Administración de correo electrónico en Linux

Horst von Brand
vonbrand@inf.utfsm.cl

30 de octubre de 2001

Resumen

El servicio de correo electrónico es extremadamente importante, muchas personas dependen de él en múltiples formas para sus actividades diarias. Los usos actuales del correo son muy variados, mucho más de lo que pudiesen haber sospechado sus creadores.

Por otro lado, al ser un servicio que tiene estrechas interacciones con el medio (la red), es un blanco apetitoso para posibles atacantes (los problemas de seguridad en el correo electrónico han sido endémicos). Además, el correo eletrónico es un punto de contacto directo con los usuarios del sistema, por lo que el servicio tiene también impacto sobre la configuración más interna del sistema.

Por ser un tema muy amplio, sólo trataremos una alternativa de uso común en configuración relativamente sencilla.

1 Introducción

El correo electrónico es un caso curioso: En los proyectos iniciales que dieron lugar a Internet nunca se consideró esta clase de servicio. Sin embargo, al poco andar el correo electrónico era con mucho el servicio de red que más tráfico generaba. Una encuesta reciente en *El Mercurio* muestra que en promedio los usuarios de Internet en Chile pasan un 46% del tiempo leyendo su correo. Si el correo electrónico era importante hace unos años, hoy en día es imprescindible. No sólo los asiduos a la computación y los usuarios frecuentes dependen hoy del correo electrónico para sus tareas diarias. Ha reemplazado en gran medida los memorandos internos en las organizaciones, ha desplazado a las llamadas telefónicas a un segundo plano, y es una fuerte competencia al correo tradicional.

Se tratan los aspectos más importantes de la administración del correo electrónico en Linux. Este documento se centra en los paquetes más comunes en uso, no tenemos el espacio para cubrir todas las alternativas disponibles. Asimismo, se cubre únicamente transporte de correo a través de TCP/IP, dado que los demás medios de transporte son de uso muy restringido hoy en día. En este sentido, hoy vivimos un mundo más simple que el que vió el nacimiento del correo electrónico: Las direcciones de correo hoy son uniformemente Internet (el familiar formato de `usuario@dominio`). Con la creciente importancia del correo electrónico y el crecimiento explosivo de todo Internet los volúmenes de correo que deben manejarse hoy día (particularmente en los ISPs importantes) van mucho más allá de lo que las configuraciones originales del sistema son capaces de manejar.

En el sistema de transporte de correo, las mayores distribuciones Linux son estándar. Por esto, lo que se discute acá debiera aplicarse sin grandes cambios a cualquiera de ellas, y también a Unix en general. Eso sí que muchas versiones propietarias de Unix traen sus propias versiones "mejoradas" de los sistemas discutidos acá. Mi experiencia es que lo más sano en tal caso instalar el sistema de correo desde fuentes y olvidarse de las "mejoras" (que pueden ser importantes, pero generalmente sólo en ambientes homogéneos; en ambientes heterogéneos donde tienen que convivir con las "mejoras" incompatibles de los demás provocan más dolores de cabeza que lo que valen). Esta operación es una que debe efectuarse junto con el obligatorio `GNU > /usr/local` al arribar una nueva máquina UNIX.

Un problema particularmente importante en la actualidad es lo que se conoce coloquialmente como *spam*, o más académicamente como UCE, por *Unsolicited Commercial Email* (correo electrónico comercial no solicitado). Distribuir correo electrónico a miles, o incluso millones, de potenciales clientes es muy barato. En especial si el costo de hacerlo llegar no lo paga uno, sino terceros incautos.

1.1 Otros sistemas de los cuales depende el correo

Por su relación con las demás máquinas en Internet obviamente necesita del correcto funcionamiento de la red. Depende íntimamente del servicio de nombres (DNS), incluso hay registros especiales en DNS para correo electrónico, como el RR (Resource Record) MX (Mail eXchanger). Estos registros controlan el flujo de correo en Internet, y en particular es importante manejar correctamente la interrelación entre registros MX, servidores de correo, y configuraciones de cada uno.

1.2 Una confusa colección de abreviaturas

Toda área técnica adquiere una colección de abreviaturas impresionantes para darse importancia frente a los incautos. El correo electrónico no es excepción. Además, algunas de las abreviaturas no son de uso universal, hay ligeras variantes en uso también. Un esquema general aparece en la figura 1, adaptado de [4].

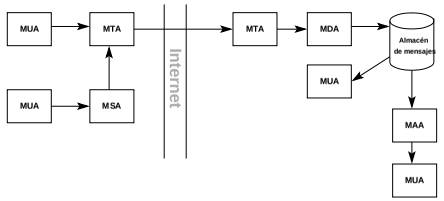

Figura 1: Esquema de las piezas

MTA: Por *Mail Transfer Agent*, agente de transferencia de correo. Es el programa que transfiere correo de un sitio a otro. Se encuentra en el port TCP 25.

MUA: Por *Mail User Agent*, o agente de correo del usuario. Es el programa que el usuario emplea para componer o leer su correo.

