Agente de correo

POSTFIX

Asignatura: Administración de Sistemas
y Servicios Telemáticos

Alumno: Rubén Salas Casado

Índice

1. Objetivos y descripción del servicio.

Lo primero de todo: ¿qué es *Postfix*? Según su autor, Wietse Veneman, es un intento de conseguir una alternativa al tan usado programa *Sendmail*, el cual es el principal encargado de entregar los correos electrónicos que circulan a diario por Internet.

Postfix intenta ser rápido, fácil de administrar, suficientemente seguro y al mismo tiempo compatible con el programa *Sendmail.* Además, *Postfix* es un directo competidor de *qmail*; pero "competidor" no es "enemigo". Esta competencia amistosa ayudará a mejorar ambos programas.

Algunos de los objetivos principales de *Postfix* son:

- **Amplia difusión.** *Postfix* debe ser usado por muchas personas para causar un impacto significativo en el funcionamiento y seguridad del correo electrónico en Internet. Por lo tanto, el software es ofrecido gratis.

- **Funcionamiento.** *Postfix* es hasta tres veces más rápido que su más cercano competidor. Un PC de escritorio corriendo *Postfix* puede recibir y enviar un millón de mensajes distintos al día. *Postfix* usa procedimientos de servidores web para reducir la creación de procesos y usa otros procedimientos para reducir el *overhead* del sistema de archivos sin perder fiabilidad.

- **Compatibilidad.** *Postfix* está diseñado para ser compatible con *sendmail* y así hacer una migración hacia el primero de una manera sencilla. *Postfix* admite `/var[/spool]/mail`, `/etc/aliases`, NIS y los archivos `~/.forward`. También intenta ser fácil de administrar, por lo que no usa `sendmail.cf`.

- **Seguridad y robustez.** *Postfix* está diseñado para comportarse racionalmente bajo situaciones complicadas. Cuando el sistema local se queda sin espacio en el disco o sin memoria libre, *Postfix* se desactiva en vez de agravar el problema. Por características de diseño, el programa *Postfix* no se colapsa cuando el número de mensajes aumenta; está diseñado para permanecer controlado.

- **Flexibilidad.** *Postfix* está construido sobre varios pequeños programas que realizan sólo una específica tarea cada uno: recibir un mensaje vía SMTP, entregar un mensaje vía SMTP. Entregar un mensaje localmente, reescribir una dirección,… También es fácil desactivar funcionalidades: cortafuegos y clientes no necesitan entrega local.

- **Seguridad.** *Postfix* usa múltiples capas de defensa para proteger el sistema local contra intrusos. Casi todos los demonios de *Postfix* pueden ejecutarse con bajos privilegios. No hay un camino directo desde la red hasta los programas de entrega local sensibles a la seguridad; un intruso tiene que atravesar muchos otros programas primero. Incluso *Postfix* no confía en los contenidos de su propia cola de archivos, ni en los contenidos de sus propios mensajes IPC. Además, *Postfix* no se ejecuta como *set-uid*.

Otras características de interés son:

- **Transportes múltiples.** Los sistemas *sendmail* tienen que ser configurados para que puedan funcionar entre Internet, DECnet, X.400 y UUCP. *Postfix* está diseñado para ser lo suficientemente flexible como para que pueda operar en tales entornos sin requerir un dominio virtual o algún tipo de alias. Sin embargo, la versión inicial sólo habla con SMTP y sólo ha limitado su soporte a UUCP.

- **Dominios virtuales.** En la mayoría de los caso, añadiendo el soporte para un dominio virtual requiere el cambio a sólo una tabla *lookup* de *Postfix*. Otros agentes de correo normalmente necesitan niveles múltiples de aliases o redirecciones para conseguir el mismo resultado.

- **Control de UCE (Unsolicited Commercial Email; correo publicitario no solicitado).** *Postfix* admite restricciones en cuanto al correo entrante. Implementa los aspectos habituales (listas negras, tablas de búsqueda DNS HELO/sender,…), aunque el filtrado de contenido todavía no ha sido implementado.

- **Tablas *lookups*.** *Postfix* todavía no ha implementado un lenguaje de reescritura de direcciones. En su lugar, hace un uso extensivo de las tablas *lookpus.* Las tablas pueden ser archivos locales **dbm** o **db**, o archivos **NIS** o mapas **NetInfo** en la red. Añadir soporte para otros mecanismos de *lookup* es relativamente fácil.

