de la estructura de ficheros de *qmail*. Esta configuración puede cambiarse al compilar *qmail*, pero es buena idea dejarlo sin cambios, de tal manera que otros administradores sepan dónde encontrar las cosas. Si realmente desea reubicar parte o todo el árbol de fuentes de *qmail*, es mejor que lo haga usando enlaces simbólicos. Para más detalles, véase la subsección Creación de directorios de la sección Instalación.

Los subdirectorios del nivel superior son:

Subdirectorios del nivel superior	
Directorio	**Contenido**
alias	ficheros .qmail para los alias del sistema
bin	binarios y guiones del programa
boot	guiones de inicio
control	ficheros de configuración
doc	documentación (excepto páginas man)
man	páginas man
queue	la cola de los mensajes por enviar
users	Los ficheros de base de datos de qmail-users

Estructura de la cola

El fichero INTERNALS (en http://www.es.qmail.org/documentacion/distro/text/ cuenta con traducciones de casi todos los ficheros de documentación de la distribución de qmail) en el directorio de compilación discute con detalle el encolamiento de mensajes. Esta es una visión más general de la estructura de la cola.

Subdirectorios de la cola	
Subdirectorio	**Contenido**
bounce	errores permanentes en la entrega
info*	direcciones de envoltura del remitente
intd	envolturas en construcción por parte de qmail-queue
local*	direcciones de destinatarios locales
lock	ficheros de bloqueo
mess*	ficheros con mensajes
pid	usado por qmail-queue para obtener un número de inodo
remote*	direcciones de remitentes remotas
todo	envolturas completas

Nota: los directorios marcados con * contienen una serie de subdirectorios de subdivisión (split) llamados 0, 1, ..., hasta (conf-split-1), en donde conf-split es un establecimiento de configuración en tiempo que está contenido en el fichero conf-split dentro del directorio de compilación. Por defecto su valor es 23. El propósito de dividir estos directorios es reducir el número de ficheros en un único directorio, para servidores muy cargados.

Los ficheros bajo el subdirectorio mess reciben el nombre a partir de su número de inodo. Esto quiere decir que no puede moverlos manualmente usando utilidades UNIX estándar como mv, dump/restore y tar. Hay algunas utilidades, contribución de los usuarios, que renombran los ficheros de la cola correctamente. Pueden encontrarse en http://www.qmail.org.

Nota: No es seguro modificar los ficheros de la cola mientras *qmail* se está ejecutando. Si desea modificar la cola, detenga primero *qmail*, trastee con cuidado en la cola, y luego reinicie *qmail*.

Ficheros de configuración

Todos los ficheros de configuración de sistema de *qmail*, con la excepción de los ficheros .qmail en ~alias, están en /var/qmail/control. La página man de qmail-control contiene una tabla como la que aparece con el nombre de «Ficheros de control».

Ficheros de control

Control	**Predeterminado**	**Usado por**	**Finalidad**
badmailfrom	ninguno	qmail-smtpd	From: en lista negra
bouncefrom	MAILER-DAEMON	qmail-send	nombre de usuario del remitente de la devolución
bouncehost	me	qmail-send	nombre de máquina del remitente de la devolución
concurrencylocal	10	qmail-send	máximo simultáneo de entregas locales
concurrencyremote	20	qmail-send	máximo simultáneo de entregas remotas
defaultdomain	me	qmail-inject	nombre dominio predeterminado
defaulthost	me	qmail-inject	nombre máquina predeterminado
databytes	0	qmail-smtpd	máximo número de bytes en el mensaje (0=sin límite)
doublebouncehost	me	qmail-send	nombre de mánquina del remitente de devolución doble
doublebounceto	postmaster	qmail-send	usuario que recibirá las dobles devoluciones
envnoathost	me	qmail-send	dominio predeterminado para direcciones sin arroba
helohost	me	qmail-remote	nombre de máquina usado en la orden SMTP HELO
idhost	me	qmail-inject	nombre de máquina para Message-ID's
localiphost	me	qmail-smtpd	nombre que sustituye a la dirección IP local
locals	me	qmail-send	dominios que entregamos localmente
me	FQDN del sistema	varios	predeterminado para varios ficheros de control
morercpthosts	ninguno	qmail-smtpd	base de datos secundaria para rcpthosts
percenthack	ninguno	qmail-send	dominios que pueden usar % en el relay (retransmisión)

plusdomain	me	qmail-inject	dominio por el que se sustituye el + anexado
qmqpservers	ninguno	qmail-qmqpc	direcciones IP de los servidores QMQP
queuelifetime	604800	qmail-send	segundos que un mensaje puede permanecer en la cola
rcpthosts	ninguno	qmail-smtpd	dominios para los que aceptamos correo
smtpgreeting	me	qmail-smtpd	mensaje de saludo SMTP
smtproutes	ninguno	qmail-remote	rutas SMTP artificiales
timeoutconnect	60	qmail-remote	segundos que se esperará una conexión SMTP
timeoutremote	1200	qmail-remote	segundos que se esperará a un servidor remoto
timeoutsmtpd	1200	qmail-smtpd	segundos que se esperará a un cliente SMTP
virtualdomains	ninguno	qmail-send	usuarios y dominios virtuales

/var/qmail/control/locals

Utilizado por *qmail-send*, diferencia que es lo que se ha de albergar dentro del servidor. Contiene los nombres de máquina considerados como **locales**, aquellos que se quedarán dentro del servidor en la cola *local*. Por defecto, esta variable tiene el mismo valor que el fichero de control me (su nombre de máquina).

```
mx.upsam.net
```

/var/qmail/control/me – Contiene el nombre del servidor de correo.

```
mx.upsam.net
```

defaulthost y defaultdomain

Los dos ficheros de control /var/qmail/control/defaulthost y /var/qmail/control/defaultdomain se revelan de gran utilidad. Permiten especificar la forma en la que *qmail-inject* completa las direcciones de destino incompletas. Así, una dirección que no contiene la parte del dominio será completada con defaulthost y una dirección en la que la parte del dominio no contenga punto se completará con defaultdomain. El valor por defecto es el mismo que el de me, y que no corresponde forzosamente al nombre de máquina que usted desearía para el correo saliente.

Las variables de entorno QMAILDEFAULTDOMAIN y QMAILDEFAULTHOST tienen prioridad sobre el contenido de los ficheros de control correspondientes.

Por defecto tenemos **/var/qmail/control/defaultdomain** y **/var/qmail/control/defaulthost** tiene el valor contenido del fichero **me**

Según nuestra configuración:

```
upsam.net
```

rcpthosts (Ver RELAY)

El fichero de control /var/qmail/control/rcpthosts lo utiliza *qmail-smtpd*. Permite controlar la retransmisión (relay) de correo por parte de la máquina vía SMTP.

Mientras el fichero rcpthosts exista, *qmail-smtpd* rechazará todo correo cuya parte del dominio del destinatario no figure en rcpthosts. Por defecto, es decir**, en ausencia de rcpthosts, *qmail-smtpd* acepta todos los correos. → MODO SERVIDOR ABIERTO, CUIDADO!!**

Según nuestra configuración:

```
mx.upsam.net
```

qmail-smtpd rechazará en este ejemplo todo correo que no tenga como destino el dominio mx.upsam.net

Por ejemplo:

jesussot@wanadoo.es , es rechazada por este servidor desde cualquier ordenador.

jesussot@mx.upsam.net, es aceptada por tener como destino el dominio mx.upsam.net

Quede claro que es muy importante que rcpthosts autorice al menos a las máquina que figuran en el fichero de control locals para que pueda recibir por SMTP los correos destinados a ellas.

rcpthosts es ignorado si la variable de entorno RELAYCLIENT está asignada. En ese caso, *qmail-smtpd* añade su contenido a la dirección de destino de los correos que pasen por él.

El control de retransmisión (relay) por rcpthosts se asocia a menudo a un control en el nivel de TCP mediante un envoltorio o «wrapper». Nuestro envoltorio es TCPSERVER, en la sección de RELAY lo trataremos.

badmailfrom

badmailfrom es muy útil para evitar el correo entrante no solicitado. Permite especificar las direcciones de remitentes prohibidas. *qmail-smtpd* rechazará todo correo que proceda de ellas. Véase este ejemplo:

```
@contaminacion.com
contaminador@basura.com
```

qmail-smtpd rechazará todo correo que proceda del dominio contaminacion.com o de la dirección contaminador@basura.com.

smtproutes

El fichero de control /var/qmail/control/smtproutes permite especificar los puntos de retransmisión (relay) SMTP con los que *qmail-remote* debará contactar para encaminar los correos salientes. Véase la página de manual para más detalles. He aquí un ejemplo:

```
primer.dominio:correo.primer.dominio
segundo.dominio:correo.segundo.dominio:24
:correo.otro.dominio
```

Los correos con destino a premer.dominio se transmitirán por SMTP al puerto TCP 25 de correo.primer.dominio, los que tengan destino a segundo.dominio se transmitirán por SMTP al puerto 24 de correo.segundo.dominio. En cuanto al resto de los correos, se transmitirán al puerto TCP 25 de correo.otro.dominio.

qmail-remote utiliza asimismo los ficheros de control helohost, timeoutconnect y timeoutremote; véase la página de manual de *qmail-remote.*

aliasempty

El primer fichero de control que deseará modificar es sin duda /var/qmail/control/aliasempty. Contiene el nombre de la BAL en la cual el agente de entrega de correo (MDA) *qmail-local* entrega el correo cuando no existe el fichero .qmail. Su valor por defecto es ./Mailbox.

Esta regla de entrega por defecto se le puede, sin embargo, especificar a *qmail-start* en la línea de órdenes en el momento de ejecutar *qmail*; véase la página de manual. Se transmitirá a *qmail-lspawn*.