MSA: Por *Mail Submission Agent*, o agente de recepción de correo. En el caso de `sendmail` le llaman MSP (por *Mail Submission Program*). La idea de interponer un MSA es que así se descarga del MTA la tarea de verificar los mensajes entrantes, lo que es importante para grandes flujos de correo. Además, esta configuración permite controlar mejor los privilegios con los cuales se efectúa cada tarea, lo que redunda en mayor seguridad. En tal caso, el MSA puede correr en una máquina diferente de la que alberga al MTA. MSA usa el port TCP 587.

MDA: Por *Mail Delivery Agent*, agente de entrega de correo. Es el encargado de almacenar el correo recibido en algún repositorio. En la documentación de `sendmail` le llaman LDA, por *Local Delivery Agent*, agente de entrega local.

MAA: Por *Mail Access Agent*, o agente de acceso al correo. Permite al usuario acceder a su correo en el repositorio, típicamente en forma remota. Muchos MUAs populares incluyen esta funcionalidad.

Además de estas piezas que son los programas que intervienen en el manejo del correo, hay otros términos importantes:

MIME: Por *Multipurpose Internet Mail Extensions*, extensiones multipropósito para correo electrónico. Básicamente es un esquema para estructurar el cuerpo de mensajes, ofreciendo la posibilidad de incluir datos de diversos tipos.

1.3 Sistema a considerar

Como ya se indicó, es imposible tratar todos (o siquiera una selección grande) de los paquetes y protocolos involucrados en el correo electrónico. Concentraremos la discusión en un MTA, `sendmail` [6, 2], en particular la discisión se centra en las últimas versiones de este sistema (actualmente es 8.12.1) y el protocolo ESMTP standard de los MTA. Algunas de las características de `sendmail-8.12.1` que consideraremos son nuevas con esta versión, por lo que no necesariamente todo lo dicho es aplicable sin cambios a versiones anteriores. Sin embargo, esta versión (o alguna muy similar) será la de uso corriente en un futuro no lejano.

Hay una variedad de otros MTAs en uso, entre los que se cuentan `qmail` [1], `exim` [3], y `postfix` [7]. Apuntan a ser MTAs más simples de manejar que `sendmail` (apovechando que el mundo del correo electrónico hoy es mucho más simple que en sus inicios) y más seguros por diseño que el venerable `sendmail`, pero no han logrado hacerle real mella a la popularidad de éste…

En todo caso, la mayor parte del presente documento es aplicable a cualquier sistema de correo razonablemente estándar.

El repositorio de mensajes puede estar implementado de diversas formas, en la configuración tradicional es simplemente un archivo (generalmente llamado *mbox*, por *mail box* o casilla de correo). Es tarea del MDA y del MAA administrarlo, y éstos suelen ser programas independientes del MTA mismo. Más adelante discutiremos el formato de un mensaje de correo, y cómo aparece en el *mbox*. En el área de MAA los protocolos más comunes son POP3 (port TCP 110) e IMAP (port TCP 142).

Un lugar curioso lo ocupa `fetchmail` [5], usado con conexiones intermitentes. Usa alguna funcionalidad de MAA (generalmente POP3 o IMAP) para recoger correo remotamente, y luego de “corregir” los encabezados que indican el destino reinyecta el mensaje en el sistema para entrega local. También puede hacer uso del comando ESMTP ETRN para solicitar que se procesen los mensajes en la cola remota destinados a un sitio, logrando efecto similar.

2 El formato de un mensaje

El formato básico de un mensaje de correo en Internet está descrito por el RFC822. El formato en sí es sencillo, pero muy flexible.

2.1 Direcciones de correo

Internet, como una colección de redes conectadas permamentemente, es un fenómeno relativamente nuevo. Antes de la posibilidad de acceder directamente a casi cualquier lugar de la red el sistema de correo funcionaba mediante *store-and-forward*, vale decir, cada MTA tenía como misión recibir correo, almacenarlo, y eventualmente enviarlo un paso más cerca del destino. No siempre los medios de transporte usados para recibir y reexpedir el correo eran los mismos, y cada medio de transporte usaba su propio formato de direcciones. En este esquema las direcciones no eran absolutas, la dirección de correo indicaba explícitamente el camino a seguir por el mensaje. Como una simplificación para el usuario en UUCP se podían dar las rutas desde ciertos nodos *bien conocidos*, con lo que el sistema tenía que

saber cómo llegar a éstos. Esto, junto con la necesidad de manejar simultáneamente diversos formatos de dirección en conjunto, explican en gran medida la complejidad de la configuración de `sendmail`. Si se ve en la necesidad de manejar esto, refiérase a [2] y a la documentación que viene con el programa.

Como el correo originalmente funcionaba por store-and-forward, en un mundo más amable los MTAs aceptaban correo de cualquier origen haciendo un esfuerzo para acercar el mensaje a su destino. Con la epidemia de spam en la red lentamente los MTAs se están configurando para sólo aceptar correo a entregar a terceros (hacer de *relay*) desde máquinas específicamente autorizadas. En particular, nuestro MTA ejemplo (`sendmail`) tiene provisiones al efecto desde la versión 8.8.0 (1996), y desde la versión 8.9.0 (1998) aparecen en la configuración recomendada. Las configuraciones por omisión de las distribuciones mayores hoy en día son tremendamente restrictivas para evitar que una estación de trabajo se transforme sin querer en “distribuidor de avisos económicos,” con el consiguiente consumo de recursos, además del riesgo de caer en una lista negra de máquinas que originan spam y quedar fuera del sistema de correo mientras no se corrija la configuración.