2. Requisitos.

Postfix no necesita ningún hardware específico adicional para funcionar. Tampoco tiene limitaciones por las características técnicas (velocidad, memoria,…) de la máquina donde se esté ejecutando.

3. Arquitectura del servicio.

Postfix está basado en procesos semi-residentes que cooperan mútuamente para realizar tareas específicas los unos a los otros, sin ningún tipo de relación jerárquica *padre-hijo*. De nuevo, trabajando con procesos separados se obtiene un mejor aislamiento que usando un programa grande. Además, *Postfix* aprovecha la ventaja de que un servicio con esa reescritura de direcciones está disponible para cualquier programa-componente del programa *Postfix*, sin incurrir en el coste de la creación del proceso por sólo reescribir una dirección.

Postfix está implementado como un servidor maestro residente que corre procesos demonio de *Postfix* bajo demanda: el demonio procesa enviar o recibir mensajes de correo de la red; el demonio procesa entregar localmente el correo, etc. Estos procesos son creados hasta un número configurable, y son reusados durante un número configurable de veces, y mueren tras pasar una cantidad configurable de tiempo *idle*. Este acercamiento reduce drásticamente el *overhead* de creación de procesos.

Postfix intenta ser el sustituto de *Sendmail*. Por esta razón, intenta ser compatible con la infraestructura ya existentes. Sin embargo, muchas partes del sistema Postfix, tales como el programa de entrega local, son fácilmente reemplazadas editando un archivo de configuración del estilo del `inetd`. Por ejemplo, podemos usar un programa de entrega local alternativo que funcione con unos privilegios mínimos para los usuarios de POP/IMAP que nunca han entrado en la *shell*, y que posiblemente no tengan cuenta UNIX.

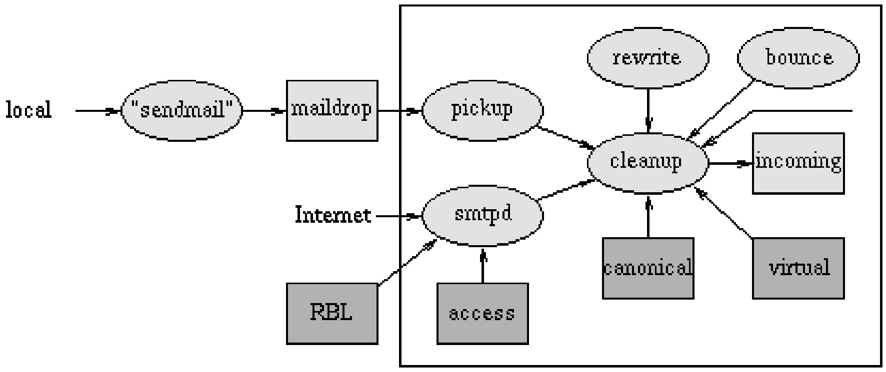
Como resultado de esta arquitectura, *Postfix* es fácil de desmenuzar. Los subsistemas que están desactivados no pueden ser vulnerados ni atacados. Los cortafuegos no necesitan entrega local. En las estaciones-cliente, uno desactiva el *smtp listener* y los subsistemas de entrega local; o el cliente monta el directorio `maildrop` de un servidor de archivos, y ejecuta de todas maneras procesos *Postfix* no residentes.

Ahora hablemos un poco de la **comunicación entre procesos** *Postfix*. El núcleo del sistema *Postfix* está implementado por una docena de programas semi-residentes. Por razones de privacidad, estos procesos *Postfix* se comunican vía sockets de dominio UNIX, o con FIFOs que funcionan en un directorio protegido. En lugar de esta privacidad, los procesos de *Postfix* realmente no confían en los datos que han recibido de esta manera, exactamente igual que los contenidos de los archivos de colas de *Postfix*.