Vista la estructura de directorios y los ficheros de configuración procedemos a iniciar QMAIL

Hemos de crear el script de arranque: /var/qmail/rc

El directorio /var/qmail/boot contiene ejemplos de guiones de arranque de *qmail* para diferentes configuraciones: /var/spool/mail frente a $HOME/Mailbox, uso de procmail o de dot-forward, y varias combinaciones de ambos. Tómese la libertad de examinarlos, pero en este caso y para nuestra instalación, usaremos los siguientes:

vi /var/qmail/rc

```
#!/bin/sh

# Usamos la salida estándar para llevar un registro
# Usamos el control/defaultdelivery de qmail-local como forma predeterminada
# para entregar mensajes

exec env - PATH="/var/qmail/bin:$PATH" \
qmail-start "`cat /var/qmail/control/defaultdelivery`"
```

Utilice su editor para crear el archivo anterior /var/qmail/rc, y luego ejecute las siguientes órdenes:

chmod 755 /var/qmail/rc
mkdir /var/log/qmail

Llegados a este punto, tendremos que decidir el modo predeterminado de entrega para los mensajes que no se entregan según las instrucciones de un fichero **.qmail**. La siguiente tabla apunta algunas de las opciones más comunes:

Formato Buzón	**Nombre**	**Localización**	**Entrega Por Defecto**	**Comentarios**
Mbox	Mailbox	$HOME	./Mailbox	lo más común, lo soportan más clientes
maildir	Maildir	$HOME	./Maildir/	Más fiable, e idoneo para

				qmail-pop3
Mbox	nombreusuario	/var/spool/mail	Ver INSTALL.vsm	Buzón tradicional UNIX

Véase http://www.es.qmail.org/documentacion/distro/text/INSTALL.mbox.php3, http://www.es.qmail.org/documentacion/distro/text/INSTALL.maildir.php3, e http://www.es.qmail.org/documentacion/distro/text/INSTALL.vsm.php3 para más información.

Para seleccionar su tipo de buzón por defecto, introduzca el valor "Entrega Por Defecto" de la tabla en /var/qmail/control/defaultdelivery. Por ejemplo para seleccionar la entrega estándar de *qmail* en Maildir, introduzca:

echo ./Maildir/ >/var/qmail/control/defaultdelivery

Nota: defaultdelivery no es un fichero de control estándar de *qmail*. Es una característica del fichero /var/qmail/rc visto más arriba.

Maildir vs Mailbox. Defaultdelivery

Para el almacenamiento de mensajes recibidos, es necesario establecer un formato de buzón que reconocerá nuestro servidor de POP3. Por defecto, qmail trabaja con el formato **Maildir**, que almacena cada mensaje en un archivo independiente, ubicado en el directorio /HOME$/Maildir . También es posible usar el formato clásico de Unix, Mailbox, que almacena todos los mensajes en un solo fichero.

El formato del mensaje se indica como hemos visto anteriormente en el fichero **/var/qmail/control/defaultdelivery**

Ejecutamos el SCRIPT

/var/qmail/rc &

Vemos los procesos en ejecución: ps –e | grep qmail

17424 pts/1 00:00:00 qmail-send

17426 pts/1 00:00:00 qmail-lspawn

17427 pts/1 00:00:00 qmail-rspawn

17428 pts/1 00:00:00 qmail-clean

Hasta ahora tenemos los demonios gestores de colas y de entrega activos, ahora comprobamos la correcta instalación inyectando un mensaje participando como MUA en la comunicación.

RELAYING (RETRANSMISIÓN)

Introducción

¿Qué es el relaying o retransmisión? Sucede cuando un MTA acepta un mensaje por SMTP que no es para una dirección local, ni proviene de un remitente **local**.

En los días anteriores al envío masivo de correo no solicitado, era común que los MTA (agentes de transporte de correo) se configuraran para el reenvío abierto **(*open relay*)**. Eran servidores promiscuos, que aceptaban correo de cualquiera, hacia cualquiera.

La mayor parte de los MTA de hoy día están configurados para desactivar completamente el relay, o para permitir que sólo ciertos usuarios de confianza, o sistemas de confianza, utilicen el MTA como medio de reenvío.

Desactivación de la retransmisión

Si se siguen las instrucciones oficiales de instalación de *qmail*, el reenvío o relay estará desactivado por defecto. Ello se consigue completando el fichero **/var/qmail/control/rcpthosts** con los nombres de dominio completos y calificados que se listan en locals y virtualdomains (los servidores locales). El nombre del fichero de control, rcpthosts, proviene de la orden RCPT (destinatario) del protocolo SMTP. En una sesión SMTP, RCPT se usa para especificar la dirección de los destinatarios de un mensaje. rcpthosts listará, pues, los nombres de servidor válidos que pueden aparecer en una dirección RCPT.

Retransmisión selectiva (*selective relaying*)

La mayor parte de los servidores pueden desactivar completamente el reenvío, pero si tiene usted que dar soporte a una comunidad de usuarios desperdigada, necesitará una manera de permitir que sus usuarios, y **sólo** sus usuarios, usen su sistema como relay o retransmisor. Esto se consigue usando **tcpserver** para activar la variable de entorno RELAYCLIENT que le dice a qmail-smtpd que ignore el fichero rcpthosts.

Si sigue las instrucciones de instalación de este documento, el reenvío selectivo estará activado por defecto. Para proporcionar a un cliente acceso al reenvío, añada una entrada como la siguiente a /etc/tcp.smtp:

Dirección_IP_del_cliente:allow,RELAYCLIENT=""

Luego reconstruya la base de datos de acceso SMTP ejecutando:

```
tcprules /etc/tcp.smtp.cdb /etc/tcp.smtp.tmp < /etc/tcp.smtp
chmod 644 /etc/tcp.smtp
```

Creamos el SCRIPT de Ejecución del QMAIL-SMTPD

mkdir /var/qmail/supervise
mkdir /var/qmail/supervise/qmail-smtpd
mkdir /var/qmail/supervise/qmail-pop3d

/var/qmail/supervise/qmail-smtpd/run

```
#!/bin/sh
QMAILDUID=`id -u qmaild`
NOFILESGID=`id -g qmaild`
exec /usr/local/bin/softlimit -m 2000000 \
   /usr/local/bin/tcpserver -v -p -x /etc/tcp.smtp.cdb \
   -u $QMAILDUID -g $NOFILESGID 0 smtp /var/qmail/bin/qmail-smtpd 2>&1
```

Damos permisos de ejecución: **chmod 755 run**

creamos las reglas de filtrado del TCPSERVER

Creamos el fichero:

vi /etc/tcp.smtp

```
127.:allow, RELAYCLIENT=""
192.168.25.:allow, RELAYCLIENT=""
:allow
```

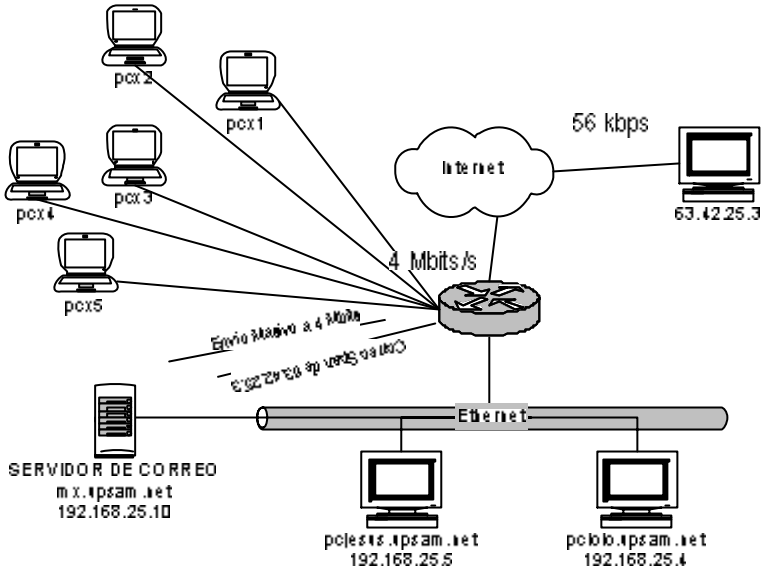
Guardamos el fichero, y ejecutamos:

tcprules tcp.smtp.cdb tcp.smtp.tmp < tcp.smtp

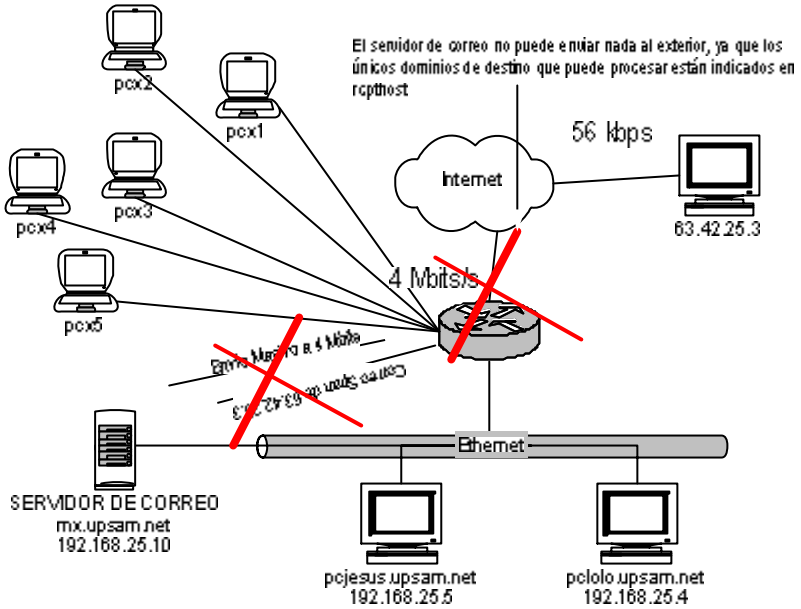
chmod 644 /etc/tcp.smtp

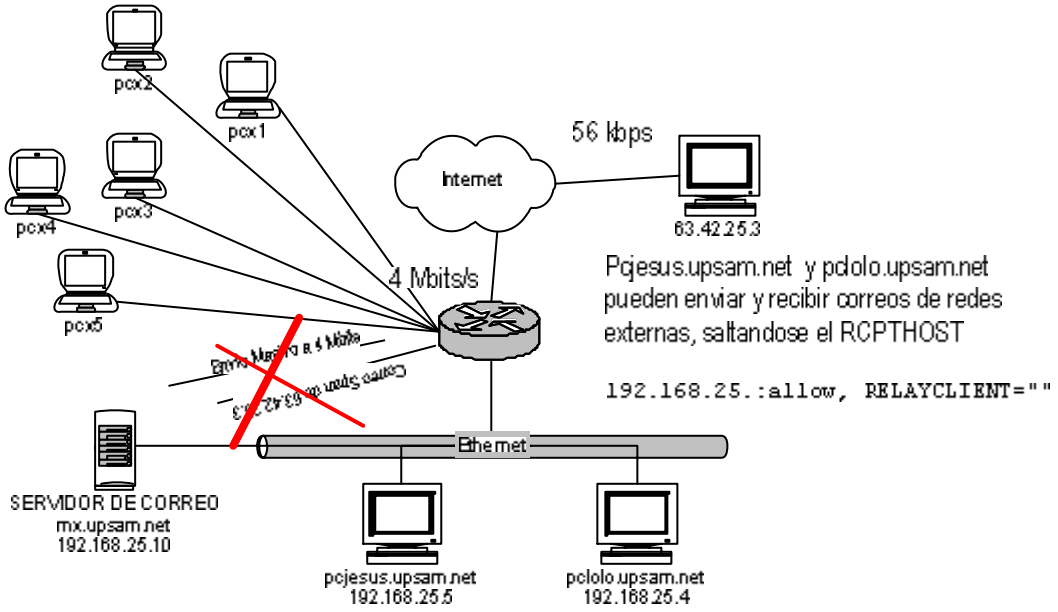
CON TCPSERVER EVITAMOS EL ENVIO DE CORREO POR PARTE DE 3ºs A OTRAS ENTIDADES GRACIAS A LAS REGLAS INTRODUCIDAS EN tcp.smtp.cdb

SIN /VAR/QMAIL/CONTROL/RCPTHOST – SERVIDOR EN MODO ABIERTO.


pcx2
pcx1
pcx4
pcx3
pcx5
56 kbps
Internet
63.42.25.3
4 Mbits/s
Ethernet
SERVIDOR DE CORREO
mx.tpsam.net
192.168.25.10
pcjesus.tpsam.net
192.168.25.5
pcjolo.tpsam.net
192.168.25.4

SIN REGLAS EN TCPSERVER Y CON RCPHOST





Desactivar el MTA SENDMAIL si existe.

Si bien es posible ejecutar simultáneamente *qmail* y el MTA existente, que probablemente sea Sendmail, no lo recomiendo a menos que sepa lo que está haciendo. Y francamente, si está leyendo estas líneas, entonces es que no sabría lo que estaba haciendo :-)

Si su MTA existente es Sendmail podrá detenerlo ejecutando el guión de inicio con el argumento stop. Por ejemplo, alguno de los siguientes debe funcionar:

/etc/init.d/sendmail stop
/sbin/init.d/sendmail stop
/etc/rc.d/init.d/sendmail stop

Si no encuentra un guión de inicio de init.d/sendmail, puede localizar el PID de sendmail's PID usando ps -ef | grep sendmail o bien ps waux | grep sendmail y detenerlo utilizando:

kill PID-de-sendmail

Si su MTA no es Sendmail compruebe la documentación para el correcto procedimiento de detención.

También debería pensar en eliminar completamente de su sitema el MTA anterior. Al menos, desactive el guión init.d de manera que no intente arrancar de nuevo cuando el sistema reinicie.

Sustituya cualquier /usr/lib/sendmail existente con la versión de *qmail*:

mv /usr/lib/sendmail /usr/lib/sendmail.old # ignore los errores
mv /usr/sbin/sendmail /usr/sbin/sendmail.old # ignore los errores
chmod 0 /usr/lib/sendmail.old /usr/sbin/sendmail.old # ignore los errores
ln -s /var/qmail/bin/sendmail /usr/lib
ln -s /var/qmail/bin/sendmail /usr/sbin

Arrancamos nuestro servicio de SMTP – QMAIL-SMTPD sobre TCPSERVER

/var/qmail/supervise/qmail-smtpd/run **&**

Vemos si el proceso esta activo y escaneamos el puerto

ps –e | grep tcpserver

nmap localhost → Puerto 25 abierto.

Alias

En el sistema *qmail*, un usuario de nombre x (En nuestra explicación usaremos "login", puede ser cualquier otro nombre) controla todas las direcciones de correo de formato "login"-* donde la estrella representa una cadena cualquiera, formada por caracteres válidos para una dirección. Esta regla general puede alterarse por la colocación de un fichero de alias.

Para cada dirección login-loquesea, el usuario puede decidir crear el fichero ~login/.qmail-loquesea que contendrá las instrucciones que habrá de seguir *qmail-local*, el agente de entrega de correo (MDA) de *qmail*. Por convención, el fichero ~login/.qmail corresponde a la dirección corta login. Además, el fichero especial ~login/.qmail-default, cuando existe, tiene la función de fichero por defecto.

En ausencia de los ficheros ~login/.qmail y ~login/.qmail-default, *qmail-local* entregará el correo con destino a ~login siguiendo la regla de entrega por defecto especificada en el momento de arrancar *qmail-send*; véase the section called *aliasempty*. En general, esta entrega se hace en el buzón de correo, de formato mbox, llamado ~login/Mailbox pero esto queda a la elección del administrador.

Cuestión totalmente distinta es la entrega de los correos con destino login-loquesea. ***qmail-local* no entregará los correos con destino login-loquesea si no encuentra ni el fichero ~login/.qmail-loquesea ni ~login/.qmail-default en el momento de la entrega.** De hecho, este correo será interceptado por el usuario especial alias si su fichero ~alias/.qmail-default existe, o en caso contrario lo devolverá al remitente.

Los ficheros ~login/.qmail-* tienen todos la misma sintaxis.

Notas sobre *QMAIL-LOCAL*

Reiteramos aquí que es necesario prestar mucha atención a los derechos de acceso a los directorios personales de los usuarios. En efecto, ***qmail-local* se negará a entregar el correo de un usuario cuyo directorio personal tenga permisos de escritura para su grupo o para todo el mundo.** Lo mismo se aplica para el bit t «sticky». La misma apreciación es válida para los ficheros .qmail*. Véase la página de manual *dot-qmail*.

Hay que señalar asimismo que *qmail-local* convierte todas las mayúsculas en minúsculas en los nombres de fichero .qmail-*. Convierte asimismo el carácter punto en carácter dos puntos y a la inversa. Así, para que el correo con destino a jesus.soto se entregue al usuario tambor, basta con que tambor tenga un .qmail-jesus.soto en su directorio personal ~jesussoto

- **Postmaster**:. Usted no es un ciudadano de Internet si esta dirección no funciona. Bastará con hacer un touch (y chmod 644) ~alias/.qmail-postmaster; todo mensaje remitido a Postmaster será entregado en **~alias/Mailbox.**

- **MAILER-DAEMON:** No es un requisito, pero a veces los **usuarios responden a mensajes de devolución.** Haga un touch (y chmod 644) ~alias/.qmail-mailer-daemon

- Bajo qmail, **root nunca recibe correo**. Su sistema puede que genere mensajes para root cada noche; si no establece un alias para root, dichos mensajes serán devueltos. (Acabarán siendo devueltos doblemente a postmaster). Establezca un alias para root en ~alias/.qmail-root. Los ficheros .qmail son similares a los .forward, tenga en cuenta no obstante que están orientados a líneas; ver dot-qmail.0 para más información.

- Otras cuentas de usuarios no reales. Bajo qmail, las cuentas de usuarios no reales no reciben correo; usuario real significa todo aquel usuario que sea dueño de ~cuenta. Establezca alias para toda cuenta de usuario no real que deban recibir correo.

  Tenga en cuenta que cuentas especiales como ftp, www y uucp deben contar con sus respectivos directorios bajo la propiedad de root.

- **Default**. Si lo desea, cree con un touch ~alias/.qmail-default el fichero de control para capturar todo lo demás. Tenga en cuenta que también atrapará errores de tecleado y direcciones que probablemente debieran ser devueltas. No atrapará direcciones que comiencen por un nombre de usuario; dicho usuario puede establecer su propio ~usuario/.qmail-default.

El fichero de ALIAS /var/qmail/users/assign

El sistema *qmail* puede poseer una lista de alias, compilada en formato CDB y almacenada en el fichero /var/qmail/users/cdb, y cuya versión ASCII se almacena en el fichero /var/qmail/users/assign. La orden **qmail-newu** permite transformar la versión ASCII en versión compilada CDB. Esta última es leída por *qmail-lspawn* antes de la llamada a *qmail-local.*

La versión ASCII se asemeja a un fichero passwd. Ver la página de manual *qmail-users* para una descripción de su formato. La orden **qmail-pw2u** permite por otra parte construir la base de datos /var/qmail/users/assign a partir de su fichero /etc/passwd :

En el fichero /var/qmail/users/assign tenemos reflejado el filtro inicial realizado a las direcciones de correo. los usuarios que poseen correo y que dirección es la aceptada.

```
Root# qmail-pw2u < /etc/passwd > /var/qmail/users/assign
root# qmail-newu
```

El fichero de alias compilado es leído por *qmail-lspawn*. Esta última aplicación lo utiliza para determinar qué usuario recibirá la entrega de un correo mediante *qmail-local* en modo de usuario. *qmail-local* se encarga de la lectura y de la interpretación de los ficheros .qmail*. Es preciso comprender que la lista de alias es leída por *qmail-lspawn* mientras que los .qmail* son leídos por *qmail-local.* Los dos mecanismos no son mutuamente excluyentes, sino sucesivos. El uno opera en UID 0; el otro opera en modo de usuario.