2.2 Encabezados de correo

Un mensaje puede dividirse en tres partes:

- El sobre (*envelope*)
- Los encabezados (*headers*)
- El cuerpo del mensaje (*body*)

El sobre determina a quién entregar el mensaje, o en caso que la entrega resulte infructuosa, a quién retornarlo.

El sobre es invisible para los usuarios, `sendmail` lo usa internamente para direccionar el mensaje.

Los encabezados son una colección de pares propiedad/valor según el formato de RFC822. Registran una variedad de información sobre el mensaje, tales como la fecha y hora a la que fue enviado y por cuáles MTAs pasó en camino. Son parte integrante del mensaje, aunque el MUA suele ocultar los menos interesantes al desplegar el mensaje para el usuario.

El cuerpo del mensaje es el contenido a ser enviado. Debe consistir únicamente de texto ASCII, aunque puede contener codificaciones a prueba de correo electrónico de una variedad de formatos binarios. Esta es la función de MIME.

Cada línea de encabezado comienza con una palabra clave como To, From, o Subject, seguido por dos puntos (‘:’) y el contenido de ese encabezado. El formato de los encabezados estándar está definido por el RFC822; sin embargo, se permiten encabezados adicionales. Cualquier encabezado que comienze con “X-” es propagado por el sistema de correo, quien simplemente los ignora. Así, puede agregarse un encabezado como “X-Chiste-del-Dia” sin interferir con el funcionamiento del correo.

Algunos de los encabezados son provistos por el MUA, otros son agregados por el MTA. Los encabezados que trazan el camino seguido por el mensaje generalmente son considerados “poco interesantes” por los MUA, que no los muestran, pero generalmente hay una opción para verlos todos. La habilidad de leer encabezados es importante a la hora de trazar a su fuente una pieza de spam, de forma de tomar medidas para erradicarlo.

Un ejemplo de spam a analizar aparece en la figura 2. La numeración de las líneas en la figura es sólo para referencia. Como puede apreciarse, cuentan como continuación líneas indentadas. Me tomé algunas libertades para hacer caber las líneas en la página. Las líneas 1 a 31 son los encabezados, el cuerpo (líneas 33 a 38) está separado de éstas mediante una línea en blanco. Se resumió el cuerpo para ahorrar espacio.

```
 1  Return-Path: jthzb8nsskqsb@hotmail.com
 2  Delivery-Date: Sun Oct 21 19:57:12 2001
 3  Return-Path: <jthzb8nsskqsb@hotmail.com>
 4  Received: from localhost (IDENT:vonbrand@localhost [127.0.0.1])
 5          by sleipnir.valparaiso.cl (8.12.1/8.12.1)
 6          with ESMTP id f9LMudXV031822
 7          for <vonbrand@localhost>; Sun, 21 Oct 2001 19:57:07 -0300
 8  Received: from inti.inf.utfsm.cl [200.1.19.1]
 9          by localhost with IMAP (fetchmail-5.9.4)
10          for vonbrand@localhost (single-drop);
11          Sun, 21 Oct 2001 19:57:07 -0300 (CLST)
12  Received: from server.transtek.com.tw ([211.21.101.250])
13          by inti.inf.utfsm.cl (8.11.6/8.11.6) with ESMTP id f9LJq3W07405;
14          Sun, 21 Oct 2001 16:52:06 -0300
15  Received: from 216.143.146.191 (ppp-216-143-146-191.mclass.broadwing.net
16          [216.143.146.191])
17          by server.transtek.com.tw (8.9.3/8.8.7) with SMTP id JAA06275;
18          Fri, 19 Oct 2001 09:08:33 +0800
19  From: jthzb8nsskqsb@hotmail.com
20  Message-Id: <200110190108.JAA06275@server.transtek.com.tw>
21  To: <c2ct3qewu@hotmail.com>
22  Subject: Get REAL Viagra Immediatly
23  Date: Thu, 18 Oct 2001 20:16:48 -0400
24  MIME-Version: 1.0
25  Content-Type: text/html;
26          charset="iso-8859-1"
27  Content-Transfer-Encoding: quoted-printable
28  X-Priority: 3
29  X-MSMail-Priority: Normal
30  Errors-To: fucrin1an9u@kki.com.pl
31  X-Mailer: Microsoft Outlook Express 5.50.4133.2400
32
33
34  <html>
35  <body>
36  [...]
37  </body>
38  </html>
```

Figura 2: ¿Alguien interesado en Viagra?

Las líneas 24 a 27 identifican la versión de MIME en que viene el cuerpo (incidentalmente mostrando que el cuerpo *es* MIME, línea 24) y el tipo de contenido (HTML escrito en Latin-1, líneas 25–26), que a su vez fue codificado como quoted-printable (línea 27), ya que es texto ASCII con ocasionales caracteres adicionales, que se codifican en hexadecimal, para evitar que el correo que según estándar es ASCII los maltrate.