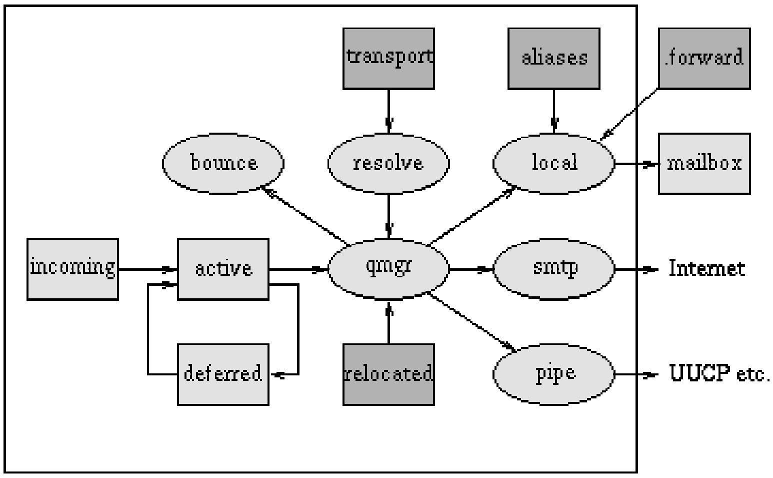
La cantidad de información intercambiada entre los procesos *Postfix* está limitada. En muchos casos, sólo la información intercambiada entre procesos *Postfix* es un nombre de un archivo de cola y una lista de personas a las que va dirigida el correo electrónico, o alguna información sobre el estado del correo. Una vez que un mensaje es salvado a un archivo, ahí permanece hasta que sea leído por un programa de entrega de e-mails.

Postfix tiene un especial cuidado con las pérdidas de información: `flush` y `fsync()` todos los datos antes de un ACK de recepción de mensaje, y comprueba todos los resultados de las llamadas para condiciones de error. Este estilo de programar puede ser nuevo para algunas personas, pero se puede asegurar que ha sido una práctica habitual durante años en muchos lugares.

Podemos ver en las siguientes figuras los componentes principales del sistema *Postfix*, y los principales flujos de información entre ellos. Primero se muestra el sistema desde que llega el mensaje hasta que se prepara para su entrega:



En la segunda figura se muestra el subsistema encargado de la entrega del correo electrónico:



En ambas figuras, las elipses amarillas son programas de correo; las cajas amarillas son archivos o colas de correo; las cajas azules son tablas *lookup*. Los programas que están dentro de las cajas grandes están controlados por el demonio maestro residente de *Postfix*. Los datos dentro de esas mismas cajas son propiedad del sistema de correo *Postfix*.

4. Configuración.

Como en la configuración de *Postfix* nos podríamos extender mucho, vamos a intentar ser lo más breve en cada explicación, sin perder claridad.

Todo el sistema *Postfix* se configura desde el archivo `main.cf`. En este archivo aparecen cerca de 100 parámetros, aunque para hacer funcionar todo el sistema nos valdrá con modificar sólo dos o tres parámetros. Veamos los parámetros más importantes del archivo `main.cf`, dando valores como ejemplo:

Dominio del correo saliente. Se define con otra variable que será definida más adelante:

```
myorigin = $myhostname
```

Dominios que la máquina no reenviará a otra máquina para ser enviado.

```
mydestination = $myhostname localhost.$mydomain
```

Cuándo avisamos al administrador de correo. Tenemos varias clases de alarmas: `bounce, 2bounce, delay, policy, protocol, resource, software.`

```
notify_classes = resource, software
```

Nombre de la máquina.

```
myhostname = descartes.micasa.es
```

Nombre del dominio.

```
mydomain = micasa.es
```

Mis propias redes. Se pueden definir rangos con el carácter "/", como en el ejemplo.

```
mynetworks = 168.100.189.0/28, 127.0.0.0/8
```

Mis propias direcciones de red. Me permite tener dominios virtuales. Se suele dejar el valor por defecto:

```
inet_interfaces = all
```

Límite de procesos.

`default_process_limit = 50` (valor por defecto. Demasiado grande para pequeñas máquinas).

Concurrencia en el destino. Define los límites inicial, por defecto y del mismo destinatario local. Los valores son el número de conexiones simultáneas, no de procesos (de eso se encarga el parámetro anterior).

```
initial_destination_concurrency = 2
local_destination_concurrency_limit = 2
default_destination_concurrency_limit = 10
```

Límites en el tamaño de los objetos. Tenemos varios parámetros para estas limitaciones. Presentamos los valores por defecto.