Consideremos por ejemplo un correo con destino local que salga de la cola de espera. Ese correo se trasmite a *qmail-lspawn* por su padre el demonio *qmail-send*. *qmail-*

lspawn, que opera en UID 0, determina, al leer el fichero de alias compilado, el nombre de usuario usuario, el UID uid, el GID gid y el fichero ~usuario/.qmail* en virtud de los cuales es preciso hacer la entrega por *qmail-send*. *qmail-send* se ejecuta entonces con UID uid y efectúa la entrega de acuerdo con el fichero ~usuario/.qmail* y con las reglas explicadas en the section called *Entrega mediante qmail-local y usuario alias* in Chapter 4.

```
qmail-send
  \_ qmail-lspawn UID 0
     \_ qmail-local UID uid
```

Fichero de ejemplo **/var/qmail/users/assign**

=jesussoto:jesussoto:503:505:/home/jesussoto:::
+jesussoto-:jesussoto:503:505:/home/jesussoto:-::

El archivo assign

Este archivo contiene TODOS los usuarios que recibirán correo electrónico, usted deberá generar este archivo bajo el directorio /var/qmail/users , su composición es muy similar al archivo /etc/passwd , observe:

```
=informes:daniel:231:121:/home/daniel:::
```

Por decirlo de cierta forma Qmail posee **dos tipos de filtros** (aunque no son filtros explicitamente): archivos .qmail y el archivo assign .

Los archivos .qmail se explicarán posteriormente pero: si existe uno por nombre .qmail-xxx será recibido el mensaje para xxx de otra forma será rechazado, sin embargo, sería ilogico pensar que todos los usuarios fueran definidos .qmail-usuario1, .qmail-usuario-2, ... , el archivo assign ofrece este "filtro" previo a llegar a los archivos .qmail .

Una linea del archivo assign sería la siguiente:

```
=informes:daniel:231:121:/home/daniel:::
```

Esta linea significa que cualquier correo electronico destinado para informes sera enviado al Buzón del usuario daniel con UID=231, GID=121 que se encuentra en el directorio /home/daniel , también es posible indicar una extensión abierta para el destinatario ("wildcard"):

```
+web:jose:234:342:/home/jose:::
```

La linea anterior indica que todo correo **cuyo nombre empieze con web** será enviado al buzón del usuario jose en el directorio /home/jose , esto seria: webmaster@, web-hosting@, webservicio@ ,etc...Un archivo assign básico sería el siguiente:

```
+web:jose:234:342:/home/jose:::
=web:admin:121:23:/home/admin:::
```

```
.
```

Debe terminar en punto este archivo. **La secuencia de eventos en Qmail para este archivo seria:**

1. Si llega un mensaje para web@ este será enviado al buzón del usuario admin
2. Cualquier mensaje que inicie en webxxxx@ (excluyendo web@) será enviado al buzón del usuario jose
3. Todo mensaje que no entre dentro de estos criterios será enviado al buzón del usuario alias
4. **NOTA:** Esta es solo la primera etapa de "filtros",la recepción final dependerá de los archivos .qmail presentes en cada directorio del usuario.

Cada vez que sea agregado un usuario al archivo assign **es necesario ejecutar el comando qmail-newu** ubicado en el directorio /var/qmail/bin , esto generará un archivo binario con el contenido de assign , la intención de este archivo es agilizar el acceso de información a Qmail, este archivo binario se llama cdb y se encuentra en el directorio /var/qmail/users

Las asignaciones anteriores en el archivo assign asumieron dos cosas:

1. Todos los mensajes estan dirigidos a un solo dominio, aquel definido en el archivo /var/qmail/control/defaultdomain, esto es, si solo se define un nombre (web:), Qmail asume la dirección web@<defaultdomain>
2. Todos los usuarios que reciben mensajes deben tener cuentas Unix en el "Host" para poder tener un buzón propio

Configuracion para MAILDIR

Primeramente se debe crear el directorio del tipo Maildir que substituira el uso de Mailbox, para esto se utiliza el comando **maildirmake** ubicado en /var/qmail/bin . De la misma en que es generado el archivo Mailbox en el directorio raiz ("Home Directory") del usuario, se debe ejecutar el comando maildirmake dentro del directorio raiz del usuario ("Home Directory").

Si el usuario admin tiene su directorio raiz ("Home Directory") en /home/admin , entonces dentro de este directorio se debe ejecutar maildirmake Maildir , este comando generará un directorio llamado Maildir que contendrá tres subdirectorios cur, new, tmp .

Ahora se debe indicarle a Qmail que el usuario empezará a recibir sus mensajes en el formato Maildir , esto se realiza agregando la siguiente linea al archivo .qmail del usuario:

```
./Maildir/
```

Lo anterior indica que todo mensaje será colocado dentro del directorio Maildir .

Si se desea empezar a utilizar este formato para recibir los mensajes (algo muy recomendable) para todos los usuarios del Sistema, se deben modificar los parametros "default" de Qmail, estos se encuentran en el archivo /var/qmail/rc , basta cambiar la linea: qmail-start ./Mailbox splogger qmail por

```
qmail-start ./Maildir/ splogger qmail
```

Si opta por este esquema se recomienda agregar el directorio Maildir a /etc/skel (Vea Administración de usuarios en Unix)

Consideraciones de MAILDIR

A pesar de las ventajas que presenta **Maildir** , presenta un pequeño problema de configuración; debido al amplio uso del formato **Mailbox** algunos Mailers (MUA "Mail User Agents") del mercado no son capaces de accesar los mensajes en este formato *directamente* , sin embargo, Qmail ofrece opciones que simulan el formato **Mailbox** que comunmente espera un Mailer.

Ficheros .QMAIL

La entrega del correo de un usuario normalmente está controlada por uno o más ficheros .qmail (pronunciar punto qmail). Son ficheros en el directorio del usuario, cuyos nombres comienzan con .qmail. La página de manual de dot-qmail describe el uso de los ficheros .qmail.

Los ficheros .qmail contienen una lista de instrucciones de entrega, una instrucción por línea. El primer carácter de la línea determina de qué tipo de entrega se trata:

Instrucciones de entrega

Carácter	**Tipo de entrega**	**Valor**
#	ninguna (comentario)	ignorado
|	programa	orden que ejecuta intérprete de órdenes
/ o bien **.**	mbox (si el último carácter no es /)	ruta de mbox (incluyendo / o .)
/ o bien **.**	maildir (si el último carácter es /)	ruta de maildir (incluyendo / o .)
&	reenvío	dirección para el reenvío
letra o número	reenvío	dirección para el reenvío (incluyendo primer carácter)

Entrega a un programa

Cuando se encuentra una instrucción de entrega mediante un programa, *qmail* inicia el intérprete de órdenes (/bin/sh) para ejecutar la orden, y le pasa a la orden, por entrada estándar, una copia del mensaje recibido. La página man qmail-command documenta los detalles del proceso.

La entrega a programas es muy potente, y puede usarse para implementar una amplia gama de funciones, como el filtrado de mensajes, respuesta automática a mensajes y entrega mediante Agentes de Entrega de Correo como procmail.

Por ejemplo:

|preline /usr/ucb/vacation djb

Hace que *qmail* inicie preline, le pasa como parámetros el programa /usr/ucb/vacation y djb, y le proporciona una copia del mensaje por entrada estándar.

Entrega en MBOX

mbox es la forma de *qmail* de referirse a el formato estándar de unix, mailbox, por el que se almacenan múltiples mensajes en un único fichero, y en el que los mensajes van encabezados por una línea From. Esta línea parece un campo de encabezado, pero no lo es: sólo es lo que el agente de entrega de correo añade para que los agentes de usuario de correo (programas clientes de correo) puedan averiguar dónde comienza cada mensaje.

Por ejemplo:

./Mailbox

Esta línea hará que los mensajes se anexen a $HOME/Mailbox, con la adición de una línea previa From . Un «buzón de correo» mbox con un único mensaje presenta este aspecto:

From usuario1@ejemplo.net Thu May 13 18:34:50 1999
Received: (qmail 1287205 invoked from network); 13 May 1999 18:34:49 -0000
From: usuario1@ejemplo.net
To: usuario2@proveedor.com
Subject: Buenas
¿Qué tal?

La primera línea la añadió *qmail* en la entrega.

Entrega en MAILDIR

maildir es un formato de buzón de correo creado por Dan Bernstein para obviar las limitaciones del formato mbox. Un buzón maildir es un directorio que contiene tres subdirectorios, new, cur y tmp.

Cada mensaje en un buzón de tipo maildir se guarda en archivos separados dentro de uno de los subdirectorios, dependiendo de sus estado: new es para correo sin leer, cur es para mensajes que ya se han leído, y tmp es para mensajes que se están entregando. La página man de maildir describe en detalle el formato de maildir.

Una de las ventajas del formato maildir es que, aunque no utiliza el bloqueo para prevenir actualizaciones simultáneas desde diferentes agentes de entrega de correo, sin embargo es fiable. Esto quiere decir que los buzones de maildir pueden alojarse con seguridad en sistemas de ficheros montados mediante NFS (sistema de ficheros de red)

Por ejemplo:

./Maildir/

Hará que los mensajes se guarden en $HOME/Maildir, un buzón de correo de formato maildir.

Nota: los buzones de correo maildir deben crearse con el programa maildirmake que se incluye con *qmail.* Por ejemplo, maildirmake ~/Maildir.

Entrega en reenvío

El reenvío de una entrega consigue que el mensaje redireccionado se reenvíe a la dirección especificada. Las direcciones se especifican en los ficheros .qmail y deben ser direcciones completamente cualificadas. No pueden contener campos de comentario ni espacios extra.

Éstos son **incorrectos**:

&<usuario@proveedor.com>
& usuario@proveedor.com
&Jorge Usuario <usuario@proveedor.com>
&usuario

Éstos son correctos:

&usuario@proveedor.com
usuario@proveedor.com

Ambos provocan que usuario@proveedor.com reciba una copia del mensaje.

