UNIVERSIDAD TÉCNICA DE AMBATO

FACULTAD DE INGENIERÍA EN SISTEMAS

GUÍA METODOLÓGICA PARA LA ADMINISTRACIÓN DE SERVIDORES DE CORREO ELECTRÓNICO EN GNU/LINUX

AUTOR: GUILLERMO MARCELO GUERRERO SÁNCHEZ

DIRECTOR: Ing. DAVID GUEVARA

ASESOR: Ing. M. Sc. FRANKLIN MAYORGA

Tesis de grado, previo a la obtención del título de Ingeniero en Sistemas

Ambato - Ecuador

Junio - 2005

DEDICATORIA

A mis inolvidables padres, a mis hermanos, a mis sobrinos

AGRADECIMIENTOS

A la Facultad de Ingeniaría en Sistemas de la Universidad Técnica de Ambato por su valiosa contribución en mi formación profesional y a todas aquellas persona que de una u otra manera me han apoyado en el transcurso de mi vida estudiantil y en el desarrollo de la presente tesis de grado.

DECLARACIÓN, AUTENTICIDAD Y RESPONSABILIDAD

Yo, Guillermo Marcelo Guerrero Sánchez

Número de cédula 180302333-0

Declaro que la investigación enmarcada en el diseño de la tesis es absolutamente original, autentica y personal. En tal virtud declaro que el contenido, efectos legales y académicos que se desprenden del trabajo de tesis son y serán de mi sola y exclusiva responsabilidad legal y académico.

_____

Guillermo Marcelo Guerrero Sánchez

INDICE DE CONTENIDO

INDICE DE TABLAS

INDICE DE FIGURAS

INTRODUCCIÓN

El sistema de correo electrónico es hoy día un elemento de comunicación básico para cualquier empresa. Disponer de un servicio de correo eficiente, robusto, y adaptado a las necesidades de una organización es esencial.

El servicio de correo electrónico ha sufrido relativamente pocos cambios a lo largo de los años; aun así, sigue siendo uno de los más utilizados por los usuarios de la red. Una red cada día está más sujeta a ataques y a intromisión de la privacidad, el correo electrónico no puede ser ajeno a las nuevas circunstancias, ya que los usuarios cada vez demandan un servicio mejor y más seguro sin perder flexibilidad y utilidad.

CONTENIDO

CAPÍTULO II. MARCO TEÓRICO

En este capítulo el correo electrónico se presenta como un sistema de mensajería y de la relación que guarda con las RFCs.

Esta parte ilustra los estándares y protocolos aplicables al correo electrónico y en concreto se encontrara en ella las descripciones de los servicios y mecanismos que hacen posible la entrega de mensajes de correo que son: sumisión, transmisión,

entrega, registros MX de DNS, acceso al buzón, listas de correo, servidor de noticias y los diferentes programas utilizados para la implementación de un servidor de correo.

Se presenta un análisis de spam, técnicas spam para el envío publicidad no solicitada a las cuentas de correo electrónico, servidores de noticias. También se indica algunos programas antispam existentes en el mercado.

En este apartado se indica además una clasificación de los virus y sus técnicas de infección, propagación y consecuencias. Además se presenta algunos programas antivirus comerciales y de libre distribución.

CAPÍTULO III. CONFIGURACIONES

Este capítulo corresponde a la implementación del sistema. El punto de partida es la configuración del DNS, siguiendo por la instalación de Sendmail como servidor de correo, configuración de los archivos para iniciar el servicio, prueba del servicio, activación de los demonios.

También se indica la configuración de fetchmail para recoger correo electrónico externo, y procmail para filtrado de correo no solicitado. Además de la instalación de programas antivirus y antispam, así como de su configuración para trabajar en forma independiente o su combinación para disminuir el riesgo de infección en la recepción de correo electrónico.

Adicionalmente se indica la configuración de un webmail para el acceso al correo electrónico desde un navegador Web.

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

Resume las conclusiones a las que se llegaron después de realizado el trabajo de investigación y las respectivas recomendaciones para su correcta implementación en una organización.

CAPÍTULO I. GENERALIDADES

1.1. PLANTEAMIENTO DEL PROBLEMA

Es muy común que los administradores inexpertos no se molesten siquiera en establecer un nivel de seguridad apropiado en sus redes locales, y mucho menos en el servidor de correo, el cual ven como un servicio más. Es un error común el configurar el servidor de correo para que permita enviar correo como sea a cualquier costo. Usualmente este costo significa convertirse en open relay, y por lo tanto en un paraíso para personas que se dedican al envío masivo de correo comercial (spam).

Los servicios SMTP tienen una larga historia de debilidades de seguridad, ya que permiten accesos al sistema general en términos de usarse como modo de envío de correo masivo. Esto permite que cualquiera en la Internet pueda enviar correo a cualquier otra persona o a grandes grupos de personas. Esta característica de SMTP es lo que hace posible el correo basura o spam. Los servidores SMTP modernos intentan minimizar este comportamiento permitiendo que sólo los hosts conocidos accedan al servidor SMTP. Los servidores que no ponen tales restricciones son llamados servidores open relay.

1.2. OBJETIVOS

1.2.1. Objetivo General

Generar una guía metodológica para la administración de servidores de correo electrónico aplicable a organizaciones a las empresas que posean un servidor Linux.

1.2.2. Objetivos Específicos

- Configurar un servidor de correo para disponer servicios de correo locales independientes de los que ofrece el ISP (Internet Service Provider, Proveedor de servicio de Internet).
- Identificar el usuario que accede a revisar su correo electrónico.
- Limitar tamaños de correo y máximo de conexiones
- Limitar el relay de correo
- Reducir drásticamente el SPAM: la entrada masiva de correo no solicitado, también denominado correo basura, usualmente utilizado para publicidad.
- Configurar programas antivirus para la detección de virus en correos entrantes y salientes.
- Administrar listas de correo y servidor de noticias.

1.3. JUSTIFICACIÓN

El correo electrónico, por su conveniencia, facilidad de uso y precio, se ha convirtiendo en una herramienta cada día más utilizada. Una de las mayores preocupaciones hoy en día son los virus que se reciben adjuntos a los correos electrónicos.

El correo basura o no solicitado es uno de los múltiples resultados de la inseguridad y fragilidad del correo electrónico en Internet. El correo electrónico clásico es un canal inseguro y fácilmente manipulable. Los protocolos de correo electrónico utilizados por la mayor parte de los usuarios no pueden garantizar la autentificación del emisor ni la confidencialidad del contenido. Pero esto no es inconveniente para que considerar la manipulación y falsificación de mensajes como una acción ilegal e indigna. No se debe olvidar que es necesario perseguir este tipo de actividades y una de las tareas prioritarias por parte de las instituciones que ofrecen el servicio, es informar de ello a los usuarios. Esta información deberá facilitarse a través de las políticas de uso propias de cada institución.

CAPÍTULO II. MARCO TEÓRICO

2.1. CORREO ELECTRÓNICO

El e-mail (Electronic Mail, correo electrónico) es el medio que permite enviar mensajes privados a otros usuarios de Internet que se encuentren en cualquier parte del mundo. Para ello, los usuarios de este servicio tienen una dirección electrónica que cumple el mismo objetivo que la dirección postal: poder enviar y recibir correspondencia.

El correo electrónico es muy popular, a tal punto que hoy en día el intercambio de mensajes constituye una porción importante del tráfico de Internet, siendo la principal razón (y también la primera), por la cual, la mayoría de las personas se conectan a la Red.

El servidor de correo se encarga de gestionar el correo, este puede atender miles de cuentas de correo y permite definir una cantidad ilimitada de buzones de correo electrónico dentro de un dominio.

Linux dispone de paquetes de software para tener correo electrónico. Este puede ser tanto local (entre usuarios de su sistema) como remoto (mediante una red TCP/IP

o UUCP). El software de e-mail consta normalmente de dos partes: un agente de usuario o mailer y un programa de transporte. El agente de usuario es el software que el usuario utiliza para crear mensajes, leerlos, etc. El programa de transporte es quien se ocupa de entregar correo tanto remoto como local, conociendo protocolos de comunicaciones y demás. El usuario nunca interactúa directamente con este programa, sino que lo hace a través del agente de usuario. Sin embargo, el administrador del sistema debe conocer como funciona el programa de transporte, con el fin de configurarlo según sus necesidades.

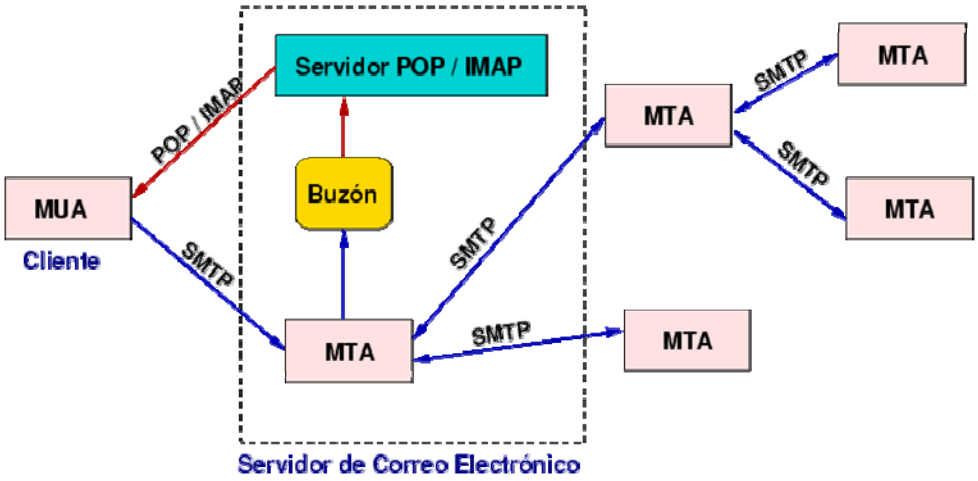

Figura 2.1. Representación esquemática del servicio de correo electrónico

2.1.1. VENTAJAS DEL CORREO ELECTRÓNICO

- **Costo**: El correo electrónico es mucho más barato que el correo postal. No importa la distancia que el mensaje electrónico deba recorrer para llegar al destino: ya sea Japón o una ciudad vecina, el costo es el mismo ya que en todos los casos representa el de una llamada local. Generalmente se calcula

el costo del correo electrónico en base al tiempo consumido para transferir el mensaje a través del ISP. Este tiempo de transferencia depende del tamaño del archivo: generalmente insume unos pocos segundos.