Las líneas 21, 22, y 24 a 31 las pone el MUA de origen (To, Subject, MIME-Version, Content-Type, Errors-To son encabezados estándar, los demás son propios). Nótese que el encabezado To nada tiene que ver con mi dirección de correo, es el destinatario de encabezado. En el sobre (que acá no se ve, ya indicamos que es una estructura temporal que usa el MTA para direccionar el mensaje) es donde aparece mi dirección, junto con quién sabe cuántas otras víctimas. Como el From lo pone el MTA de origen según lo que le dice el MUA, obviamente no es confiable (en este caso, el mensaje proviene de Taiwán como veremos, la dirección de origen en Hotmail que aparece en From seguramente sea falsificada).

Los demás encabezados los ponen los MTA en tránsito. El encabezado Date da el instante en que se envió el mensaje (según el reloj de la máquina en que corre el primer MTA en camino al destino, que no necesariamente tenga relación con la realidad). El encabezado Message-Id es un identificador único de este mensaje en particular, puede usarse para trazar el mensaje en los registros de los diversos MTA que lo manipularon.

Los encabezados más interesantes son los Received. Cada vez que un MTA recibe el mensaje, agrega un nuevo encabezado de éstos antes de los ya existentes. Por tanto, de abajo hacia arriba trazan el camino del mensaje. El primero (línea 15) indica que la máquina `server.transtek.com.tw` (en Taiwán) recibió el mensaje de una máquina que se identificó como `216.143.146.191`. Una consulta a DNS por esta dirección dió el nombre `ppp-216-143-146-191.mclass.broadwing.net`, correspondiente a la dirección IP `216.143.146.191` desde la cual se inició la conexión. A su vez (línea 12), `inti.inf.utfsm.cl` recibió el mensaje desde una máquina que se identificó como `server.transtek.com.tw` desde la dirección IP `211.21.101.250` (indicado por los corchetes). Luego el mensaje fue recogido por `fetchmail` desde `inti.inf.utfsm.cl` (línea 8) usando IMAP. Finalmente (línea 4), el MTA local recibió el mensaje desde un proceso iniciado por la cuenta `vonbrand` en `localhost`, vale decir, `sleipnir.valparaiso.cl`.

De estos encabezados se pueden creer únicamente los que provee la máquina local, todo el resto viene a través de la red y puede falsificarse. En todo caso, acá los indicios son coherentes. Desde una máquina ostensiblemente conectada via PPP (a juzgar por el nombre inscrito en DNS) se envió el mensaje a `server.transtek.com.tw`, quien torpemente aceptó el mensaje para entrega a terceros. Dadas las diferentes direcciones IP de las dos máquinas, y el hecho que sus dominios no coinciden, es un caso de un spammer que aprovecha una máquina con relay abierto para distribuir su basura. Lo que corresponde en este caso es dirigirse al alias semi-estándar abuse o al estándar postmaster, vale decir, enviar el mensaje con los encabezados *completos* a abuse@server.transtek.com.tw y a abuse@mclass.broadwing.net, y a los respectivos postmaster indicando el abuso que se está cometiendo. Con los encabezados el ISP del spammer puede determinar el usuario con sus registros, tomando las medidas del caso. Esperamos que el incauto usado como punto de distribución se dé cuenta del problema que se le está causando y corrija su configuración.

2.3 Entrevista con un MTA

La mayor parte de los protocolos de más alto nivel en Internet son protocolos basados en texto sobre TCP, con comandos y respuestas. De esta forma, siempre es posible usar `telnet` como herramienta de diagnóstico. La figura 3 muestra una amena conversación con un MTA, la figura 4 muestra el mensaje resultante. Las consultas y comandos son texto, las respuestas del MTA vienen precedidos por un código que indica la situación al cliente. En este caso particular (`sendmail-8.12.1` con configuración bastante estándar) además del código según RFC821 va una explicación en texto. En general, códigos `2xx` son informativos, `3xx` solicitan entrada del cliente, `4xx` son errores, mientras `5xx` son errores fatales (errores de sintaxis, comandos no implementados). En nuestro caso, el MTA habla ESMTP, y ofreció ENHANCEDSTATUSCODES (línea 10), por lo que además nos da códigos detallados según RFC2034 (por ejemplo, el 2.1.0 de la línea 20). Acá las líneas indentadas son simplemente continuación de líneas muy largas que fue necesario

```
 1  [vonbrand@tigger vonbrand]$ telnet pincoya.inf.utfsm.cl smtp
 2  Trying 200.1.19.3...
 3  Connected to pincoya.inf.utfsm.cl.
 4  Escape character is '^]'.
 5  220 pincoya.inf.utfsm.cl ESMTP Sendmail 8.12.1/8.12.1;
 6      Mon, 22 Oct 2001 11:05:04 -0300
 7  ehlo tigger.valparaiso.cl
 8  250-pincoya.inf.utfsm.cl Hello IDENT:vonbrand@laptop13.inf.utfsm.cl
 9      [200.1.19.247], pleased to meet you
10  250-ENHANCEDSTATUSCODES
11  250-PIPELINING
12  250-8BITMIME
13  250-SIZE 2000000
14  250-DSN
15  250-ETRN
16  250-AUTH GSSAPI
17  250-DELIVERBY
18  250 HELP
19  mail from: vonbrand@valparaiso.cl
20  250 2.1.0 vonbrand@valparaiso.cl... Sender ok
21  rcpt to: vonbrand@pincoya.inf.utfsm.cl
22  250 2.1.5 vonbrand@pincoya.inf.utfsm.cl... Recipient ok
23  data
24  354 Enter mail, end with "." on a line by itself
25  To: victima@inf.utfsm.cl
26  Subject: Mensaje de prueba
27
28  Como estai, loco?
29  .
30  250 2.0.0 f9ME53av029531 Message accepted for delivery
31  quit
32  221 2.0.0 pincoya.inf.utfsm.cl closing connection
33  Connection closed by foreign host.
```