Crear la estructura de nuestro buzón para los usuarios de nuestro servidor de correo

echo ./Maildir/ > /etc/skel/.qmail
/var/qmail/bin/maildirmake /etc/skel/Maildir

CREAMOS NUESTRO SCRIPT PARA AÑADIR UN USUARIO A NUESTRO SISTEMA:

vi /var/qmail/nuevousuario
chmod 755 /var/qmail/nuevousuario

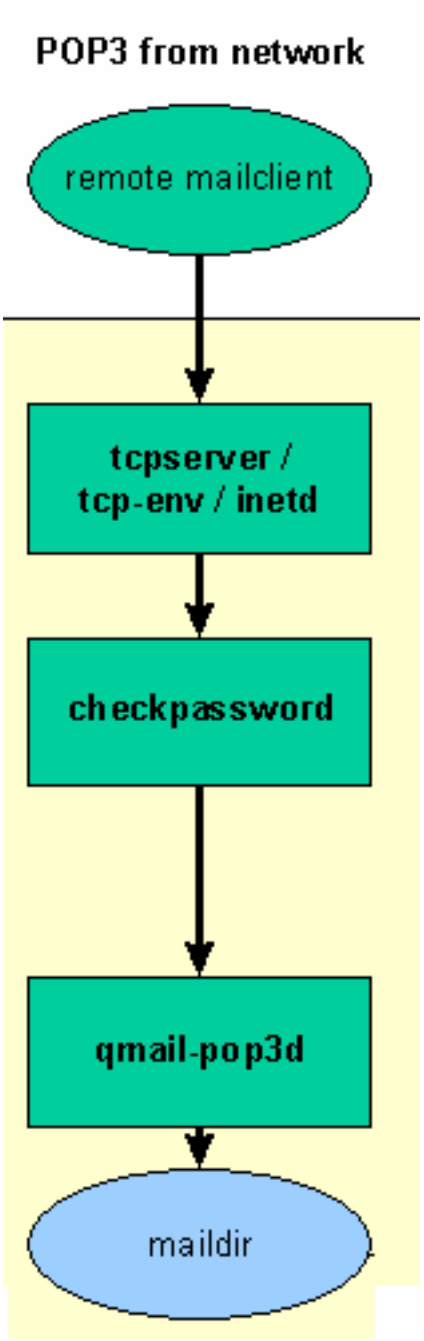
```
#!/bin/sh

LUSER=`wc /var/qmail/users/assign | awk '{print $1}'`
useradd $1 -s /sbin/nologin
passwd $1
ULTUID=`cut -f 3 -d ":" /etc/passwd | tail -1`
ULTGID=`cut -f 4 -d ":" /etc/passwd | tail -1`
rm /var/qmail/users/assign.tmp
echo Entradas en /var/qmail/users/assign:$LUSER
head -`expr $LUSER - 1` /var/qmail/users/assign > /var/qmail/users/assign.tmp
echo =$1:$1:$ULTUID:$ULTGID:/home/$1::: >> /var/qmail/users/assign.tmp
```

```
echo . >> /var/qmail/users/assign.tmp
rm /var/qmail/users/assign
cp /var/qmail/users/assign.tmp /var/qmail/users/assign
/var/qmail/bin/qmail-newu
```

Hacemos pruebas con los Relays para ver si funcionan correctamente

QMAIL-POP3D



QMAIL-POP3D

Nos proporciona los mecanismos para la manipulación de buzones de correo. Habilita el servidor POP (Ver RFC 1939) , dejando abierto el puerto 110.

Con un cliente de correo (Outlook, Evolution) podemos conectarnos a el y manipular nuestro correo.

La validación de un usuario se realiza con checkpassword usando los usuarios del sistema.

Si hemos arrancado el MTA para que albergue los mensajes en el directorio Maildir, el sevicio de qmail-pop3d busca:

1º- El fichero **.qmail** dentro del directorio raiz del usuario $HOME

2º- Procesa la regla indicada. Lo normal es que albergue la regla de entrega bajo el directorio **./Maildir/**

Analizaremos el mecanismo básico de manipulación de mensajes POP, bajo QMAIL.

CHECKPASSWORD

checkpassword proporciona una interfaz de comprobación de contraseñas simple, y uniforme para todas las aplicaciones raíz. Se puede utilizar con aplicaciones tales como login, ftpd y pop3d.

Existen herramientas compatibles con *checkpassword* que permiten bases de datos de contraseñas alternativas, *login*s secretos, contraseñas largas, subcuentas, contraseñas de un sólo uso, registro de uso detallado, y muchas otras prestaciones. Toda aquella aplicación que utilice la interfaz checkpassword funcionarán con todas estas herramientas. Algunas de ellas han sido diseñadas específicamente para soportar servidores POP masivos.

Instalación de CHECKPASSWORD

1- DESCARGAMOS EL PAQUETE:

http://cr.yp.to/checkpwd/checkpassword-0.90.tar.gz

2- LO DESEMPAQUETAMOS:

```
tar -xzvf checkpassword-0.90.tar
cd checkpassword-0.90
```

3- COMPILAMOS

```
make
```

4- INSTALAMOS: Como usuario root, instalamos checkpassword en /bin/checkpassword ejecutando:

```
make setup check
```

Crear SCRIPT de inicio de QMAIL-POP3D

mkdir /var/qmail/supervise/qmail-pop3d

vi /var/qmail/supervise/qmail-pop3d/run

```
#!/bin/sh
tcpserver -v -R 0 pop3 /var/qmail/bin/qmail-popup FQDN \
      /bin/checkpassword /var/qmail/bin/qmail-pop3d Maildir 2>&1 | \
      /var/qmail/bin/splogger pop3d &
```

chmod 755 /var/qmail/supervise/qmail-pop3d/run

Arrancar QMAIL-POP3D

/var/qmail/supervise/qmail-pop3d/run &

KILL QMAIL-POP3D

kill -9 [pid tcpserver qmail-pop3d]

Ficheros de inicio del sistema

Si tuviera que ejecutar manualmente el guión /var/qmail/rc, *qmail* sólo se iniciaría en parte. Pero queremos que *qmail* se inicie automáticamente cada vez que el sistema arranque, y que *qmail* se pare cada vez que el sistema se detenga.

Esto se consigue creando un guión de inicio/parada como el siguiente:

```
#!/bin/sh

PATH=/var/qmail/bin:/usr/local/bin:/usr/bin:/bin
export PATH

case "$1" in
  start)
    echo -n "Iniciando qmail: svscan"
    cd /var/qmail/supervise
    env - PATH="$PATH" svscan &
    echo $! > /var/run/svscan.pid
    echo "."
    ;;

  stop)
    echo -n "Deteniendo qmail: svscan"
    kill `cat /var/run/svscan.pid`
    echo -n " qmail"
    svc -dx /var/qmail/supervise/*
    echo -n " logging"
    svc -dx /var/qmail/supervise/*/log
    echo "."
    ;;

  stat)
    cd /var/qmail/supervise
    svstat * */log
    ;;

  doqueue|alrm)
    echo "Enviando una señal ALRM a qmail-send."
    svc -a /var/qmail/supervise/qmail-send
    ;;

  queue)
    qmail-qstat
    qmail-qread
    ;;

  reload|hup)
    echo "Enviando una señal HUP a qmail-send."
    svc -h /var/qmail/supervise/qmail-send
    ;;
```

```
 pause)
   echo "Congelando qmail-send"
   svc -p /var/qmail/supervise/qmail-send
   echo "Congelando qmail-smtpd"
   svc -p /var/qmail/supervise/qmail-smtpd
   ;;

 cont)
   echo "Reanudando qmail-send"
   svc -c /var/qmail/supervise/qmail-send
   echo "Reanudando qmail-smtpd"
   svc -c /var/qmail/supervise/qmail-smtpd
   ;;

 restart)
   echo "Reiniciando qmail:"
   echo "* Deteniendo qmail-smtpd."
   svc -d /var/qmail/supervise/qmail-smtpd
   echo "* Enviando a qmail-send la señal SIGTERM y reiniciando."
   svc -t /var/qmail/supervise/qmail-send
   echo "* Reiniciando qmail-smtpd."
   svc -u /var/qmail/supervise/qmail-smtpd
   ;;

 cdb)
   tcprules /etc/tcp.smtp.cdb /etc/tcp.smtp.tmp < /etc/tcp.smtp
   chmod 644 /etc/tcp.smtp*
   echo "Releído /etc/tcp.smtp."
   ;;

 help)
   cat << HELP

stop    --      detiene el servicio de correo (conexiones smtp rehusadas, nada
                sale afuera)

start   --      inicia el servicio de correo (conexiones smtp aceptadas, el
                correo puede salir)

pause   --      congela temporalmente el servicio de correo (se aceptan
                conexiones, pero no sale nada afuera)

cont    --      continúa con el servicio de correo antes congelado

stat    --      muestra el estado del servicio de correo

cdb     --      reconstruye el fichero cdb de tcpserver para smtp

restart --      detiene y reinicia smtp, envía a qmail-send una señal TERM y lo
                reinicia
```

```
doqueue --      envía a qmail-send una señal de ALRM, reprogramando los
                mensajes salientes para su entrega

reload  --      envía a qmail-send una señal de HUP, leyendo de nuevo locals y
                virtualdomains

queue   --      muestra el estado de la cola de correo

alrm    --      lo mismo que doqueue

hup     --      lo mismo que reload
HELP
   ;;
 *)
   echo "Uso: $0 {start|stop|restart|doqueue|reload|stat|pause|cont|cdb|queue|help}"
   exit 1
   ;;
esac

exit 0
```

Este guión también puede conseguirse por web en http://Web.InfoAve.net/~dsill/qmail-script-dt61.txt.