- **Versatilidad**: Además del cuerpo del texto, es posible adjuntar al mensaje cualquier tipo de archivo guardado en su computadora: revistas, planillas de cálculo, sonidos, fotos, etc. Para ello simplemente se debe codificar estos archivos de una forma especial (el programa de correo lo hace de manera automática). Los archivos enviados son despachados y recibidos en formato digital, lo cual permite que quien los reciba pueda modificarlos a su gusto.
- **Velocidad**: El correo electrónico es muy veloz y no tardará más de unos pocos minutos en llegar al destino. Pueden existir demoras en la lectura del mensaje, ya sea por la forma en que algunas empresas distribuyen internamente los mensajes electrónicos a cada destinatario final, o bien porque el destinatario mismo demora en leerlo.
- **Comodidad**: Quien recibe un mensaje puede responderlo en el momento que desee, sin la presión de tiempo que implica una llamada telefónica.

2.1.2. COMPONENTES DE UN MENSAJE.

Un mensaje de Correo Electrónico consta de dos partes. La primera se denomina Encabezado, la que contiene el mensaje en si, recibe el nombre de Cuerpo del mensaje.

El encabezado posee Información sobre el remitente, los destinatarios, la fecha de e 7nvío, el tema del mensaje, etc.

Las líneas más importantes del encabezado son:

- **From:** Es la dirección del remitente. Solo puede haber una línea de este tipo en el encabezado.
- **To:** El o los destinatarios de este mensaje. Esta línea puede especificar más de una dirección de destino.
- **Cc:** Copia a destinatarios. Esta equivale a la copia en papel carbón en el caso del correo normal. Se manda a los destinatarios indicados una copia de la carta.
- **Bcc:** Esta seria una copia oculta. Se mandara una copia a la dirección aquí indicada sin que los otros destinatarios tengan conocimiento de ello.
- **Subject:** Tema del mensaje. El texto es libre, pero es conveniente elegir uno que sea breve y que describa el contenido del mensaje. Teniendo cuidado con los signos de puntuación.
- **Organization:** La organización que posee la máquina desde la que se ha enviado el mensaje. Si la máquina usada es la suya propia no incluya este campo, o bien indique ``privado'' o cualquier trivialidad sin sentido. Este campo es opcional.
- **Date:** Indica la fecha y hora en que el mensaje fue enviado.
- **MessageId:** Es un identificador de cada mensaje, este es único y lo inserta la computadora que lo envía.

- **Received:** Es la Información que se utiliza para comprobar los problemas que hayan aparecido en el reparto de un mensaje. Se muestran las direcciones de las maquinas, junto con la fecha y hora en que el mensaje paso por ellas en dirección a su destino.
- **Resent-From:** Dirección de la persona o programa desde el cual llega el mensaje. El hecho de decir reenviado notifica que el mensaje le ha llegado a la persona que se indica en este campo y ella a su vez manda una copia.
- **Reply-To:** Obviamente, la dirección a la que se debe contestar. No tiene que ser la misma desde donde se ha enviado la carta.
- **X-cualquier-cosa*:*** Ningún programa relacionado con el correo debe protestar sobre cualquier encabezado que comience con X-. Esto se usa para implementar características adicionales que aún no han sido incluidas en un RFC, o que no lo serán nunca. Esto se usa, por ejemplo, en la lista de correo de los activistas de Linux, donde el canal a usar se selecciona con el campo de encabezado X-Mn-Key.

2.1.3. AGENTES DE CORREO

Todas las aplicaciones de correo caen en al menos una de tres clasificaciones. Cada clasificación juega un papel específico en el proceso de mover y administrar los mensajes de correo.

2.1.3.1. MTA (Mail Transfer Protocol, Agente de transferencia de correo)

Transfiere mensajes de correo electrónico entre hosts usando SMTP. Un mensaje puede envolver a muchos MTAs a medida que este se mueve hasta llegar a su destino.

Aunque la entrega de mensajes entre máquinas puede parecer bien simple, el proceso completo de decidir si un MTA particular puede o debería aceptar un mensaje para ser repartido, es más bien complicada. Además, debido a los problemas de spam, el uso de un MTA particular está usualmente restringido por la configuración del MTA o por la falta de acceso a la red MTA.

Muchos programas clientes de correo modernos pueden actuar como un MTA cuando estén enviando correo. Sin embargo, esta acción no debería ser confundida con el papel de un MTA verdadero. La única razón de que los programas de correo cliente son capaces de enviar mensajes (como un MTA) es porque el host ejecutando la aplicación no tiene su propio MTA. Esto es particularmente cierto para programas de cliente o para sistemas que no están basados en el sistema operativo Unix. Sin embargo, estos programas clientes sólo envían mensajes hacia afuera hacia un MTA para el cual están autorizados a utilizar y no directamente al servidor de correos recipiente.

2.1.3.2. MDA (Mail Delivery Agent, Agente de entrega de correos)

Invocado por un MTA para archivar correo entrante en el buzón de correo del usuario. En muchos casos, el MDA es en realidad un Agente de entrega local (LDA Local Delivery Agent), tal como mail o Procmail.

Cualquier programa que realmente maneja un mensaje para entrega al punto en que puede ser leído por una aplicación cliente de correos se puede considerar un MDA. Por esta razón, algunos MTAs (tales como Sendmail y Postfix) pueden tener el papel de un MDA cuando ellos anexan nuevos mensajes de correo al archivo spool de correo del usuario. En general, los MDAs no transportan mensajes entre sistemas tampoco proporcionan una interfaz de usuario; los MDAs distribuyen y clasifican mensajes en la máquina local para que lo acceda una aplicación cliente de correo.

2.1.3.3. MUA (Mail User Agent, Agente de usuario de correo)

Un MUA es un programa que permite a los usuarios leer y redactar mensajes de correo. Muchos MUAs son capaces de recuperar mensajes a través de los protocolos POP o IMAP, configurando los buzones de correo para almacenar mensajes y enviando los mensajes salientes a un MTA.

Los MUAs pueden ser de interfaz gráfica, tal como Mozilla Mail, Outlook Express, Eudora o tener una interfaz basada en texto muy sencilla, tal como mutt o pine.

2.1.3.4. MSA (Mail Submission Agent, Agente de Recepción de Correo)

En el caso de sendmail le llaman MSP (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.

2.1.4. PROTOCOLOS DE CORREO

2.1.4.1. SMTP (Simple Mail Transfer Protocol, Protocolo simple de transferencia de correo)

SMTP sirve para enviar correo electrónico. Los mensajes salientes utilizan SMTP para pasar de la máquina del cliente al servidor, lugar desde el que se trasladan hasta el destino final. También dos servidores de correo que intentan

transferir entre sí un mensaje utilizan SMTP para comunicarse, incluso si utilizan plataformas totalmente distintas.

SMTP usa el puerto 25 del servidor para comunicarse. Empieza un intercambio SMTP básico con el sistema conectado mediante la emisión del comando MAIL From: <dirección de correo electrónico> para iniciar el intercambio. El sistema que recibe el comando responde con un mensaje 250 para informar de que se ha recibido el primer comando. A continuación, el sistema conectado comunica las direcciones de correo electrónico para recibir el mensaje del sistema receptor, seguido de un mensaje con el comando DATA. Este mensaje notifica al sistema receptor que la siguiente parte de la comunicación será el cuerpo real del mensaje de correo electrónico. Cuando el sistema conectado finaliza el envío del mensaje de correo electrónico, coloca un punto "." en una línea. A partir de ese momento, se considera que el mensaje se ha enviado.

El protocolo SMTP también permite gestionar el reenvío de mensajes entre sistemas si el sistema receptor sabe el destino al que tiene que enviar el mensaje. El protocolo puede verificar si determinados usuarios utilizan realmente un servidor de correo concreto (VRFY) o ampliar una lista de distribución de correo (EXPN). También se puede retrasar el envío de correo electrónico entre dos servidores SMTP si en los dos sistemas se permite realizar esta actividad.

SMTP no requiere autentificación en su forma más básica. Esto ha provocado mucho correo basura o spam, ya que un usuario externo puede utilizar el sistema de

otro para enviar o transmitir el correo a listas completas de destinatarios con los recursos y ancho de banda del sistema. Las aplicaciones SMTP modernas han progresado enormemente al minimizar este comportamiento y restringir las transmisiones de modo que sólo los hosts conocidos envíen correo electrónico.

En el documento RFC-821 se describe el comportamiento básico del protocolo SMTP, aunque varias extensiones de SMTP, posibles gracias a RFC-1869, han agregado nuevas funciones al SMTP a lo largo de los años con nuevos comandos. Al iniciar una conversación con un servidor SMTP mediante un comando EHLO, en lugar de HELO, el servidor conectado puede identificarse a sí mismo como un servidor compatible con las extensiones SMTP. El servidor receptor contesta con una línea 250 que contiene las distintas extensiones SMPT compatibles. A continuación, el servidor conectado puede utilizar las extensiones compatibles como desee para obtener los objetivos de la comunicación.

La RFC-2554 describe la incorporación de autentificación SMTP mediante el comando AUTH. Otra extensión SMTP muy utilizada se explica en detalle en el documento RFC-2034, que describe el uso entre aplicaciones SMTP de códigos de error estándar separados por puntos. La lectura de los documentos RFC en los que se describen aspectos del protocolo SMTP proporciona los conocimientos básicos sobre la forma de transferir el correo electrónico en Internet. Además, puede conectarse a un servidor SMTP mediante telnet si especifica el puerto 25, como, por ejemplo telnet localhost 25.

2.1.4.2. POP (Post Office Protocol, Protocolo de oficina de correo)

POP permite a los clientes de correo electrónico recuperar los mensajes de los servidores remotos y guardarlos en las máquinas locales. La mayoría de los clientes de correo que utilizan el protocolo POP se configuran automáticamente para eliminar el mensaje del servidor de correo después de transferirlo correctamente al sistema del cliente, aunque esto se puede cambiar.