Figura 3: Conversación en ESMTP

cortar. Las líneas que comienzan con números son respuestas del MTA, las demás son comandos y consultas.

Paso a paso: En la línea 1 invocamos `telnet` para conectarnos al MTA. Las líneas 2 a 4 son de este programa, al igual que la línea 33 final. En la línea 5 nos saluda el MTA: Da el nombre de la máquina, indica que habla ESMTP, y amablemente nos dice que se trata de `sendmail-8.12.1`, con configuración versión `8.12.1`. En la línea 7 nos identificamos (EHLO es *Extended HELO*, indica que deseamos hablar ESMTP). La respuesta (línea 8) indica que nuestra mentira no llegó lejos acá... Las líneas 10 a 18 dan los comandos que el MTA está dispuesto a aceptar. La línea 13 indica el límite del tamaño de mensajes aceptables. En 19 indicamos que `vonbrand@valparaiso.cl` envía correo, lo que se considera aceptable (línea 20). El mensaje va dirigido a `vonbrand@pincoya.inf.utfsm.cl`, lo que es aceptable (líneas 21 y 22). En 23 indicamos que estamos listos para enviar el mensaje (podríamos haber dado más receptores del mensaje), la respuesta 24 da indicaciones sobre qué hacer. Las líneas 25–28 son el mensaje propiamente tal, cerrado por '.' en la línea 29. Las líneas 31 y 32 son la despedida (podríamos haber dado más mensajes a entregar en esta transacción), como corresponde a una conversación civilizada.

La conversación entre dos MTAs o un MSA y un MTA es muy similar a la indicada.

El mensaje resultante con los encabezados completos es la figura 4, nuevamente con líneas demasiado largas plegadas. Queda claro que nuestros intentos de ocultar el origen del mensaje fueron infructuosos. También puede

```
 1  Return-Path: vonbrand@valparaiso.cl
 2  Delivery-Date: Mon Oct 22 11:09:40 2001
 3  Return-Path: <vonbrand@valparaiso.cl>
 4  Received: from pincoya.inf.utfsm.cl
 5            (IDENT:root@pincoya.inf.utfsm.cl [200.1.19.3])
 6        by inti.inf.utfsm.cl (8.11.6/8.11.6)
 7            with ESMTP id f9ME9aW24995
 8        for <vonbrand@inti.inf.utfsm.cl>;
 9            Mon, 22 Oct 2001 11:09:37 -0300
10  Received: from tigger.valparaiso.cl
11            (IDENT:vonbrand@laptop13.inf.utfsm.cl [200.1.19.247])
12        by pincoya.inf.utfsm.cl (8.12.1/8.12.1)
13            with ESMTP id f9ME53av029531
14        for vonbrand@pincoya.inf.utfsm.cl;
15            Mon, 22 Oct 2001 11:08:39 -0300
16  Date: Mon, 22 Oct 2001 11:05:04 -0300
17  From: vonbrand@valparaiso.cl
18  Message-Id: <200110221408.f9ME53av029531@pincoya.inf.utfsm.cl>
19  To: victima@inf.utfsm.cl
20  Subject: Mensaje de prueba
21
22  Como estai, loco?
```

Figura 4: El mensaje entregado

verse qué partes del mensaje recibido provienen directamente de la conversación entre el MUA y el MTA.