`line_length_limit` = 2048 (bytes)
`header_size_limit` = 102400 (bytes)
`message_size_limit` = 10240000 (bytes)

`message_size_limit` = sin restricción
`bounce_size_limit` = 50000 (bytes)

Límites de tiempo. Al invocar programas externos, les ponemos un tiempo límite. Se establece un tiempo por defecto, que se puede modificar individualmente con cualquier programa:
`command_time_limit = 1000` (segundos; por defecto)
programa-externo`_time_limit` (`$command_time_limit` por defecto)

Bases de datos de aliases. Veamos algunos ejemplos:
`alias_maps = hash:/etc/aliases`
`alias_maps = dbm:/etc/aliases, nis:mail.aliases`
`alias_database = hash:/etc/aliases` (4.4BSD, LINUX)
`alias_database = dbm:/etc/aliases` (4.3BSD, SYSV<4)
`alias_database = dbm:/etc/mail/aliases` (SYSV4)

El resto de los parámetros no se han comentado porque, aunque son importantes, sólo se tienen que modificar en unos casos muy específicos.

Después de cualquier modificación en el archivo de configuración `main.cf`, tenemos que cargar esta configuración con `postfix reload`.

5. Pruebas.

Las pruebas que haremos servirán para comprobar que hemos configurado como queríamos el sistema. Lo primero es comprobar que hemos modificado correctamente el archivo `main.cf`. Para ello, podemos usar una utilidad del *Postfix* llamada `postconf`. Este programa mostrará por pantalla todas las variables de nuestro sistema *Postfix.*

Después las pruebas que haremos serán las mismas que haríamos a cualquier agente de correo. Seguiremos este orden, por si se produce algún error para poder detectarlo antes:

- Comprobar que mandamos y recibimos mensajes en nuestra propia máquina.
- Comprobar que mandamos y recibimos mensajes en nuestro propio dominio.
- Comprobar que mandamos y recibimos mensajes a/del exterior.

Si se han realizado con éxito estas pruebas, ahora tenemos que comprobar que los valores que hemos escogido para los parámetros de configuración son correctos. Veremos si las direcciones en las cabeceras son correctas, tanto para el correo entrante como para el saliente; veremos qué ocurre cuando mandamos un mensaje a un usuario inexistente, o un mensaje excesivamente grande…

Es muy probable que después de estas pruebas tengamos que reajustar algunos parámetros si queremos dejar nuestro sistema totalmente a nuestro gusto. En general, no habrá que modificar casi ningún parámetro, ya que los valores por defecto son apropiados para la mayoría de los sistemas.

6. Gestión diaria.

Aunque la gestión no será muy dura para el administrador, siempre tiene que estar atento a cualquier imprevisto. El correo electrónico es un medio que usa la mayoría de los usuarios, y con mucha intensidad. Por lo tanto, no podemos permitir fallos en el sistema de correo.

Tenemos que estar atentos a que no *reboten* mensajes; que no se manden mensajes excesivamente grandes en tamaño (si no queremos que sean tan grandes, podemos reducir la limitación en el archivo de configuración `main.cf`); a que si hay mucho tráfico, revisar las limitaciones de procesos, conexiones… en el archivo de configuración, etc.

Como cada usuario es un mundo, podemos encontrarnos con situaciones dispares. Por eso, cuanto más robusta sea la configuración para que nuestra máquina soporte todo el tráfico de mensajes sin sobrecargarla, menos mantenimiento será necesario, ya que con las limitaciones que se pueden configurar se puede hacer que no se *caiga* nuestra máquina.

Algo que sí es importante es el manejo de aliases. En muchos sistemas se da un gran uso de ellos. Si la lista es muy grande, necesita un mantenimiento si se dan de alta/baja a usuarios muy a menudo. No podemos permitir redirigir el correo mediante un alias a un usuario que ya no existe.

También es importante revisar "por encima" los archivos *logs* que produce el *Postfix*, ya que nos puede mostrar errores que se están produciendo y no nos damos cuenta. Decimos que "por encima" porque si se maneja mucho tráfico de correos electrónicos es imposible leer todos los *logs*.

Y por último y como administrador, hay que estar atentos a los posibles agujeros de seguridad para evitar que con un *exploit* nos puedan atacar la máquina. Por eso conviene mantener actualizado (y parcheado si es necesario) nuestro sistema de correo, ya que es una pieza fundamental en cualquier sistema.