Para establecer una conexión a un servidor POP, el cliente de correo abre una conexión TCP en el puerto 110 del servidor. Cuando la conexión se ha establecido, el servidor POP envía al cliente POP una invitación y después las dos máquinas se envían entre sí otros comandos y respuestas que se especifican en el protocolo. Como parte de esta comunicación, al cliente POP se le pide que se autentifique en lo que se denomina “Estado de autenticación”, donde el nombre de usuario y la contraseña del usuario se envían al servidor POP. Si la autentificación es correcta, el cliente POP pasa al “Estado de transacción”, fase en la que se pueden utilizar los comandos LIST, RETR y DELE para mostrar, descargar y eliminar mensajes del servidor, respectivamente. Los mensajes definidos para su eliminación no se quitan realmente del servidor hasta que el cliente POP envía el comando QUIT para terminar la sesión. En ese momento, el servidor POP pasa al “Estado de actualización”, fase en la que se eliminan los mensajes marcados y se limpian todos los recursos restantes de la sesión.

POP es un protocolo mucho más sencillo que IMAP, porque no se tienen que enviar tantos comandos entre el cliente y el servidor. POP también es en cierta medida más conocido, aunque la mayoría de los clientes de correo electrónico pueden utilizar cualquiera de estos protocolos.

En varios documentos RFC se proporciona una explicación sobre el protocolo POP, aunque es en el documento RFC-1939 en el que se ofrece una descripción básica del protocolo POP3, la versión actual del protocolo.

También se pueden ejecutar otras versiones del protocolo POP menos utilizadas, como, por ejemplo:

- *APOP* — POP3 con autentificación MDS. En este protocolo, el cliente de correo envía un hash codificado de la contraseña al servidor, en lugar de enviar la contraseña en texto plano.
- *KPOP* — POP3 con autentificación Kerberos.
- *RPOP* — POP3 con autentificación RPOP, que utiliza un identificador de usuario similar a una contraseña para autentificar las peticiones del protocolo POP. No obstante, este identificador está codificado, de modo que RPOP no es más seguro que el protocolo POP normal.

En Linux hay disponibles muchos servidores, clientes y otras aplicaciones POP. Si prefiere utilizar un cliente de correo electrónico gráfico, Mozilla Mail es una

opción perfecta. Además, con otros programas, como fetchmail, se puede recuperar el correo electrónico mediante el protocolo POP.

2.1.4.3. IMAP (Internet Message Access Protocol, Protocolo de acceso a mensajes de Internet)

IMAP es un método que utilizan las aplicaciones cliente de correo electrónico para obtener acceso a los mensajes almacenados remotamente. Al utilizar el protocolo IMAP, normalmente denominado IMAP4 después de la versión del protocolo en cuestión, los mensajes de correo electrónico se conservan en el servidor de correo remoto, donde el usuario puede leerlos o eliminarlos, además de cambiar el nombre o eliminar los buzones de correo para almacenar correo electrónico.

El protocolo IMAP es totalmente compatible con importantes estándares de mensajes de Internet, como, MIME, que permiten recibir ficheros adjuntos. Muchos clientes de correo electrónico que utilizan el protocolo IMAP también se pueden configurar para que se almacene temporalmente en caché una copia de los mensajes localmente, de modo que el usuario puede examinar los mensajes que ha leído anteriormente si no está conectado directamente al servidor IMAP.

IMAP lo utilizan principalmente los usuarios que pueden obtener acceso a su correo desde varias máquinas, como mensajes almacenados en una ubicación central y a los que puede acceder cualquier sistema que utilice un cliente de correo IMAP y

una conexión con el servidor IMAP remoto. También los usuarios que se conectan a Internet o a una red privada a través de una conexión de ancho de banda baja utilizan a menudo el protocolo IMAP, puesto que sólo la información de cabecera del correo se obtiene inicialmente. Esto les permite posponer la descarga de mensajes que tienen ficheros adjuntos de gran tamaño hasta el momento en el que no se utilice la banda de ancho limitada. De la misma manera, el usuario puede eliminar el correo electrónico que no le interesa sin tener que ver antes el cuerpo del mensaje, lo cual evita el tener que descargar un mensaje mediante la conexión de red que utilicen.

Los documentos RFC del protocolo IMAP contienen detalles y especificaciones sobre cómo debe funcionar el protocolo. El documento RFC-1730 define en primer lugar el modo en el que el protocolo IMAP debe utilizar la versión 4, pero RFC-2060 contiene las cuestiones de implantación de IMAP actuales que utilizan muchos servidores IMAP y que se denomina versión IMAP4rev1.

El paquete imap de Linux permite a los usuarios conectarse al sistema y recibir correo electrónico con el protocolo IMAP. Se admiten conexiones IMAP gracias a la integración de la tecnología SSL (Secure Socket Layer) en el demonio imapd, que permite usar el fichero de certificados /usr/share/ssl/certs/imapd.pem. No es necesario utilizar el programa stunnel para el cifrado SSL de las conexiones IMAP, aunque no obstante se puede usar.

También hay disponibles otros clientes y servidores IMAP gratuitos y comerciales, los cuales amplían el protocolo IMAP y proporcionan funciones adicionales.

2.1.4.4. DMSP (Distributed Mail System Protocol, Protocolo de sistema de correo distribuido)

DMSP no supone que todo el correo electrónico está en un solo servidor, como el POP3 y el IMAP. En cambio, permite a los usuarios descargar correo del servidor a una estación de trabajo, PC portátil y luego desconectarse. El correo electrónico puede leerse y contestarse estando desconectado. Al ocurrir una reconexión después, el correo electrónico se transferirá y el sistema se resincronizará.

2.1.5. FORMATO DE CORREOS

El estándar de formato de envió de correo electrónico entre servidores SMTP tiene algunas limitaciones que son principalmente:

- Envío de mensajes en ASCII formado por 7 bits.
- Cada línea del mensaje no puede exceder los 100 caracteres.

Estas condiciones limita a:

- No poder utilizar caracteres extendidos (vocales acentuadas y ¤)
- No poder enviar archivos binarios.

2.1.5.1. UUENCODE & UUDECODE

Para solucionar parte del problema se implementaron estándares. La primera solución consistió en poder enviar archivos binarios en un mensaje. Para eso se desarrollo un sistema por la cual, los archivos binarios podrían adjuntarse dentro de un mensaje en formato ASCII y así podría viajar por la red. Para esto se necesitaría que hubiera algo que convierta el binario a texto por parte del emisor y otra que haga el proceso inverso por parte del receptor. A esto se llamo UUENCODE a la acción de codificar el archivo y UUDECODE para decodificar. Estos archivos se los incluía en la cabecera de cada mensaje y esta característica debe estar implementada en el MUA.

2.1.5.2. MIME (Multipurpose Internet Mail Extensions, Extensiones Multipropósito para Correo Electrónico)

Con lo anterior se soluciono parte del problema, pero aun quedaba por resolver el problema de la visualización de las vocales acentuadas y letra ¤, que no están dentro de la tabla ASCII de 7 bits, también la definición de estándares para la resolución de los saltos de línea, la conversión del TAB como caracteres en blanco.

Para esto se trato de desarrollar un estándar que se denomino MIME, que puede funcionar de varias maneras, una de ellas es la utilización de la transmisión de caracteres de 8 bits entre los servidores, pero para que realmente funcione este modelo, todos los servidores deberían ajustarse a esta modalidad y no es seguro que así lo hagan.

Lo más común es dejar que la transmisión siga siendo en 7 bits y se emplee otra alternativa de utilizar "Quoted Printable" "código imprimible" de esta forma el mensaje es enviado en 7 bits, y los caracteres extendidos se los transforma en una secuencia de 3 caracteres.

Ejemplo , ==> '=E9' en la transmisión.

Pero para que esto funcione sin inconveniente, es necesario elegir un conjunto de caracteres para la transmisión, el más utilizado habitualmente en Europa y América es el ISO8859-1 denominado Latin 1, aunque hay muchos otros.

Si por ejemplo un usuario que tiene activado MIME con quoted printable y eligió como conjunto de caracteres el ISO8859-1 y envía un correo a otro usuario que en su cuerpo tiene letras acentuadas o eñes y lo envía a otro que también tiene configurado su correo con MIME y quoted printable pero eligiendo otra tabla de caracteres, ej ISO 8859-7. Las letras enviadas en un formato, el las va a recibir de otra manera, ya que su MUA va ha interpretar otra cosa.

2.1.6. ALIAS

Los alias de correo son una poderosa opción que permite que el correo sea dirigido a otros apartados postales que son nombres alternativos de usuarios o procesos en un servidor destinatario. Por ejemplo, es una práctica común tener retroalimentación o comentarios con respecto a un servidor Web y que estén dirigidos a “webmaster”. Con frecuencia no hay un usuario llamado “webmaster.” en el servidor, en vez de ello, hay un alias a otro usuario del sistema. Otro uso común para los alias de correo es utilizarlos por los programas de gestión de listas de correo en los cuales un alias dirige todos los mensajes que ingresan al programa de gestión de la lista para que sea interpretado.

El fichero /etc./aliases es el lugar en donde los alias se almacenan. Los programas de correo consultan este fichero cuando están determinando cómo manejar un mensaje que ingresa. Si encuentra una línea en este fichero que coincide con el usuario a quien va dirigido el mensaje, lo redirigen al lugar que indica dicha línea.

Hay tres cosas que los alias permiten:

- Otorgan un nombre corto o bien conocido para el correo que será dirigido hacia una o más personas.
- Pueden invocar a un programa con el mensaje de correo como entrada hacia dicho programa.

- Pueden mandar el correo a un fichero.

Todos los sistemas requieren de alias para el Postmaster y el MAILER-DAEMON para cumplir con el RFC.

2.2. DNS (Domain Name System, Sistema de Nombres de Dominio)

Sistema para asignar nombres a equipos y servicios de red que se organiza en una jerarquía de dominios. La asignación de nombres DNS se utiliza en las redes TCP/IP, como Internet, para localizar equipos y servicios con nombres sencillos. Cuando un usuario escriba un nombre DNS en una aplicación, los servicios DNS podrán traducir el nombre a otra información asociada con el mismo, como una dirección IP.

Las direcciones IP identifican unívocamente a las máquinas de una red permitiendo la comunicación entre estas a través de diversos protocolos con el objetivo de acceder o brindar múltiples servicios. La implementación más difundida actualmente del protocolo IP identifica a los hosts a través de un número binario de 32 bits que en decimal se expresa utilizando cuatro números entre 0 y 255.

La primera forma que se utilizó para convertir nombres a números IP y viceversa fue a través de un fichero nombrado hosts.txt el cual debería estar

distribuido en todas las máquinas que necesitaban el servicio. Este contenía una tabla que expresaba la correspondencia entre un número IP y uno o varios nombres.