Los MUA en UNIX tienen alguna forma de solicitar que se muestre la conversación con el MSA. En el caso del venerable `mail` un ejemplo es la figura 5 (nuevamente, se han plegado líneas demasiado largas). En la figura las

```
 1  [vonbrand@sleipnir vonbrand]$ mail -v vonbrand
 2  Subject: Nada util
 3  Simplemente un mensaje al azar con un
 4  From totalmente gratuito
 5  para mostrar que es lo que el MDA hace con el.
 6  --
 7  Horst von Brand                      vonbrand@sleipnir.valparaiso.cl
 8  Casilla 9G, Vin~a del Mar, Chile                        +56 32 672616
 9  Cc:
10  vonbrand... Connecting to localhost via relay...
11  220 sleipnir.valparaiso.cl ESMTP Sendmail 8.12.1/8.12.1;
12      Mon, 22 Oct 2001 21:21:27 -0300
13  >>> EHLO sleipnir.valparaiso.cl
14  250-sleipnir.valparaiso.cl Hello IDENT:vonbrand@localhost [127.0.0.1],
15      pleased to meet you
16  250-ENHANCEDSTATUSCODES
17  250-PIPELINING
18  250-EXPN
19  250-VERB
20  250-8BITMIME
21  250-SIZE 2000000
22  250-DSN
23  250-ETRN
24  250-AUTH GSSAPI
25  250-DELIVERBY
26  250 HELP
27  >>> MAIL From:<vonbrand@sleipnir.valparaiso.cl>
28      SIZE=267 AUTH=vonbrand@sleipnir.valparaiso.cl
29  250 2.1.0 <vonbrand@sleipnir.valparaiso.cl>... Sender ok
30  >>> RCPT To:<vonbrand@sleipnir.valparaiso.cl>
31  >>> DATA
32  250 2.1.5 <vonbrand@sleipnir.valparaiso.cl>... Recipient ok
33  354 Enter mail, end with "." on a line by itself
34  >>> .
35  250 2.0.0 f9N0LRXM027628 Message accepted for delivery
36  vonbrand... Sent (f9N0LRXM027628 Message accepted for delivery)
37  Closing connection to localhost
38  >>> QUIT
39  221 2.0.0 sleipnir.valparaiso.cl closing connection
```

Figura 5: Conversación entre `mail` y `sendmail`

líneas 2 a 9 son lo que el usuario escribe, las demás son la salida de este comando. Las líneas marcadas con >>> son lo que `mail` escribe, las que comienzan con dígitos son las respuestas. La única diferencia entre el mensaje recibido dado en la figura 4 y el mensaje en un *mbox* es una línea que comienza From seguido por un espacio (el encabezado es From:) indicando el origen y el instante de entrega. Esta línea es usada por los MAA para determinar los límites de los mensajes, como se ve en el ejemplo de la figura 6, línea 1. Para evitar problemas, el MDA modifica líneas que comienzen con From, como en la línea 19. Nótese también que aparecen dos encabezados Received: `sendmail` actuó como MSA y luego como MTA, a pesar que el correo era puramente local en este caso.

3 Configurando `sendmail`

El programa `sendmail` es controlado por un archivo de configuración, el vilipendiado *sendmail.cf*. Como ya se indicó, `sendmail` fue concebido para manejar correo electrónico en un ambiente extremadamente complejo, lo que explica en


```
 1  From vonbrand@sleipnir.valparaiso.cl  Mon Oct 22 21:21:32 2001
 2  Return-Path: <vonbrand@sleipnir.valparaiso.cl>
 3  Received: from sleipnir.valparaiso.cl
 4            (IDENT:vonbrand@localhost [127.0.0.1])
 5        by sleipnir.valparaiso.cl (8.12.1/8.12.1)
 6            with ESMTP id f9N0LRXM027628
 7        for <vonbrand@sleipnir.valparaiso.cl>;
 8            Mon, 22 Oct 2001 21:21:31 -0300
 9  Received: (from vonbrand@localhost)
10        by sleipnir.valparaiso.cl (8.12.1/8.12.1) id f9N0LRWg027627
11        for vonbrand; Mon, 22 Oct 2001 21:21:27 -0300
12  Date: Mon, 22 Oct 2001 21:21:27 -0300
13  From: Horst von Brand <vonbrand@sleipnir.valparaiso.cl>
14  Message-Id: <200110230021.f9N0LRWg027627@sleipnir.valparaiso.cl>
15  To: vonbrand@sleipnir.valparaiso.cl
16  Subject: Nada util
17
18  Simplemente un mensaje al azar con un
19  >From totalmente gratuito
20  para mostrar que es lo que el MDA hace con el.
21  --
22  Horst von Brand                      vonbrand@sleipnir.valparaiso.cl
23  Casilla 9G, Vin~a del Mar, Chile                        +56 32 672616
```

Figura 6: Un mensaje en un *mbox*

buena parte la complejidad de la configuración. Súmese a ésto que este archivo se basa en programar la transformación de las direcciones mediante reglas de reescritura. Éstas son una manera de programar adecuada para manipular strings, el funcionamiento es similar a búsqueda y reemplazo en un editor de texto; pero totalmente desacostumbrada para los no especialistas. El formato del archivo fue diseñado específicamente para ser procesado eficientemente, tarea que se efectúa frecuentemente. Por tanto, resulta bastante ilegible.

En los planes de los desarrolladores estuvo "desde siempre" la idea de crear un lenguaje de alto nivel para ocultar esto. Nunca se materializó realmente. Sin embargo, con la simplificación del mundo es posible hoy día usar configuraciones bastante similares para todos los casos. La distribución trae maquinaria de configuración basada en el procesador de macros `m4`, la cual con su gran variedad de piezas preconstruidas debiera cubrir la inmensa mayoría de las necesidades.

3.1 Una introducción somera a `m4`

El procesador de macros `m4` (como su nombre lo indica, fue el cuarto intento de crear uno) es simple, pero muy poderoso. La maquinaria de configuración en `sendmail` usa bastantes de sus funcionalidades más avanzadas, pero para escribir una configuración maestra sólo se necesitan algunas características simples.