Nota: Si encuentra que *qmail* se detiene poco después de reiniciar el sistema, puede anteponer la orden supervise en la sección de start del guión con nohup. Por ejemplo:

```
nohup env - PATH="$PATH" svscan &
```

Cree el guión usando su editor de texto u obténgalo de Internet con su navegador y luego instálelo en el directorio init.d de su sistema, que debería estar en una de las localizaciones siguientes:

- /etc/init.d
- /sbin/init.d
- /etc/rc.d/init.d

Llame al guión *qmail*. También tendrá que hacer un vínculo simbólico al guión en algunos de los directorios rc. Estos directorios se nombran rcN.d, donde N es el nivel de ejecución (*runlevel*) al que se aplican. Las interioridades del árbol de directorios del inicio quedan más allá de la finalidad de este documento. Si no le bastan estas instrucciones simplificadas, consulte la documentación de sus sistema. Su directorios rc estarán probablemente en uno de estos sitios:

- /etc
- /sbin
- /etc/rc.d

Para crear los vínculos simbólicos, ejecute las siguientes órdenes, cambiando RCDIR por la localización de los directorios rc de su sistema.

```
ln -s ../init.d/qmail RCDIR/rc0.d/K30qmail
ln -s ../init.d/qmail RCDIR/rc1.d/K30qmail
ln -s ../init.d/qmail RCDIR/rc2.d/S80qmail
ln -s ../init.d/qmail RCDIR/rc4.d/S80qmail
ln -s ../init.d/qmail RCDIR/rc5.d/S80qmail
ln -s ../init.d/qmail RCDIR/rc6.d/K80qmail
```

Dominios Virtuales

Los dominios virtuales son parecidos a los servidores múltiples tratados en la sección anterior, pero con algunas diferencias importantes. En primer lugar, si ejemplo.net aloja el dominio virtual virtual.proveedor.com, generalmente **not** se verifica que los mensajes dirigidos a pepe@ejemplo.net deban acabar en el mismo buzón de usuario que los dirigidos a pepe@virtual.proveedor.com. El espacio de nombres para cada dominio virtual es distinto.

Con *qmail*, los dominios virtuales se configuran en el fichero virtualdomains, que presenta una o más entradas del formato:

usuario@dominio:prefijo

qmail convierte usuario@dominio a prefijo-usuario@dominio y trata el resultado como si dominio fuera local. La parte usuario@ es opcional. Si se omite, la entrada comprenderá *todas* las direcciones de @dominio.

Volviendo al ejemplo del escenario anterior, si el administrador de correo de ejemplo.net quisiera crear un dominio virtual, proveedor.virtual.com, bajo el control administrativo del usuario juan, la siguiente entrada en virtualdomains conseguiría el efecto deseado:

proveedor.virtual.com:juan

Un mensaje de entrada para pepe@proveedor.virtual.com sería reescrito como juan-pepe@proveedor.virtual.com y entregado localmente. Véase la sección .qmail y la subsección Direcciones Ampliadas para más información sobre la forma en que juan puede gestionar su dominio virtual.

De la misma manera que sucedía con múltiples nombres de servidor, todos los dominios virtuales deben estar listados en rcpthosts para que qmail-smtpd sepa si aceptar mensajes dirigidos a ellos. Sin embargo, a diferencia de los nombres de servidor múltiples, los dominios virtuales **no se deben** añadir a locals.

Nota: los registros (MX) (*Mail eXchanger* = «intercambiador de correo») de su servidor de nombres de dominio (DNS) han de estar configurados para enviar al servidor de correo apropiado los mensajes del dominio virtual. Esta es tarea del administrador de nombres y de dominio y queda fuera del propósito de esta guía.

NOMBRES DE SERVIDOR MÚLTIPLES

Si su sistema tiene más de un nombre, es decir, direcciones del formato usuario@servidor1.proveedor.com se pueden escribir también como usuario@proveedor.com o bien usuario@correo.proveedor.com, entonces tiene que indicárselo a *qmail* para que sepa qué direcciones debe entregar localmente, y qué mensajes debe aceptar para sistemas remotos.

Para hacer esto, añada todos los nombres a dos ficheros de control:

- rcpthosts, que indica a qmail-smtpd que acepte correo dirigido a estos servidores, y también
- locals, que indica a qmail-send que las direcciones sobre estas máquinas han de entregarse localmente.

3 Referencias

IETF

http://www.ietf.org/rfc.html

- rfc 821 – SMTP.
- rfc 974 – Enrutado de MAIL.
- rfc 822 – Formato de los mensajes.
- rfc 2045,2046,2047,2048 – MIME 1.0
- rfc 1939 – POP3 Actual
- rfc1725 - POP3 - Post Office Protocol version 3
- rfc1082 - POP3 - Extended Service Offerings
- rfc1734 - POP3 - AUTHentication command

RELAYING

http://www.palomine.net/qmail/relaying.html

http://www.palomine.net/qmail/selectiverelay.html.

QMAIL

www.qmail.org

ANEXO A – SERVICIOS DE RED INETD/ XINETD / TCPSERVER

http://www.bandaancha.st/documentos.php?docid=51

INTRODUCCIÓN

Cuando un sistema oferta diferentes servicios de red, es habitual que cada uno de ellos esté escuchando en el puerto que le corresponde. Si un cliente quiere acceder a alguno de esos servicios, establece una conexión y posteriormente usa el protocolo asociado a dicho servicio.

De este modo, en el servidor, habrá un proceso que esté escuchando en cada uno de los puertos asociados a los servicios que ofrece. Si el sistema oferta N servicios, habrá N procesos escuchando en N puertos. Si un cliente establece una conexión a un puerto, el proceso a la escucha en dicho puerto lanzará un nuevo proceso que atienda al cliente.

Ya que todos los procesos que están a la escucha en los puertos harán esa pequeña tarea, podrían ahorrarse muchos recursos del sistema teniendo un único proceso a la escucha en todos los puertos que lance nuevos procesos que atiendan a los clientes dependiendo del puerto al que se conecten.

Por otro lado, si el servicio es fuertemente solicitado, puede crearse un cuello de botella, ya que el hecho de tener un proceso que dependiendo del puerto al que llegue la solicitud debe lanzar un programa puede ralentizar mucho la atención al servicio. Además, no todos los programas cumplen los requisitos necesarios para poder ser utilizados con un superserver, aunque la mayoría de ellos sí.

En resumen, esos programas que se dedican a escuchar puertos y lanzar un programa u otro dependiendo del puerto al que llegue la conexión se les suele llamar "super-servers". Los más famosos y extendidos son: inetd e xinetd -éste es una versión mejorada del primero-.

INETD

Es el "super-server" tradicional en los sistemas UNIX. Es lanzado al inicio del sistema y dependiendo de ciertos archivos de configuración, se pondrá a la escucha de los puertos especificados.

Tradicionalmente el archivo de configuración se encuentra bajo el directorio /etc y se llama inetd.conf. Su funcionamiento depende indirectamente de otros ficheros de configuración como services, protocols o rpc, situados todos ellos en /etc. El primero de ellos no es más que una asociación entre números de puerto y servicio. El segundo es una relación entre nombres de protocolo y números de protocolo. Y el tercero es otra asociación entre servicios rpc y números de programa rpc. Rara vez los archivos services, protocols o rpc son modificados, por lo que no se comentará más sobre ellos y pasaremos a la sintaxis utilizada en el archivo inetd.conf.

El archivo de configuración se trata de un fichero de texto que contiene líneas siguiendo el patrón:

servicio tipo_de_socket protocolo espera/no_espera usuario programa argumentos

Servicio: indica el nombre del servicio o programa RPC que atenderá. Buscará el/los ficheros /etc/services o /etc/rpc para averiguar el puerto o número de programa RPC asociado.

Tipo de socket: Se indicará el tipo de socket que se creará para escuchar ese servicios. Puede tener diversos valores, pero los más habituales son "dgram" para servicios que utilicen UDP y "stream" para servicios que se apoyen en TCP.

Protocolo: Indica el protocolo de este servicio. Dicho protocolo debe figurar en el archivo /etc/protocols. Lo más habituales son "tcp" o "udp".

Espera/No espera: Cuando inetd lance un proceso para atender a un cliente, tiene dos opciones: esperar a que se termine de atender al cliente antes de ponerse a escuchar de nuevo en el puerto, o no esperar y ponerse inmediatamente a la escucha. Si no queremos esperar, el valor a especificar es "nowait"; si queremos esperar, se especificará "wait".

Usuario: Aquí se especifica el usuario bajo cuya identidad se lanzarán los procesos servidores.

Programa: Path absoluto y nombre del programa que se lanzará con cada conexión cliente.

Argumentos: Argumentos del programa servidor, siendo el primero de ellos el nombre del programa, aunque esta vez no es necesario especificar el path absoluto.

Como ejemplo, vamos a suponer que queremos lanzar un servicio de FTP mediante inetd. Vamos a suponer que el ejecutable se encuentra en /usr/sbin y se llama myftpd y que tras consultar el manual hemos decidido lanzarlo con las opciones -s y -a. Además, no vamos a esperar a que se termine de atender a un cliente para escuchar de nuevo en el puerto. La línea que debería figurar en el archivo /etc/inetd.conf sería:

ftp stream tcp nowait root /usr/sbin/myftpd myftpd -s -a

XINETD

Inetd existe desde el comienzo de los siglos de Internet, unos tiempos en los que no había "chicos malos" que quisieran atacar máquinas, por lo que la seguridad de este programa no está muy desarrollada. El problema de seguridad de inetd es muy simple, no hemos visto ni el más mínimo parámetro que nos permita controlar el número máximo de clientes que queremos atender. Si algún o algunos chicos malos se pusieran a solicitar muchos servicios, inetd se limitaría a lanzar tantos procesos como fuera necesarios. Podría llegar a lanzar tantos que acabaría con la memoria del sistema. Esta es una rápida explicación de lo que se llama ataque por agotamiento de memoria.

Xinetd nace para mejorar este y muchos otros aspectos de inetd, pero la filosofía es siempre la misma: un proceso que escucha y que lanza otros procesos que atiendan a los clientes. Es mucho más completo y flexible que inetd, por lo que únicamente explicaremos aquellas cosas que considere más importante. Para un conocimiento más detallado, diríjanse a http://www.xinetd.org.

El fichero de configuración debe encontrarse en /usr/local/etc. La sintaxis del archivo es diferente que la utilizada por inetd, pero como he dicho antes, los conceptos son los mismo. El fichero se organiza en zonas, las cuales hacen referencia a cada servicio que se va a atender. Existe una zona especial donde se sitúan los valores por defecto de las demás zonas. Un ejemplo podría ser el siguiente:

```
defaults

{

instances = 25

log_type = FILE /var/log/xinetd

log_on_success = HOST PID

log_on_failure = HOST RECORD

}
```

Como ejemplo, se impondrá un número máximo de 25 instancias para cada servicio. Además, se registrará el HOST y el PID cuando se lance el servicio correctamente; y el HOST y toda la información existe sobre el cliente en caso contrario. El logging se hará en el fichero /var/log/xinetd.