En la medida que las redes fueron creciendo e interconectándose entre sí esta forma generaba demasiados inconvenientes pues era muy difícil de mantener, además de que las continuas actualizaciones generaban mucho tráfico, en su mayor parte innecesario, y se dificultaba el mantener la consistencia.

Actualmente el fichero hosts.txt se conoce como /etc./hosts en los ambientes Unix y se continúa utilizando como una forma básica de resolución de nombres.

Posteriormente a esta forma primaria de resolver los nombres en la red, se creó y perfeccionó lo que se conoce actualmente como DNS, que en esencia es una base de datos distribuida, gracias a lo cual permite la administración y control local de los fragmentos en que se divide. Funciona a través del esquema cliente-servidor y está diseñado de forma eficiente para lograr un buen rendimiento, además de permitir la replicación y el cache.

Un servidor de nombres es la máquina que ejecuta el programa que implementa la parte servidora del esquema. Este se encarga de almacenar la información asociada al segmento de la base de datos que controla, y de mantenerla accesible a los clientes, que se conocen como resolver. Un resolver es una subrutina que genera consultas y las envía a través de la red hacia el servidor de nombres correspondiente.

2.2.1. Tipos de servidores de nombres

Existen cuatro tipos de configuración de servidores de nombres primarios:

- **Maestro**. Almacena los registros de las zonas originales y de autoridad para un cierto espacio de nombres, contestando preguntas de otros servidores de nombres buscando respuestas concernientes a ese espacio de nombres.

- **Esclavo.** Responde a las peticiones que provienen de otros servidores de nombres y que se refieren a los espacios de nombres sobre los que tiene autoridad. Sin embargo, los servidores esclavos obtienen la información de sus espacios de nombres desde los servidores maestros.

- **Sólo caché.** Ofrece servicios de resolución de nombres a direcciones IP pero no tiene ninguna autoridad sobre ninguna zona. Las respuestas en general se introducen en un caché por un período de tiempo fijo, la cual es especificada por el registro de zona recuperado.

- **Reenvío.** Reenvía las peticiones a una lista específica de servidores de nombres para la resolución de nombres. Si ninguno de los servidores de nombres especificados puede resolver los nombres, la resolución falla.

Un servidor de nombres puede ser uno o más de estos tipos. Por ejemplo, un servidor de nombres puede ser un maestro para algunas zonas, un esclavo para otras y sólo ofrecer el reenvío de resoluciones para otras.

2.2.2. Espacio de los nombres de dominio

La estructura de la base de datos del DNS posee una forma jerárquica similar al sistema de ficheros de Linux. Esta es una especie de árbol invertido donde cada nodo representa un segmento o dominio. Los nodos a su vez pueden poseer varios subnodos hijos que constituyen subdominios en el DNS. Los nodos que no poseen hijos pueden verse como los nombres de los hosts que pertenecen al dominio definido por el nodo padre. Cada nodo se identifica utilizando una etiqueta cuyo tamaño no debe exceder los 63 caracteres. El nodo raíz tiene una etiqueta vacía. Para indicar el nombre completamente cualificado de un host en el DNS se utilizan todas las etiquetas de los nodos desde este hacia la raíz según la jerarquía. Las etiquetas se separan utilizando el carácter "." y se ordenan de abajo hacia arriba a diferencia de los caminos absolutos en el sistema de ficheros de Linux.

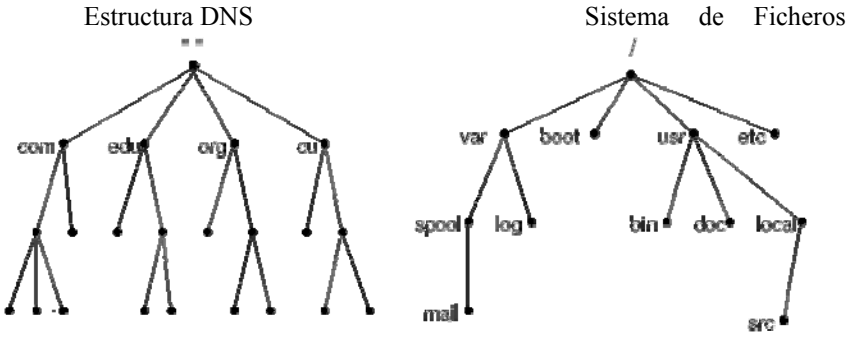

Figura 2.2. Semejanzas entre la estructura DNS y sistema de ficheros Linux

En el DNS cada dominio es administrado por una organización o empresa determinada. Esta puede decidir dividir el o los dominios que administra en subdominios, así como asignar la administración de estos a otras entidades. Cada dominio puede contener tanto subdominios como hosts independientes, al igual que un directorio posee subdirectorios y ficheros a la vez.

El DNS en la actualidad sigue ciertos patrones en cuanto a su organización. Esta se basa en niveles de acuerdo a la posición del dominio. El nivel superior o primer nivel lo forman aquellos dominios descendientes del dominio raíz. Los fundamentales son:

- **com.** Organizaciones comerciales.
- ***edu.*** Organizaciones de propósitos educacionales.
- **net**. Organizaciones dedicadas al desarrollo de las redes.
- **org.** Organizaciones no comerciales.
- ***gov.*** Organizaciones gubernamentales.

Los registros MX (Mail eXchanger) del DNS son los que se utilizan para identifica cuales son las estafetas de correo, para un dominio determinado, y cual es su nivel de preferencia. La preferencia se indica con un valor entero entre 0 y 32767 siendo el valor de mayor preferencia el 0 y el de menor el 32767. La preferencia indica cual es la estafeta a utilizar de forma preferente. En caso de que la estafeta de mayor preferencia este fuera de servicio se acudiría a la siguiente estafeta en orden de preferencia.

	Tipo	**Nombre**	**Función**
Zona	SOA	Start Of Authority	Define una zona representativa del DNS
NS	Name Server	Identifica los servidores de zona, delega subdominios	
Básicos	A	Dirección IPv4	Traducción de nombre a dirección
AAAA	Dirección IPv6 original	Actualmente obsoleto	
A6	Dirección IPv6	Traducción de nombre a dirección IPv6	
PTR	Puntero	Traducción de dirección a nombre	
DNAME	Redirección	Redirección para las traducciones inversas IPv6	
MX	Mail eXchanger	Controla el enrutado del correo	
Seguridad	KEY	Clave pública	Clave pública para un nombre de DNS
NXT	Next	Se usa junto a DNSSEC para las respuestas negativas	
SIG	Signature	Zona autenticada/firmada	
Opcionales	CNAME	Canonical Name	Nicks o alias para un dominio

LOC	Localización	Localización geográfica y extensión	
RP	Persona responsable	Especifica la persona de contacto de cada host	
SRV	Servicios	Proporciona la localización de servicios conocidos	
TXT	Texto	Comentarios o información sin cifrar	

Tabla 2.1. Tipos de registros

2.3. SERVIDORES DE CORREO

2.3.1. SENDMAIL

Sendmail es un programa que proporciona el servicio de correo electrónico en sistemas Linux y Unix. Entre sus objetivos de diseño destaca un gran poder de configuración, capaz de procesar mensajes de e-mail en prácticamente cualquier tipo de red. Sin embargo, esta cualidad ha resultado en una complejidad abrumadora para los no expertos, lo que ha motivado a que su manejo resulte en un tema oscuro y algo místico.

Sendmail es el agente de transporte de correo más común de Internet (en los sistemas Linux y Unix). Aunque actúa principalmente como MTA, también puede ser utilizado como MUA Las misiones básicas de sendmail son:

- Recogida de mails provenientes de un Mail MUA como pueden ser elm, Eudora o pine; o provenientes de un MTA como puede ser el propio sendmail.
- Elección de la estrategia de reparto de los mails, basándose en la información de la dirección del destinatario contenida en la cabecera:

    - Si el mail es local en el sistema, enviará el mail al programa de reparto local de mails.
    - Si el mail no es local, sendmail utilizará el DNS del sistema para determinar el host al que debe ser enviado el mail. Para transferir el mensaje, iniciará una sesión SMTP con el MTA de dicho host.
    - Si no es posible mandar el mail a su destino (porque la maquina receptora esta desconectada, o va muy lenta), sendmail almacenará los mails en una cola de correo, y volverá a intentar el envío del mail un tiempo después. Si el mail no puede ser enviado tras un tiempo razonable, el mail será devuelto a su autor con un mensaje de error. Sendmail debe garantizar que cada mensaje llegue correctamente a su destino, o si hay error este debe ser notificado (ningún mail debe perderse completamente).

- Reformatear el mail antes de pasarlo a la siguiente máquina, según las reglas de reescritura. Según el tipo de conexión que posea con una determinada máquina,

o según el agente de transporte al que vaya dirigido el mail, necesita cambiar los formatos de las direcciones del remitente y del destinatario, algunas líneas de la cabecera del mail, o incluso puede que necesitemos añadir alguna línea a la cabecera. Sendmail debe realizar todas estas tareas para conseguir la máxima compatibilidad entre usuarios distintos.

- Otra función muy importante de sendmail es permitir el uso de alias entre los usuarios del sistema; lo que permitirá crear y mantener listas de correo entre grupos.
- Ejecución como MUA. Aunque no posee interfaz de usuario, sendmail también permite el envío directo de mails a través de su ejecutable.

2.3.1.1. PAQUETES DE SENDMAIL

La gran mayoría de usuarios no tiene necesidad de descargar Sendmail puesto que suele distribuirse en prácticamente todos los sistemas Linux y Unix. Sin embargo, las actualizaciones importantes por cuestiones de seguridad deben obtenerse regularmente del vendedor o distribuidor, o del site de Sendmail

Sendmail se distribuye en tres paquetes, a saber:

- sendmail-cf
- sendmail
- sendmail-doc

Los directorios donde se guardan los archivos de Sendmail varían de sistema en sistema. Además el paquete de preprocesamiento **"m4"** que se distribuye en prácticamente todos los sistemas Linux/Unix es requerido.

2.3.1.2. Preprocesador de macros m4

m4 es un preprocesador de macros que produce un archivo de configuración de Sendmail procesando un archivo cuyo nombre termina con .mc (macro configuration). Esto es, procesa su entrada y reúne definiciones de macros, luego las reemplaza con sus valores y saca el resultado.