Las macros en `m4` pueden no tener argumentos, en cuyo caso basta mencionar la macro para que ésta sea expandida. Para evitar que una palabra sea interpretada como macro se la debe `'citar'`. Está la posibilidad de usar macros con argumentos, que se dan entre paréntesis separados por comas. Para definir una macro sin argumentos se usa la macro predefinida `define()`, para eliminar una posible definición se usa `undefine()`. Es legal eliminar una definición inexistente, esta operación no tiene efecto alguno.

Nótese que `m4` es un procesador de macros, por lo que cualquier texto que no sea expandido permanece sin cambios en el resultado. En particular, los saltos de línea permanecen. La macro predefinida `dnl` elimina desde el punto en que se invoca hasta antes del comienzo de la siguiente línea, por lo que se usa para eliminar saltos de línea indeseados y como una manera de escribir comentarios.

3.2 Un ejemplo de configuración maestra

Un ejemplo de archivo de configuración maestro es la figura 7. Las macros empleadas son muy variadas, es necesario referirse al archivo *cf/README* para el detalle completo.

Las macros empleadas se dividen en varios grandes grupos:

- Hay macros particulares para usar una única vez, como *OSTYPE* para definir el sistema operativo, *VERSIONID* para dar un número de versión a la configuración, entre otros.
- Las macros *conf** son valores que se pueden definir en el archivo maestro. Más que nada proveen valores para ciertas opciones.
- Las *FEATURE* dan la opción de agregar ciertas opciones a la configuración. Hay una larga lista de éstas.
- Los *HACK* dan opciones, que en sí no son muy limpias y por esta razón se mantienen segregadas.
- Los *MAILER* permiten definir `mailers`, vale decir, mecanismos a quienes entregar correo para su proceso posterior.

El archivo maestro de la figura 7 es un ejemplo simple. La línea 1 incluye un archivo con definiciones de las macros que se usan después. Las líneas 2 y 4–18 definen una variedad de parámetros, algunas de ellas están comentadas (12,


```
 1  include('../m4/cf.m4')dnl
 2  define('confDEF_USER_ID',``8:12'')dnl
 3  OSTYPE('linux')dnl
 4  define('SMART_HOST', 'chac.inf.utfsm.cl')dnl
 5  define('confMAX_MESSAGE_SIZE', 2000000)dnl
 6  define('confTO_CONNECT', '1m')dnl
 7  define('confFALLBACK_MX', 'chac.inf.utfsm.cl')dnl
 8  define('PROCMAIL_MAILER_PATH',"/usr/bin/procmail")dnl
 9  define('ALIAS_FILE',"/etc/aliases")dnl
10  define('STATUS_FILE', "/var/log/sendmail.st")dnl
11  define('confUSERDB_SPEC', "/etc/mail/userdb.db")dnl
12  dnl define('confPRIVACY_FLAGS', 'authwarnings,novrfy,noexpn')dnl
13  define('confTO_QUEUEWARN', '1d')dnl
14  dnl define('confTO_QUEUERETURN', '5d')dnl
15  dnl define('confQUEUE_LA', '12')dnl
16  dnl define('confREFUSE_LA', '18')dnl
17  define('confTRY_NULL_MX_LIST',true)dnl
18  define('confDONT_PROBE_INTERFACES',true)dnl
19  dnl This changes sendmail to only listen on the loopback device 127.0.0.1
20  dnl and not on any other network devices. Comment this out if you want
21  dnl to accept email over the network.
22  DAEMON_OPTIONS('Port=smtp,Addr=127.0.0.1, Name=MTA')
23  FEATURE('smrsh',"/usr/sbin/smrsh")dnl
24  FEATURE('mailertable',`hash -o /etc/mail/mailertable')dnl
25  FEATURE('virtusertable',`hash -o /etc/mail/virtusertable')dnl
26  FEATURE(redirect)dnl
27  FEATURE(always_add_domain)dnl
28  FEATURE(use_cw_file)dnl
29  FEATURE(local_procmail)dnl
30  FEATURE('access_db")dnl
31  FEATURE('blacklist_recipients")dnl
32  FEATURE('access_db")dnl
33  FEATURE('blacklist_recipients")dnl
34  MAILER(smtp)dnl
35  MAILER(procmail)dnl
```

Figura 7: Ejemplo de configuración maestra de `sendmail`

14–16). La línea 3 incluye definiciones propias de Linux. Las líneas 19 a 21 son comentarios, indican que el efecto de la línea 22 es sólo permtir recepción de correo generado localmente. Las líneas 23 a 33 activan opciones. Estas dependen de las macros *conf**, por lo que tales macros deben definirse antes. Finalmente, las líneas 34 y 35 agregan soporte para manejar correo SMTP (esto siempre debiera estar definido) y uso de `procmail` para entrega local de correo.

En `sendmail-8.12.1` se genera además un archivo */etc/mail/submit.cf* desde *submit.mc*. Este archivo es la configuración de `sendmail` en su función de MSA, por lo que la configuración prácticamente no ofrece opciones.