Una vez que hemos puesto los parámetros por defecto, debemos empezar a crear las zonas que creamos convenientes para los servicios que vamos a ofrecer. Para definir una zona de un servicio, tomaremos el mismo ejemplo utilizado para inetd. La definición sería la siguiente:

```
service ftp

{

socket_type = stream

protocol = tcp

wait = no

user = root

server = /usr/sbin/myftpd

server_args = -s -a

}
```

Es fácil adivinar la funcionalidad de las opciones que aquí se muestran, los valores son los mismos que acepta inetd. La única salvedad es que a la hora de introducir los argumentos del programa, no es necesario incluir de nuevo el nombre del programa.

Únicamente he comentado las opciones más típicas, sin embargo hay muchas más que pueden ser de gran utilidad. Como se ha visto, la configuración es muy sencilla. Si necesitas más información, únicamente debes teclear man xinetd o man xinetd.conf y allí encontrarás todo lo que necesites.

TCPSERVER

Tcpserver se incluye en el paquete ucspi-tcp del profesor D. J. Bernstein junto con otras utilidades. Como su nombre indica, únicamente es capaz de esperar conexiones TCP.

Este programa no se utiliza de misma forma que inetd o xinetd, no es un proceso que escucha diferentes puertos. Tcpserver escuchará un único puerto y lanzará un programa. ¿Y entonces por qué utilizar tcpserver? Porque muchos programas servidores necesitan de otro programa que escuche por ellos el puerto como hace inetd o xinetd. Sin embargo tcpserver es más rápido, fiable y seguro que cualquiera de ellos dos. Tcpserver es muy recomendable utilizarlo con servicios que vayan a tener una gran demanda en nuestro sistema. La información sobre el paquete ucspi-tcp puede encontrarse en http://cr.yp.to/ucspi-tcp.html.

La sintaxis de tcpserver es la siguiente:

tcpserver opciones host puerto programa

Dejando de momento a un lado las opciones que acepta tcpserver, paso a explicar los demás parámetros:

Host: Indica en qué IP esperará las conexiones. Con un 0 (cero) se indica que se escuche en todas las direcciones IP asignadas a nuestro sistema.

Puerto: Indica el puerto TCP en el que se esperarán las conexiones. Puede ser un número o el nombre asociado según el archivo /etc/services. Por ejemplo, da lo mismo escribir 25 que smtp.

Programa: Aquí se indica el programa con los argumentos que consideremos necesarios.

Opciones: Algunas de las opciones más usadas son las siguientes:

-u: Se indica el nombre de usuario bajo el cual se ejecutará el programa servidor.

-g: Se indica el grupo bajo el cual se ejecutará el programa servidor.

-c: Establece un número máximo de programas servidores que se lanzarán simultáneamente.

-H: No buscará en el DNS el nombre de máquina de la IP del cliente.

-R: No intentará obtener información sobre el cliente vía algún protocolo de identificación.

Existe otra opción que es ampliamente utilizada que nos servirá para especificar desde qué IPs podrán establecer conexiones al servicio que estamos ofertando. Esto nos lleva a hablar de un nuevo programa que incluye el paquete ucspi-tcp: tcprules.

Con **tcprules** podemos crear bases de datos tipo cdb en las que indicaremos desde donde se podrá acceder al servicio. Para ello se deberá crear un archivo con una sintaxis específica y con tcprules creamos la base de datos asociada.

Por ejemplo, queremos de nuevo ofertar un servicio ftp, pero ahora sólo será accesible dentro de nuestra red. Para ello crearemos un fichero en /usr/local/etc/rules.ftp que contendrá lo siguiente:

172.26.0.:allow

:deny

Con esto indicamos que sólo la red 172.26.0.* podrá acceder al servicio y que todos los demás serán ignorados. Ahora debemos convertir ese archivo de texto en uno binario, mucho más rápido de consultar. Esto lo realizaremos de la siguiente manera:

```
tcprules /usr/local/etc/rules.ftp.cdb /usr/local/etc/rules.ftp.tmp < 
/usr/local/etc/rules.ftp
```

De esta manera ya hemos creado un archivo de reglas que puede ser utilizado por tcpserver añadiendo la opción: -x path_del_archivo_cdb

Para ofertar el servicio ftp con las condiciones expuestas, debemos lanzar el siguiente mandato:

```
tcpserver -R -H -c25 -u "UID-root" -g "GID-wheel" -x
/usr/local/etc/rules.ftp.cdb 0 ftp /usr/sbin/myftp -s -a
```

Como he dicho antes, tcpserver admite muchos parámetros y es muy flexible, por lo que conviene investigar más a fondo en su funcionamiento. Aunque eso es algo que os dejo para vosotros.

NOTA: Las páginas del man no se incluyen con la distribución del programa, pero pueden encontrarse en http://smarden/pape/djb .

ANEXO B: LOGS DEL SISTEMA

FICHEROS DE LOGS

Presentaremos aqui los fichero de logs mas habituales en un sistema Linux, y la configuracion de alguno de ellos como syslog.

Los ficheros de log en un sistema linux, residen en el directorio

/var/log
y aquí es donde deberían logear todos los programas.

syslog

syslog es un log del sistema y del kernel que nos puede dar importante información de eventos que suceden en el sistema y en sus programas. Syslog provee incluso alguna llamada para que los programas que corren en el sistema logeen en el propio syslog. Todas las entradas que presenta syslog tienen como mínimo una fecha y una hora, el nombre de la maquina y del programa que generó el evento. El fichero de configuración de syslog es

```
/etc/syslog.conf
y podemos ver un ejemplo de fichero de configuración.
#  /etc/syslog.conf     Configuration file for syslogd.
#
#                       For more information see syslog.conf(5)
#                       manpage.

#
# First some standard logfiles.  Log by facility.
#

auth,authpriv.*                 /var/log/auth.log
*.*;auth,authpriv.none          -/var/log/syslog
#cron.*                         /var/log/cron.log
daemon.*                        -/var/log/daemon.log
kern.*                          -/var/log/kern.log
lpr.*                           -/var/log/lpr.log
mail.*                          /var/log/mail.log
user.*                          -/var/log/user.log
uucp.*                          -/var/log/uucp.log

#
# Logging for the mail system. Split it up so that
# it is easy to write scripts to parse these files.
#
mail.info                       -/var/log/mail.info
mail.warn                       -/var/log/mail.warn
mail.err                        /var/log/mail.err

# Logging for INN news system
#
news.crit                       /var/log/news/news.crit
news.err                        /var/log/news/news.err
news.notice                     -/var/log/news/news.notice

#
```

```
# Some `catch-all' logfiles.
#
*.=debug;\
        auth,authpriv.none;\
        news.none;mail.none     -/var/log/debug
*.=info;*.=notice;*.=warn;\
        auth,authpriv.none;\
        cron,daemon.none;\
        mail,news.none          -/var/log/messages

#
# Emergencies are sent to everybody logged in.
#
*.emerg                         *

#
# I like to have messages displayed on the console, but only on a virtual
# console I usually leave idle.
#
#daemon,mail.*;\
#       news.=crit;news.=err;news.=notice;\
#       *.=debug;*.=info;\
#       *.=notice;*.=warn       /dev/tty8

# The named pipe /dev/xconsole is for the `xconsole' utility.  To use it,
# you must invoke `xconsole' with the `-file' option:
#
#    $ xconsole -file /dev/xconsole [...]
#
# NOTE: adjust the list below, or you'll go crazy if you have a
reasonably
#      busy site..
#
daemon.*;mail.*;\
        news.crit;news.err;news.notice;\
        *.=debug;*.=info;\
        *.=notice;*.=warn       |/dev/xconsole

local2.*                -/var/log/ppp.log
```

Aunque no nos pararemos a ver este archivo de configuración con demasiado detalle, si es interesante ver en base a que criterios trabaja syslog. Podemos ver en el fichero de configuración, lineas que nombran una "facility" (subsistema de aplicación) y separados por un punto las palabras crit,debug,info,notice o warn. Por ejemplo:

```
 mail.warn
-/var/log/mail.warn
```

Lo que realiza syslog es logear los eventos dependiendo de la aplicación y de la prioridad del evento (crit = critical, warn=warning, info=information, err=errors ....) y reenviandolos a syslog o a otros logs separados para "categorizar" mejor cada uno de los eventos, puede usar también comodines de modo que sea posible incluir una linea de este tipo

kern.* -/var/log/kern.log

de modo que todos los mensajes del kernel vayan a fichero kern.log, sean del tipo que sean.

Como curiosidad hacer notar que las entradas de ficheros que van precedidos por un guión (-) indican que no se hace un "sync" cada vez que existe una entrada en ese log, de modo que si hay una caida del sistema pueden perderse datos en este fichero.

OTROS LOGS EN LINUX

Además de syslog y de los logs generados por el mismo, hay otros logs que hay que tener en cuenta para saber en cada momento que ocurre o a ocurrido en nuestro sistema. Enumeramos aquí algunos y su cometido.

- */var/log/xferlog* Este fichero es creado por los servidores de ftp e indica la fecha de las transferencias de ficheros, los ficheros, cantidad de bytes, etc...
- */var/log/apache/access.log* Fichero creado por el servidor web apache e indica las conexiones al servidor, con que version http, si ha sido un GET o un put, etc.
- */var/log/apache/error.log* Da los errores (categorizados por warn, notice, etc...) que surgen en el servidor web.
- */var/log/setuid.changes* Log generado por el programa checksecurity incluido en la distribución Debian y que da un listado de los setuids en el sistema. Se activa en el cron.
- */var/log/wtmp* Es un log binario que guarda el los usuarios del sistema que han hecho logins. No se usa directamente pero si podemos usarlo con la instrucción last por ejemplo.

"ROTADO" Y REPLICADO DE LOGS.

En tanto en cuanto los logs representan a nuestros sistema, es necesario tener un cuidado con los mismos y "sanearlos" de alguna manera, veremos en primer lugar el rotado de logs y posteriormente el replicado de los mismos para asegurar una consistencia y redundancia que evite problemas en un posible compromiso de la maquina donde residen estos logs.