Con m4, las macros se definen como:

define (macro, valor)

Aquí, la macro es un nombre simbólico que se usará después. Los nombres aceptables deben empezar con una subraya o con una letra y pueden contener letras, dígitos y subrayas. El valor puede ser cualquier cadena de texto. Ambos van separados por una coma y esta a su vez, puede estar entre dos espacios en blanco.

No debe haber espacio entre el define y el paréntesis izquierdo. La definición termina con el paréntesis derecho.

Se puede usar dnl para remover líneas en blanco de un archivo de configuración.

Cuando un nombre de macro m4 está inmediatamente seguido por un paréntesis derecho, se trata como llamada a una función. Los argumentos que le son dados en ese sentido, se usan para remplazar las expresiones $digit en la definición original.

m4 tiene la capacidad de dividir su entrada en diferentes partes y para reensamblarlas más tarde en un orden más lógico. Considere, por ejemplo, el deseo de sacar todas las opciones juntas. Una forma de hacer esto es a través de los comandos divert y undivert de m4

Divert	**Descripción**
(-1)	Interno para m4, le dice que ignore todas las líneas que siguen.
(0)	Interno para m4, indica que se paren las desviaciones y que se mande a la salida de inmediato.
(1)	Detección del host local y resolución con LOCAL_NET_CONFIG.

Tabla 2.2. Parámetros divert

2.3.1.3. Feature

Para incluir alguna característica (feature), debe incluirse un comando de m4 como el que sigue en el archivo mc:

FEATURE(clave)

FEATURE(clave,argumento)

Estas declaraciones provocan que un archivo llamado _CF_DIR/feature/clave.m4 sea leído en ese lugar en el archivo.mc.

La tabla siguiente muestra algunos de los archivos clave. Algunas claves requieren un argumento adicional.

Clave	**Descripción**
allmasquarade	Enmascara también al remitente
always_add_domain	Agregar el dominio local aún en un correo local.
genericstable	Transforma las direcciones del remitente.
Local_procmail	Usa procmail(1) como el agente de distribución local.
Mailertable	La base de datos selecciona nuevos agentes de distribución.
Masquerade_entire_domain	Enmascara todos los hosts bajo un dominio.
Nodns	Omite el soporte DNS del archivo de configuración.
Nouucp	Elimina todo el soporte UUCP.
Nullclient	Difunde todo el correo a través de un host de correo.
Redirect	Agrega el soporte para address. REDIRECT.

Smrsh	Usa smrsh (shell restringido de sendmail)
use_ct_file	Busca en el archivo /etc./mail/trusted-users una lista de usuarios confiables. Es decir, usuarios que pueden cambiar la parte from de su dirección sin que les genere una advertencia.
use_cw_file	Lee los nombres alternativos para el host local del archivo /etc./mail/local-host-names.
Uucpdomain	Convierte hosts uucp a través de una base de datos.
Virtusertable	Soporte para dominios virtuales.
Promiscuous_relay	Por omisión, los archivos de configuración de sendmail no permiten la difusión de correo (relay). Esta opción configura el sitio para permitir la difusión de correo desde cualquier sitio hacia cualquier sitio.
Relay_based_on_MX	Habilita la capacidad de permitir difusión (relaying) basada en los registros MX de la porción de host de un destinatario entrante.
Relay_local_from	Permite la difusión si la porción de dominio del remitente de correo es un host local.
Relay_entire_domain	Permite la difusión (relay) a cualquier hosts del dominio.
relay_hosts_only	Por omisión los nombres listados como RELAY en

	el archivo access son nombres de dominio, no nombres de hosts. Este parámetro permite buscar nombres de hosts individuales.
accept_unqualified_senders	Normalmente, los comandos MAIL FROM: en la sesión SMTP serán rechazados si la conexión es una conexión de red y la dirección del remitente no incluye un nombre de domino.
accept_unresovable_domains	Normalmente, los comandos MAIL FROM: en la sesión de SMTP serán rechazados si la parte del host del argumento de MAIL FROM: no puede localizarse en el servicio de nombres de hosts, por ej. DNS.
access_db	Habilita la característica de base de datos de accesos. Por omisión, la especificación de base de datos de acceso es: hash -o /etc/mail/access
blacklist_recipients	Habilita la capacidad de bloquear correo entrante para ciertos destinatarios, nombres de hosts y direcciones.
Rbl	Habilita el rechazo de hosts que se encuentren en la lista de rbl.

Tabla 2.3. Argumentos feature.

2.3.2. POSTFIX

Postfix es un MTA, escrito originalmente por Wietse Venema, que comenzó siendo una alternativa a Sendmail. Sendmail controla cerca del 70% del movimiento de correo electrónico en Internet. El problema que Sendmail es demasiado complicado para configurar. Peor aun si se quiere hacer cosas mas allá de una configuración simple.

Postfix es un MTA relativamente fácil de administrar, seguro y que no sobrecarga mucho la máquina ya que solamente se cargan los módulos necesarios en cada momento. Su función es comunicarse con los otros servidores, para entregarse entre ellos el correo.

2.3.2.1. Arquitectura

A diferencia de Sendmail, que es un gestor de correo monolítico, en el diseño de Postfix se han disgregado los diversos tratamientos que se realizan sobre un mensaje a su paso por un MTA, adjudicando cada tratamiento o grupo de tratamientos a un proceso independiente. El conjunto de todos estos procesos es Postfix.

Los procesos que conforman Postfix se comunican a través de sockets que se crean, por razones de seguridad, en un directorio de acceso restringido. La información que intercambian los diversos procesos es la mínima posible,

limitándose en la mayoría de los casos a la referencia de la entrada en una cola y la relación de destinatarios, o a un simple identificador de estado.

La siguiente figura proporciona una visión global de los elementos que componen Postfix:

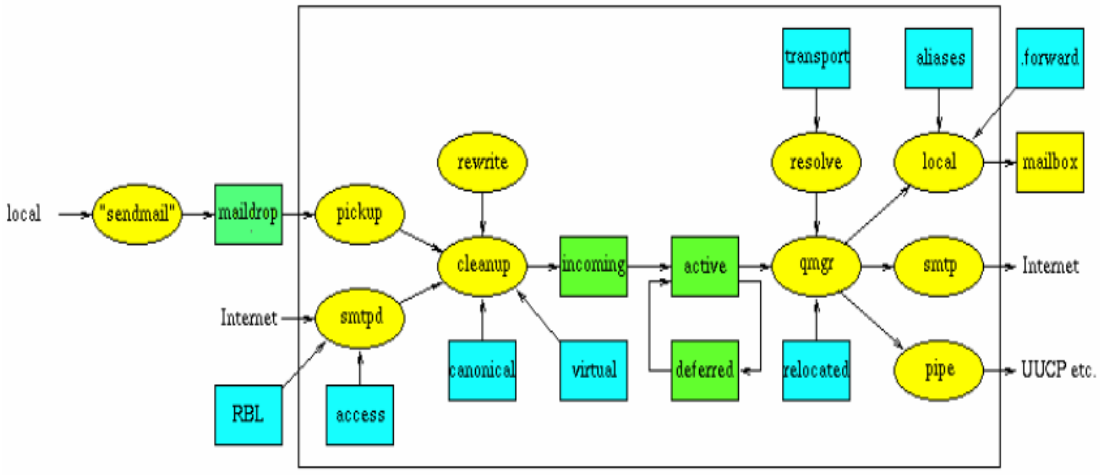

Figura 2.3. Elementos que componen Postfix

Postfix basa su funcionamiento en cuatro colas: maildrop, incoming, active y deferred (cuadrados coloreados en verde).

El correo que se genera de forma local se deposita en maildrop para su posterior proceso. El proceso pickup toma los mensajes que llegan a maildrop y los pasa a cleanup, que analiza las cabeceras de los mensajes y deposita éstos en la cola incoming.

En la cola active se encuentran aquellos mensajes que están en fase de encaminamiento, y en deferred los mensajes que por diversas causas no se pueden encaminar o están pendientes de reintentar su encaminamiento.

El proceso qmgr es el encargado de tratar los mensajes que llegan a la cola incoming, depositarlos en active y lanzar el proceso adecuado para su encaminamiento, como pueden ser local, smtp o pipe.

El correo procedente de otros sistemas se atiende a través del proceso smtpd, utilizando el protocolo SMTP, pudiendo utilizar accesos a servidores de RBL o tablas internas para aplicar las políticas de acceso a cada mensaje entrante.

Coloreadas de azul aparecen las tablas que, creadas por el administrador, sirven a los diferentes procesos para concretar el tratamiento que debe darse a cada mensaje. Se usan seis tablas: access, aliases, canonical, relocated, transport y virtual. Aunque no es obligatoria la existencia ni utilización de todas ellas.

La tabla access permite definir una relación explícita de sistemas a los que se les deben aceptar o rechazar sus mensajes. La utiliza el proceso smptd.

La tabla aliases, al igual que en Sendmail, define una serie de nombres alternativos a usuarios locales, y la consulta el proceso local.

El proceso cleanup, mediante la tabla canonical establece relaciones entre nombres alternativos y nombres reales, ya sean usuarios locales o no.

El proceso qmgr utiliza la tabla relocated para devolver los mensajes de usuarios que han cambiado de dirección: “User has moved to new-email”.

Con la tabla transport, que es utilizada por el proceso trivial-rewrite, se define la política de encaminamiento por dominios, subdominios e incluso por dirección concreta de usuario.

Para la gestión y soporte de dominios virtuales el proceso cleanup utiliza la tabla virtual. En ella se establecen las relaciones entre usuarios virtuales y reales, e incluso de dominios completos.

Todas estas tablas pueden usar alguno de los siguientes tipos de formato de base de datos:

- Fichero binario indexado (btree, hash, dbm, etc.).
- Fichero de texto basado en expresiones regulares (regexp).
- Sistema externo de base de datos (NIS, LDAP, MySQL, etc.).

2.3.2.2. Requerimientos

- postfix. Este es el paquete principal de Postfix.

- postfix-dev. Entorno de desarrollo.
- postfix-doc. Documentación.
- postfix-ldap. Soporte LDAP.
- postfix-mysql. Soporte MySQL.
- postfix-pcre. Soporte de expresiones regulares.
- postfix-snap-*. Versiones snapshot. Pueden ser inestables.
- postfix-tls. Soporte TLS y SASL (SMTP autentificado).