Se recomienda crear una configuración maestra para cada máquina (o cada clase de máquinas), y guardarla. Esto permite regenerar la configuración detallada fácilmente cuando se instala una nueva versión de `sendmail`.

3.3 Otros archivos de configuración necesarios

Además de los archivos de configuración *sendmail.cf* y *submit.cf* hacen falta varios más.

Debe haber alguna manera de manejar alias de correo, tradicionalmente a través del archivo */etc/mail/aliases* (la ubicación exacta está definida en *sendmail.cf*, y hay una colección de posibilidades para obtenerlos vía NIS y otros sistemas).

En */etc/mail/local-host-names* va la lista de dominios correo a los cuales se maneja localmente. Típicamente todos los alias de la máquina, y todas las denominaciones que la tengan como MX de máxima prioridad.

El archivo */etc/mail/access* permite controlar el uso del MTA para entrega de correo. La configuración de éste es crítica, dado que aceptar correo indiscriminadamente para entrega fuera del sitio es una invitación abierta a los distribuidores de spam. Generalmente se debe aceptar correo para entrega fuera del sitio sólo de máquinas locales, y viceversa. Por omisión, versiones más nuevas de `sendmail` no permiten relaying, hay que autorizarlo explícitamente.

3.4 Problemas típicos de configuración

Un problema común es que `sendmail` se demore una eternidad en subir, haciendo que el booteo sea muy lento. Esto sucede comúnmente porque `sendmail` no es capaz de obtener el nombre de la máquina en que corre vía DNS. La solución es inscribirla correctamente, o agregar el nombre en */etc/hosts*.

Es frecuente ver que mensajes rebotan con indicación de “mail loops back” o similar. Esto es el resultado de dar un registro MX de máxima prioridad, pero no agregar el nombre en */etc/mail/local-host-names*.

Con la configuración por omisión de Red Hat al menos, que da `DAEMON_OPTIONS` tales que sólo se acepta correo local (ver la figura 7), no se puede recibir correo desde otras máquinas, pero sí enviarlo.

Instalar una versión nueva de `sendmail` en un sistema más antiguo típicamente resulta en que `sendmail` simplemente no parta. La razón es que nuevas versiones son sumamente cuidadosas con los dueños, grupos, y permisos de archivos clave y los directorios que llevan a ellos. Si éstos dan la posibilidad que usuarios no autorizados (básicamente, no *root*) modifiquen alguno de ellos, `sendmail` simplemente falla.

4 Estándares relevantes para el correo electrónico

Como resulta obvio por la importancia del correo electrónico y los múltiples usos que se le han dado, hay una gran colección de estándares relevantes. Los más fundamentales son los siguentes.

RFC822 describe la sintaxis de los mensajes de correo electrónico. Ha sufrido múltiples actualizaciones (RFC1123,

1138, 1148, 1327, 2156). RFC1123 (actualizado por RFC2181) describe los requisitos de una máquina en Internet. RFC821 define el protocolo SMTP (*Simple Mail Transfer Protocol*), que fue luego extendido a ESMTP (*Extended SMTP*) por RFC1869, 1870, 1891, y 1985. RFC974 describe los registros MX de DNS que usa el sistema de correo. Estas son las especificaciones según las cuales se escribió `sendmail`.

Otros RFCs relevantes para el sistema de correo incluyen:

- RFC1428: Manejo de caracteres no-ASCII
- RFC1652: Extensiones de SMTP para 8BITMIME
- RFCs 1891 – 1894: Los mensajes de rebote
- RFC1985: Iniciar la cola remotamente
- RFC2033: LMTP, *Local Mail Transport Protocol*
- RFC2034: Códigos de error extendidos de SMTP
- RFCs 2045 – 2049: Extensiones MIME (actualizados por RFC2184, 2231)
- RFC2476: Especificaciones de MSA
- RFC2487: SMTP seguro sobre TLS (TLS es *Transport Layer Security*, definido por RFC2246. Corresponde a mecanismos para establecer conexiones seguras entre máquinas a traves de Internet)
- RFC2554: Extensiones de SMTP para autentificación
- RFC2852: Entrega por extensión SMTP
- RFC2920: Extensiones de SMTP para encadenar comandos

En todo caso, `sendmail` vive en un mundo más amplio, por lo que ofrece opciones que van más allá de estos estándares. Puede también configurarse para medios de transporte distintos de TCP/IP, cosa que obviamente no describen los estándares de Internet.

Referencias

[1] Dan J. Bernstein. The `qmail` home page. <`http://pobox.com/~djb/qmail.html`>.

[2] Brian Costales and Eric Allman. *sendmail*. O’Reilly & Associates, 1997.

[3] The `exim` home page. <`http://www.exim.org`>.

[4] Evi Nemeth, Garth Snyder, and Trent R. Hein. *UNIX System Administration Handbook*. Prentice-Hall, third edition, 2000.

[5] Eric S. Raymond. fetchmail. <`http://www.tuxedo.org/~esr/fetchmail/`>.

[6] The `sendmail` organization. <`http://www.sendmail.org`>.

[7] Wietse Venema. The `postfix` home page. <`http://www.postfix.org`>.