Rotado

Los logs, especialmente algunos de ellos, tienden a crecer de manera exagerada, de modo que es un uso habitual guardar (en ocasiones compromidos) los logs de vez en cuando e iniciar un nuevo log. Como ejemplo, tomemos este 'ls' de /var/log:

```
emain:/var/log $ ls -1 syslog*
syslog
syslog.0
syslog.1.gz
syslog.2.gz
syslog.3.gz
syslog.4.gz
syslog.5.gz
syslog.6.gz
```

Vemos que hay un syslog, otro numerado con 0 y los demas numerados y comprimidos. Esto lo podemos hacer con la utilidad logrotate incluyendola en el cron.

La utilidad logrotate tiene un fichero de configuración que podemos comentar:

```
# see "man logrotate" for details
# rotate log files weekly
weekly

# keep 4 weeks worth of backlogs
rotate 4

# send errors to root
errors root
```

```
# create new (empty) log files after rotating old ones
create

# uncomment this if you want your log files compressed
#compress

# RPM packages drop log rotation information into this directory
include /etc/logrotate.d

# no packages own wtmp or btmp -- we'll rotate them here
/var/log/wtmp {
    monthly
    create 0664 root utmp
    rotate 1
}

/var/log/btmp {
    missingok
    monthly
    create 0664 root utmp
    rotate 1
}

# system-specific logs may be configured here
```

Este fichero rige los rotados de logs, en este caso se haran cada semana (weekly), guardaran los logs antiguos 4 semanas y creará unos nuevos cada vez que lo rote. Además existe la posibilidad de especificar el comportamiento especifico de algunos ficheros determinados en este caso btmp y wtmp.

Replicado

El replicado de logs está más encaminado a la redundancia y seguridad de los logs que al propio saneamiento. Syslog da la posibilidad de logear de manera remota además de en la propia máquina en otras máquinas de una red, de modo que la misma información este replicada y distribuida por más nodos de la red en caso de que una determinada maquina haya sufrido una intrusión y se haya intentado borrar información.

Describiremos los pasos para hacerlo, enumerandolos:

- *Arranque de syslog:* Activaremos syslog con el parámetro -r en la máquina que va a recibir los mensajes de syslog remotos. Por defecto syslog no permite mensajes remotos.
- */etc/services:* Es importante incluir una entrada en /etc/services que indique el puerto y protocolo de syslog de este modo:
- syslog 514/udp
- */etc/syslog.conf:* Indicaremos ahora en el fichero de configuración de syslog que mensajes queremos replicar y a que nodos, por ejemplo, podríamos replicarlos todos del siguiente modo:
- # Sample syslogd configuration file to
- # messages to a remote host forward all.
- *.* @hostname

o un subsistema (facility) concreta con sus determinadas prioridades, por ejemplo:

kern.warm @hostname

Es importante hacer algunas aclaraciones en cuanto al replicado, en primer lugar que el transporte se realiza con UDP, es decir nada nos asegura que esos paquetes lleguen a su destino. Existen programas que ya realizan el transporte con TCP. También es necesario hacer notar que corremos el peligro de que exista "sniffing" en la red o que alguien intente enviar logs falsos algunos de los nodos existentes. Algunas de las soluciones pasan por encriptación o por tuneles "seguros"

Logcheck y tripwire

En esta sección veremos dos utilidades una para realizar chequeos automáticos de logs y hacer saltar alarmas por mail de modo automático y la otra para realizar chequeos de integridad de los ficheros de un sistema.

Logcheck

Logcheck permite generar alarmas por mail cuando se encuentran determinados patrones en los logs. El fichero de configuració esta en /etc/logcheck/logcheck.conf y posteriormente tenemos unos ficheros donde existen una serie de patrones que aparecen en casos de intentos de "hacking" etc..., vemos un listado de este directorio:

```
emain:/etc/logcheck# ls
logcheck.conf     logcheck.ignore      logcheck.violations.ignore
logcheck.hacking  logcheck.violations
y un mail generado por un error en una autentificación:
Date: Wed, 17 May 2000 23:02:02 +0200
From: root <root@emain.celtic.org>
To: root@emain
Subject: emain 05/17/00:23.02 system check


Security Violations
=-=-=-=-=-=-=-=-=-=-=
May 17 22:09:38 emain PAM_unix[579]: authentication failure;
david(uid=1000)
->
root for su service
May 17 22:09:40 emain su[579]: pam_authenticate: Authentication failure
```

Tripwire

Tripwire es una utilidad aparentemente simple que recorre nuestros sistemas de ficheros y extracta de ellos una suma MD5 que deberemos de guardar en un soporte completamente seguro fisicamente (por ejemplo un diskette) y la cual compararemos con la siguiente suma MD5 de los ficheros que hagamos para asegurar que siguen igual y que sus tamaños no han cambiado, de haberlo hecho es casi seguro que estamos ante una intrusión.

```
Tripwire se basa en un fichero de configuración que reside en

/etc/tripwire/
un ejemplo del mismo:
#
# tripwire.config for Linux/Debian machines
#
# I have tried to provide for a reasonable, minimal configuration file.
```

```
# You will have to tune this to your own taste and needs. -- pw.
#
# I even removed some more stuff to make it fit on a floppy. -- MM

# Define variables for searching devices, tmp directories, and logfiles
@@define DEVSEARCH E+ins
@@define TMPSEARCH E+ugp
@@define LOGSEARCH L-i

# Check all files:
# (We also mention some directories explicitly, as
#  these are often put on a separate filesystem)
/               R
/usr            R
/usr/local      R

# Don't do these
# (/mnt is for temporarily mounted filesystems;
#  do a minimal check on /home anyway;
#  no spool files except the crontab for root):
!/mnt
=/home
!/root

#
# I don't like /var since too many files change automatically. -- MM
#
#/var           R
!/var

# Log files:
#/var/log       @@LOGSEARCH
#/var/account   @@LOGSEARCH

#  /dev, /tmp and /var/tmp
#
/dev            @@DEVSEARCH
=/tmp           @@TMPSEARCH
=/usr/tmp       @@TMPSEARCH

# no checksums for less important files (documentation, word lists):
# you might want to add /usr/X11R6/man if you have X installed
!/usr/doc
!/usr/dict
!/usr/info
!/usr/man
!/usr/src

# but do check the kernel sources
/usr/src/linux
```

Vemos que en este fichero indicaremos de que directorios no queremos que se hagan copias bien por ser ficheros muy cambiantes o bien por que no tienen información sensible.

El primer paso para analizar con tripwire es la creación de una base con las sumas de los ficheros de los sistmas:

tripwire -initialize

con esto el programa creará una base de datos que guardará en /usr/lib/tripwire/databases

o en un lugar indicado previamente (homde de root) y que deberemos guardar en un medio fisicamente seguro y nunca accesible por red, para que no se puedan operar cambios en las sumas.

Posteriormente con la orden

tripwire -initialize -d basededatos

el programa irá a buscar la base de datos anterior que comparará con el estado actual del sistema para comprobar que no ha habido cambios en los binarios.

ANEXO C: COMANDOS DE QMAIL

queue management

qmail-qstat

Description: summarize status of mail queue

Command line parameters: none

Environment Variables: none

Configuration files: none

Exit codes: complains if there is a problem

Remarks: must be run either as root or with gid qmail

qmail-qread

Description: list outgoing messages and recipients

Command line parameters: none

Environment Variables: none

Configuration files: none

Exit codes: complains if there is a problem

Remarks: must be run either as root or with uid qmails and gid qmail

qmail-tcpto

Description: prints qmail-remote's current list of timeouts

Command line parameters: none

Environment Variables: none

Configuration files: none

Exit codes: complains if there is a problem

Remarks: must be run either as root or with uid qmailr and gid qmail

qmail-tcpok

Description: erases qmail-remote's current list of timeouts

Command line parameters: none

Environment Variables: none

Configuration files: none

Exit codes: complains if there is a problem

Remarks: must be run either as root or with uid qmailr and gid qmail

Hint: to reschedule every message in the queue for immediate delivery, do a "kill -ALRM *pid*" on the qmail-send pid

user and system management

qmail-pw2u

Description: build address assignments from a passwd file

Command line parameters: [-/bHuUC] [-cchar]

Environment Variables: none

Configuration files: ~users/include ~users/exclude ~users/mailnames ~users/subusers ~users/append

Exit codes: complains if there is a problem

Remarks: generates ~users/assign

qmail-newu

Description: prepare address assignments for qmail-lspawn

Command line parameters: none

Environment Variables: none

Configuration files: ~users/assign

Exit codes: complains if there is a problem with ~users/assign qmail-newu

Remarks: generates ~users/cdb

qmail-newmrh

Description: prepare morercpthosts for qmail-smtpd

Command line parameters: none

Environment Variables: none

Configuration files: ~control/morercpthosts

Exit codes: complains if there is a problem with control/morercpthosts qmail-newmrh complains

Remarks: generates ~control/morercpthosts.cdb

qmail-showctl

Description: analyze the qmail config files

Command line parameters: none

Environment Variables: none

Configuration files: all

Exit codes: complains if there is a problem

Remarks: explains the current qmail configuration

other qmail daemons

qmail-start

Description: turn on mail delivery

Command line parameters: defaultdelivery logger [args ...]

Environment Variables: none

Configuration files: none

Exit codes: does not print anything, even on failure

Remarks: make sure to clean up the environment before starting qmail

qmail-clean

Description: clean up the queue directory

Command line parameters: none

Environment Variables: none

Configuration files: none

Exit codes: none

Remarks: can only be started by qmail-start

splogger

Description: reads a series of messages and feeds them to syslog

Command line parameters: [tag [facility]]

Environment Variables: none

Configuration files: none

Exit codes: complains if there is a problem

Remarks: converts unprintable characters to question marks

R19980814 © 1998 by Andre Oppermann, opi@nrg4u.com, http://www.nrg4u.com. Free for personal use. Not free for commercial redistribution (eg. Magazines). Contact author for permission.