2.3.3. QMAIL

Qmail es un manejador de correos extremadamente robusto que corre en cualquier sistema operativo que emule a UNIX. Esto incluye a Linux y a todos los BSD's. Se trata de un sustituto completo para el sistema sendmail que se suministra con los sistemas operativos UNIX. Qmail utiliza el SMTP para intercambiar mensajes con los MTA de otros sistemas.

2.3.3.1. Arquitectura de qmail

Qmail está compuesto de diversos subprogramas que realizan tareas específicas y que en conjunto constituyen el sistema de correo electrónico. La figura proporciona una visión esquemática de estos componentes.

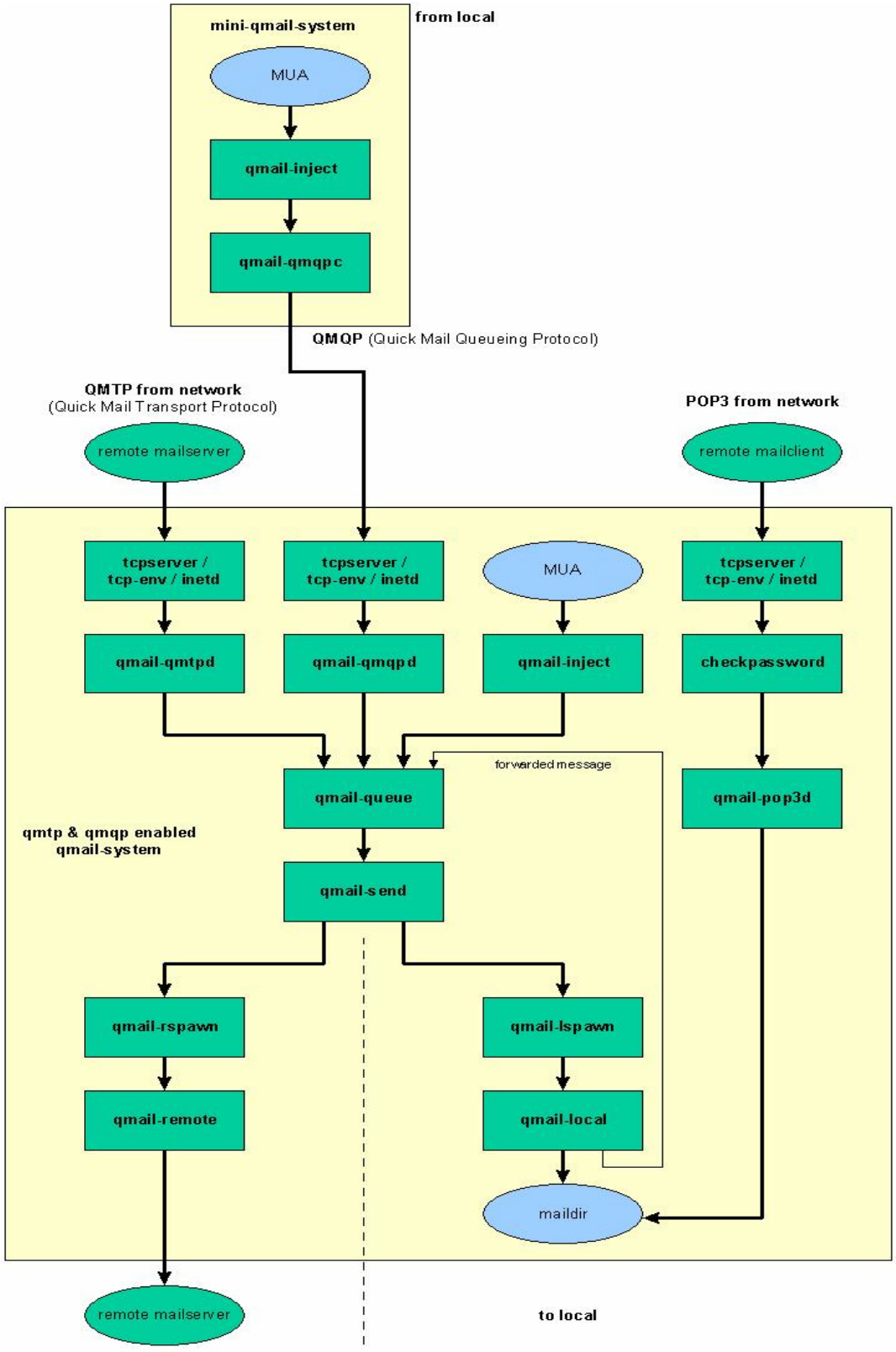

Figura 2.4. Arquitectura de Qmail

- **Origen local:** Un usuario Linux operando en el sistema remite un mensaje usando un MUA como mail. Esto normalmente origina una llamada a qmail-inyect el cual llevará el mensaje al programa de encolamiento qmailqueue. Este lo almacenará en la cola de mensajes /var/qmail/queue. Luego, qmail-send intentará remitirlo a su destino (si se puede) mediante los programas qmail-lspawn o qmail-rspawn.
- **Origen remoto exterior**: Un usuario de Internet ha enviado un mensaje hacia la red. Este mensaje debe provenir de otro servidor de e-mail mediante SMTP. qmail-smtpd descubre que el destinatario es local, y acepta el mensaje, el cual pasa a la fase de encolamiento.
- **Origen remoto de la LAN**: Un cliente de la red desea enviar un mensaje. Para esto ha configurado el MUA a fin de remitir hacia el servidor qmail. Esta remisión normalmente la efectúa usando SMTP, y por tanto qmail-smtpd es el encargado de la recepción.
- **Destino local**: Los mensajes con destino local son guardados en el mailbox por procmail (o qmail-local) para ser recogidos por los MUAs de los usuarios que trabajan en el servidor. En el caso de que el usuario no esté en el servidor, sino, en una estación de trabajo, entonces el MUA deberá conectarse a un servidor IMAP o POP para obtener los mensajes del mailbox. En cualquier caso, qmail-lspawn es el encargado de controlar el agente de delivery local.
- **Destino remoto:** Los mensajes remotos se remiten con SMTP hacia otros MTA. qmail-rspawn es el encargado de los mensajes que se envían remotamente.
- **tcpserver:** Es el encargado de iniciar qmail-smtpd de un modo seguro y relativamente inteligente.

2.3.3.2. Smtp-auth

Parche para qmail, que activa el soporte para el protocolo de autentificación SMTP con la búsqueda de diferentes tipos de autentificación como: login, plain y cram-md5. Este parche es útil ya que previene de la posibilidad que el servidor sirva para hacer spam.

2.3.3.3. Daemontools

Colección de utilidades para el manejo de servicios UNIX:

- **Supervise.** Supervisa un servicio. Este arranca el servicio y reinicia el servicio si este muere. Arrancar un nuevo servicio es fácil: Todo supervise necesita un directorio con un script run que inicie el servicio.
- **Multilog**. Guarda los mensajes de error en uno o más logs. Él opcionalmente añade en cada línea la fecha y hora, y para cada registro, incluye o excluye las líneas que emparejan patrones especificados. Rota automáticamente registros para limitar la cantidad de espacio de disco usada. Si el disco está lleno, se detiene brevemente e intenta otra vez, sin perder ningún dato.

2.3.3.4. Vpopmail

La manipulación de dominios virtuales es una edición común planteada por los nuevos usuarios en las comunidades de qmail y del postfix. Inter7 ha desarrollado el

vpopmail (vchkpw), un paquete de software libre del GLP, para proporcionar una manera fácil de manejar dominios virtuales del e-mail y cuentas del e-mail y no /etc./passwd en qmail o postfix. Además es muy útil ya que se puede tener varios dominios en una sola dirección IP.

2.3.4. EXIM

Exim es un MTA para sistemas Unix o basados en Unix. Está diseñado para organizaciones que tienen una conexión permanente a Internet. Sin embargo, puede usarse en organizaciones que no tengan una conexión permanente a Internet con ajustes convenientes.

Al igual que Sendmail, es un software monolítico, lo que lo hace no tan seguro y rápido como los dos anteriores. Por contra tiene un mayor número de características que aquellos. Es más sencillo de configurar que Sendmail y bastante compatible con él.

2.3.4.1. Estructura general del fichero de configuración

El fichero de configuración se divide en 6 bloques. Cada bloque está separado del siguiente por la palabra “end”, excepto el último que no lo lleva. Todos los bloques deben aparecer, si alguno se encuentra vacío tiene que aparecer el “end” de todos modos. Los bloques son los siguientes:

- **Configuración principal**: aquí van las directivas principales de configuración, las preferencias, etc., como el nombre de la máquina, a quién se hace relay, etc.
- **Transports**: Cuando se sabe definitivamente cómo y a donde se va a enviar un determinado mensaje, el transport correspondiente es el que se encarga de hacerlo. Cada transport tiene un driver que indica el tipo de reparto. Ejemplos de drivers: “appendfile”", que concatena el msg a un fichero smtp que hace una conexión a un smtp para enviar.
- **Directors**: Cuando un mensaje va a una dirección local, se busca un director que sepa qué hacer con ella. Son los que se encargan de buscar en el fichero de aliases, etc. El orden ES importante.
- **Routers**: Cuando una dirección no es local, se busca el primer router que sea capaz de enviarla. El orden ES importante.
- **Retry**: Aquí se especifica el tiempo que tiene que transcurrir hasta que se considere que un msg no se puede enviar.
- **Reescritura**: Aquí están las reglas de reescritura de cabeceras.

Cuando exim recibe un e-mail, lo primero que hace es aplicar las reglas de reescritura de cabeceras. Una vez reescrito, se comprueba si el destinatario es local o está en otra máquina. Si es local, se pasa por la lista de directors, hasta que alguno sepa qué hacer con él y lo reparta.

Si no es local, se pasa por la lista de routers, también hasta que alguno sepa qué hacer con él. Para saber si un e-mail es local, se compara el dominio del destinatario con la lista definida en el campo local_domains en el fichero de configuración.

2.4. FETCHMAIL

Programa que puede recuperar correo electrónico de servidores remotos para conexiones TCP/IP bajo demanda. Muchos usuarios aprecian la capacidad de separar el proceso de descarga de mensajes ubicados en un servidor remoto del proceso de lectura y organización de correo en un MUA. Fetchmail se conecta y descarga rápidamente todos los mensajes al fichero spool de correo mediante el uso de diversos protocolos, entre los que se incluyen POP3 e IMAP. Incluso permite reenviar los mensajes de correo a un servidor SMTP si es necesario.

La configuración de fetchmail se realiza en el fichero .fetchmailrc, ubicado en el directorio principal del usuario. Mediante el uso de preferencias en el fichero .fetchmailrc, Fetchmail comprobará si hay correo en un servidor remoto e intentará entregarlo al puerto 25 de la máquina local utilizando el agente MTA local para dirigir el correo al fichero de spool del usuario correcto.

2.4.1. Opciones de configuración

Aunque se pueden insertar todas las opciones en la línea de comandos pertinente para comprobar si hay correo en un servidor remoto al ejecutar Fetchmail, el uso de .fetchmailrc proporciona un método más sencillo. Todas las opciones de configuración se guardan en el fichero .fetchmailrc, pero se pueden sobrescribir cuando se ejecuta Fetchmail si se especifica esta opción en la línea de comandos.

Un fichero de usuario .fetchmailrc se divide en tres tipos concretos de opciones de configuración:

- **Opciones globales.** Indican a fetchmail las instrucciones que controlan el funcionamiento del programa o proporcionan los valores para cada conexión en la que se comprueba si hay correo electrónico.
- **Opciones de servidor.** Especifican información necesaria sobre el servidor, como nombre de host, así como las preferencias que desearía ver aplicadas con un servidor de correo concreto. Estas opciones afectan a cada opción de usuario utilizada con ese servidor.
- **Opciones de usuario.** Contienen información, como nombre de usuario y contraseña, que es necesaria para autenticar y comprobar si hay correo utilizando un servidor de correo concreto.

```
set postmaster "user1"
set bouncemail

poll pop.dominio.com proto pop3
    user 'user1' there with password 'secreto' is user1 here

poll mail.domidio2.com
    user 'user5' there with password 'secreto2' is user1 here
    user 'user7' there with password 'secreto3' is user1 here
```

Tabla 2.4. Ejemplo de fichero básico .fetchmailrc

En este ejemplo, las opciones globales son las que establecen que se le envíe correo al usuario como última instancia (opción postmaster) y que todos los errores de correo se manden al postmaster en lugar de a la persona que ha enviado el correo

bouncemail). La acción set indica a fetchmail que esta línea contiene una opción global. A continuación, se especifican dos servidores de correo: uno para que compruebe si hay correo con el protocolo POP3 y otro para que pruebe a usar varios protocolos para encontrar uno que funcione. Se comprueba el correo de dos usuarios con la segunda opción de servidor, pero todo el correo que se encuentre se envía al spool de correo del user1. Esto permite comprobar varios buzones de diversos servidores como si se tratara de un único buzón MUA. La información específica de cada usuario comienza con la acción user.

2.4.2. Opciones de comando de Fetchmail

La mayoría de las opciones de fetchmail se pueden utilizar en la línea de comando al ejecutar el comando fetchmail o reflejar las opciones de configuración de .fetchmailrc. Esto se realiza para que se use fetchmail con o sin un fichero de configuración. En ocasiones puede estar interesado en ejecutar el comando fetchmail con otras opciones para un fin concreto. Puesto que cualquier opción especificada en la línea de comando sobrescribe las opciones del fichero de configuración, puede omitir opciones de comando temporales que sobrescriban un parámetro de .fetchmailrc que causa un error.

2.4.3. Opciones informativas o de depuración

Determinadas opciones que se utilizan después del comando fetchmail pueden proporcionar información importante.

- **–configdump**. Muestra cada opción posible en función de la información de .fetchmailrc y los valores por defecto de fetchmail. No se recupera correo de ningún usuario al usar esta opción.
- **–s**. Ejecuta fetchmail en modo silencioso, con lo cual se evita que aparezcan mensajes y errores después del comando fetchmail.
- **–v**. Ejecuta fetchmail en modo detallado y muestra todas las comunicaciones entre fetchmail y los servidores de correo remotos.
- **–V**. Hace que fetchmail muestre información de versión detallada, una lista de las opciones globales y los parámetros que se utilizarán con cada usuario, incluido el protocolo de correo y el método de autenticación. No se recupera correo de ningún usuario al usar esta opción.

2.4.4. Opciones especiales

Estas opciones son en ocasiones útiles para sobrescribir los valores por defecto que a menudo contiene el fichero .fetchmailrc.

- **–a.** Indica a fetchmail descargar todos los mensajes del servidor de correo remoto, ya se hayan o no visto antes. Por defecto, fetchmail sólo descarga los mensajes nuevos.
- **–k.** Hace que fetchmail deje una copia de los mensajes en el servidor de correo remoto después de descargarlos. Esta opción sobrescribe el comportamiento por defecto de eliminar los mensajes después de descargarlos.
- **-l <*número máximo de bytes*>.** Indica a fetchmail que no descargue mensajes con un tamaño superior al indicado y dejarlos en el servidor de correo remoto.

- **–quit.** Sale del proceso de demonio de fetchmail.

2.5. PROCMAIL

Sencillo programa que permite procesar correos, haciendo con ellos lo que se desee, de una forma sencilla pero muy potente. Se puede separar los correos según determinados filtros, eliminar correos spam, reenviar a otras cuentas, activar antivirus de correos, responder automáticamente, ejecutar programas, etc.

Las posibilidades son prácticamente ilimitadas y dependen de la imaginación y habilidad, pero básicamente el proceso consiste en 2 pasos:

- Identificar el correo
- Procesar el correo

Para identificar el correo, se usa las expresiones regulares, de forma que los todos correos que cumplan unas determinadas condiciones, pasaran a realizar el proceso requerido.

2.5.1. Configuración de Procmail

Los ficheros de configuración de Procmail, y más en concreto el fichero de usuario .procmailrc, contienen variables de entorno importantes. Estas variables

indican a Procmail qué mensajes deben ordenarse, qué hacer con los mensajes que no coinciden con ninguna regla, etc.

Estas variables de entorno normalmente aparecen al principio del fichero .procmailrc con el siguiente formato en cada línea

<env-variable>="*<valor>*"

Muchas variables de entorno no son utilizadas por la mayor parte de los usuarios de procmail, y muchas de las variables de entorno más importantes ya están definidas con un valor por defecto. La mayoría del tiempo tratará con las siguientes variables:

- **DEFAULT.** Establece el buzón por defecto en el que se ubicarán los mensajes que no coincidan con las reglas. El valor por defecto DEFAULT es el mismo que $ORGMAIL.
- **INCLUDERC.** Especifica ficheros rc adicionales que contienen más reglas para los que deben comprobarse los mensajes. Esto permite desglosar las listas de reglas de Procmail en ficheros individuales según diversas funciones (como bloquear correo basura y gestionar listas de correo) que se pueden activar o desactivar con caracteres de comentario en el fichero de usuario .procmailrc.
- **LOCKSLEEP.** Establece cada cuanto tiempo, en segundos, procmail intentará usar un lockfile concreto. El valor por defecto es ocho segundos.
- **LOCKTIMEOUT.** Establece la cantidad de tiempo, en segundos, que debe transcurrir después de modificar un lockfile para que procmail asuma que este

lockfile es antiguo para que se pueda eliminar. El valor por defecto es 1024 segundos.

- **LOGFILE.** Ubicación y fichero que contienen los mensajes de error o de información de procmail.
- **MAILDIR.** Establece el directorio de trabajo actual de procmail. Si se define este directorio, todas las rutas de procmail estarán relacionadas con este directorio.
- **ORGMAIL.** Especifica el buzón original u otro lugar para colocar los mensajes si no se pueden ubicar en la ubicación de regla o por defecto.
- **SUSPEND.** Establece la cantidad de tiempo, en segundos, que procmail se detendrá si está disponible un recurso necesario, como espacio de intercambio.
- **SWITCHRC.** Permite a un usuario especificar un fichero externo que contiene reglas de procmail adicionales, como la opción INCLUDERC, excepto si la regla en examen se ha detenido en el fichero de configuración de referencia y sólo se usan las reglas del fichero SWITCHRC especificado.
- **VERBOSE.** Hace que procmail registre mucha más información. Esta opción es útil para procesos de depuración.

2.6. SERVIDOR DE NOTICIAS

Servicio de Internet que permite el intercambio de mensajes en un foro común sobre un determinado tema de interés para todos sus lectores. En cuanto a formato, son algo similar a los e-mails: simples mensajes de texto formados por la cabecera

del mensaje seguida del cuerpo, y donde a su vez la cabecera se divide en diferentes campos que indican el remitente, el grupo destinatario o el tema del mensaje.

Si se busca información de un determinado tema (programación, electrónica, diseño, imágenes, etc.), sólo es necesario apuntar el programa de news hacia el grupo adecuado para disponer de la posibilidad de intercambio de información e intereses comunes entre un gran grupo de usuarios interesados en el mismo tema. Si se envía un mensaje al grupo, todas las personas suscritas al mismo podrán leerlo y contestarlo tanto personalmente como al grupo en general, pudiendo por tanto aprender mucho sobre el tema tratado. La variedad de temas disponibles en los diferentes grupos permiten la participación activa, yendo desde la simple lectura hasta la colaboración con el grupo respondiendo a preguntas que ayuden a otros lectores a desarrollarse.

2.6.1. CNEWS

Fue diseñado para servidores que llevan noticias sobre enlaces UUCP. Funciona bajo cualquier Unix que se pueda encontrar y hay literalmente miles de sistemas usándolo alrededor del mundo.

Su mayor desventaja es que parece haber sido diseñado para conexiones UUCP por módem, y por tanto requiere la adición de un servidor NNTP para manejar transferencia en tiempo real de noticias por Internet.

La distribución newspack de sunsite contiene ficheros de configuración que funcionan en Cnews Cleanup Release bajo Linux, así como un parche de un par de líneas necesario para evitar algunos problemas de "doexplode" con bash1.12.

2.6.2. INN (InterNetNews)

Su mayor ventaja es la velocidad y el hecho de que contiene un servidor NNTP integrado. Su principal desventaja no se instala y funciona necesariamente en todos los Unix estándar todavía. Además, opera con un demonio (innd) siempre funcionando.

Los administradores de noticias noveles probablemente no deberían intentar instalar INN hasta que tengan experiencia con B-news o Cnews. A pesar de ser rápido y versátil, está prácticamente sin documentar para el principiante.

INN es muy quisquilloso con los permisos. También es muy delicado con tener un protocolo TCP/IP de calidad para trabajar. Linux no cumple necesariamente este requisito, así que se recomienda obtener una distribución de INN específica para Linux, en cualquier de los servidores de archivos de Linux.

2.6.3. NNTP (Network News Transfer Protocol)

Proporciona una forma de intercambio de noticias totalmente diferente de Cnews, para adaptarse a los protocolos de transporte usados en la Red. NNTP (Protocolo de Transferencia de Noticias de Red), y no consiste en un paquete de programas en particular, sino que es un estándar de Internet. Esta basado en una comunicación orientada a la conexión generalmente sobre TCP entre un cliente en algún lugar de la red, y un servidor que almacena las noticias en disco.

La conexión de flujo permite al cliente y al servidor negociar la transferencia de artículos interactivamente, sin apenas retrasos, manteniendo bajo el número de artículos duplicados.

Junto con los altos ratios de transferencia de Internet, esto supone un transporte de noticias que supera ampliamente a las redes UUCP originales. Mientras que hace algunos años no era extraño que un artículo tardase dos semanas o más en llegar hasta el último rincón de Usenet, ahora suele tardar menos de dos días; en la propia Internet, es incluso cuestión de minutos.

Varios comandos permiten a los clientes obtener, enviar y publicar artículos. La diferencia entre enviar y publicar es que esto último puede incluir a artículos con cabeceras incompletas. La obtención de artículos puede ser usada por clientes de transporte de noticias o por lectores de noticias. Esto hace del NNTP una excelente

herramienta para proporcionar acceso a muchos clientes de una red local sin tener que pasar por las dificultades que implica usar NFS.

2.6.3.1. Restricciones de acceso

El acceso a los recursos NNTP se rige por el fichero nntp_access en /usr/lib/news. Las líneas del fichero describen los derechos de acceso para computadores ajenos. Cada línea tiene el siguiente formato:

site read|xfer|both|no post|no [!exceptgroups]

Si un cliente se conecta al puerto NNTP, nntpd intenta obtener su nombre completo en la red a partir de su dirección IP. El nombre del computador del cliente y su dirección IP son contrastados con el campo site de cada entrada, en el mismo orden en el que aparecen en el fichero. Las coincidencias pueden ser parciales o exactas. Si una entrada coincide exactamente, se aplica; si la coincidencia es parcial, solo se aplica si no hay otra entrada posterior igual o mejor. site puede especificarse de una de las siguientes formas:

- Nombre del host
- dirección IP
- nombre del dominio
- nombre de la red

- default

2.7. LISTAS DE CORREO

Una lista de correo consiste en una dirección de correo, aparentemente como las demás, pero con la característica de que al enviar un mensaje a dicha dirección lo reciben un conjunto de direcciones de correo previamente especificado.

Es decir, la lista es un distribuidor. Se le envía un mensaje y ella lo distribuye a todas las direcciones de correo que están subscritas.

Cada subscriptor tiene derecho a enviar mensajes a una dirección específica de la lista de correo, desde allí se redirige a todos los miembros de la lista que a su vez, pueden expresar sus opiniones de la misma forma, originándose así un enriquecedor intercambio de ideas. Algunas listas disponen de un moderador que se encarga de filtrar los mensajes que se reciben.

Las listas de correo se pueden utilizar con múltiples propósitos:

- **Crear foros de discusión sobre temas específicos.** Las personas interesadas en un tema en concreto se suscriben a una lista que trate dicho tema. A partir de ahí comienzan a recibir mensajes con los comentarios que escriben otras personas suscritas y las respuestas correspondientes. También pueden participar cuando

quieran sabiendo que el mensaje que envíen lo recibirán todos los interesados en el tema. Los mensajes de la lista se distribuyen de inmediato al buzón de correo electrónico de quienes manifestaron su interés suscribiéndose.

- **Recibir información periódica sobre un tema en concreto**. En este tipo de listas suele ser una única persona la que envía los mensajes y los integrantes de la lista quienes lo reciben. Por ejemplo, una lista para recibir información sobre actualizaciones y nuevas versiones de un programa.

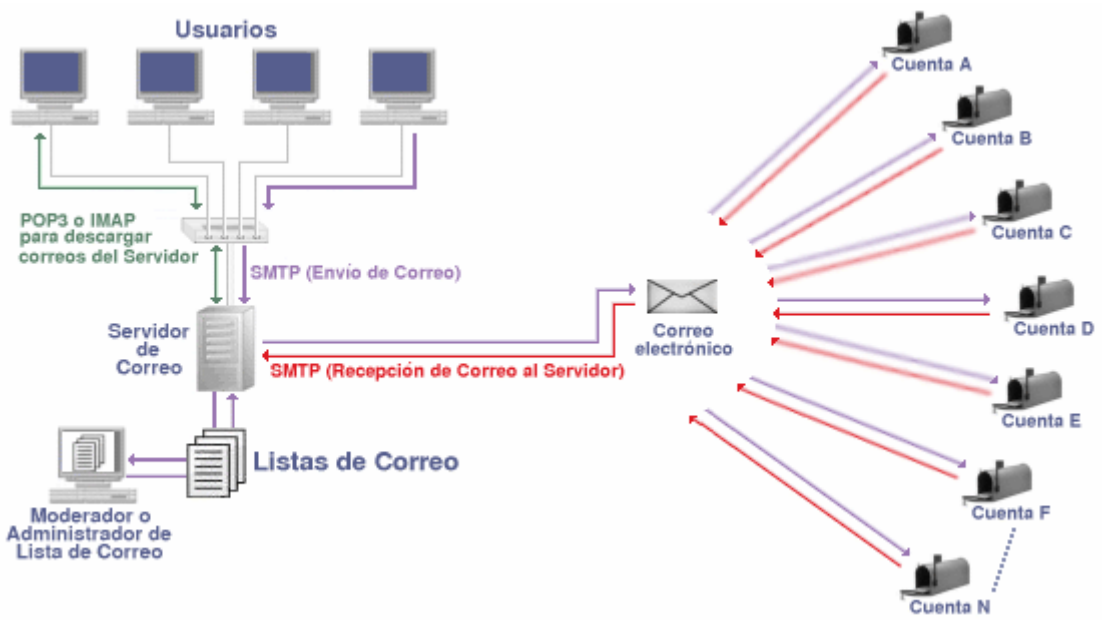

Figura 2.5. Representación funcionamiento de las listas de correo

Listas públicas o privadas. En cuanto a quién puede enviar mensajes a las listas, se puede escoger una de entre las siguientes opciones:

- Cualquiera puede enviar mensajes (lista abierta).
- Sólo los suscriptores pueden enviar mensajes (foro de discusión).

- Sólo el moderador puede enviar mensajes (lista privada).

2.7.1. BENEFICIOS

- **Seguridad.** Las direcciones registradas en las listas de correo electrónico se mantienen en confidencialidad. Lista negra configurable de direcciones de correo individuales o dominios enteros de los que no se admiten mensajes. Protege a la lista de abusadores del correo electrónico.
- **Administración.** No es necesario suscribir o dar de baja a las direcciones de correo. Puede ser hecho directamente por quienes deseen registrarse en una lista de correo o apuntarse en el foro. Una lista de correo le da la posibilidad de enviar un mensaje a múltiples buzones en su organización. Las listas de correo son ideales para distribuir boletines o memos fácil y eficientemente.
- **Facilidad de uso.** Si alguna persona envía un mensaje dirigido a la lista, el servidor se encarga de reenviar dicho mensaje a todas las direcciones que tenga asociadas esa lista.
- **Escalabilidad.**
- **Automatización.**

2.7.2. MAILMAN

Mailman es un software libre que permite gestionar listas de distribución, noticias y correo electrónicos. Mailman está integrado con la web, permitiendo a sus

usuarios una fácil administración de sus cuentas, así como a sus propietarios administrar las listas. Mailman incluye soporte para crear archivos de correos, procesamiento automático de correo rechazado, filtrado de contenido, envío en modo compendio o resumen, filtros spam, etc.

2.7.2.1. Características

- Soporta varios tipos de listas, incluyendo listas moderadas.
- Las listas son configurables por web.
- Soporta archivo y recuperación de mensajes.
- Soporta resúmenes (digests).
- Maneja gestión de rebotes (bounces).
- Maneja bajas masivas de usuarios.
- Maneja varios idiomas.
- Dispone filtros para remitentes, destinatarios.
- Maneja subscripciones masivas de usuarios.
- Maneja opciones de entrega regular.
- Está escrito en Python y es fácil de adaptar y extender

2.7.2.2. Tipos de subscripciones

- **Suscripción libre**: Cualquier persona puede solicitar apuntarse a una lista de distribución en concreto, y automáticamente queda subscrito tras esa petición.

- **Suscripción moderada**: Todas las peticiones de suscripción a una cierta lista de distribución deben ser aprobadas por el administrador o por los moderadores de dicha lista. Este método es el más útil, con lo que se evita que se apunten personas que no tengan nada que ver con la razón de existencia de dicha lista

2.7.2.3. Formas de envío de mensajes

- **Mensajes no moderados**: Todos los mensajes enviados por miembros de la lista son enviados automáticamente al resto de miembros, sin tener que aprobarlo el administrador o moderador
- **Mensajes moderados**: Todos los mensajes enviados por miembros de la lista tienen que ser aprobados por el administrador o moderador.
- **Mensajes de miembros no subscritos a la lista**: Se puede configurar para que sean eliminados automáticamente, o que los apruebe/deniegue el administrador o moderadores de dicha lista.

2.7.2.4. Requerimientos

- Python 2.1 o superior
- gcc 2.9 o superior
- Servidor Web Apache
- Sendmail, Exim, Postfix o algún otro MTA

2.7.3. MAJORDOMO

Majordomo es un programa que automatiza la gestión de listas de correo Internet. Básicamente realiza tres funciones:

- Se encarga de la distribución de mensajes a los usuarios de las listas
- Procesa comandos originados por usuarios finales a través del correo electrónico (subscribe, unsubscribe, help, etc.)
- Procesa comandos generados por los administradores de las listas, permitiendo su configuración remota.

2.7.3.1. Características

- Permite gestionar diversos tipos de lista, restringiendo en todo o en parte alguna de sus características: quién va a poder enviar mensajes a la lista, quienes la van a componer, qué tipo de información se va a poder solicitar, etc.
- Permite que, para determinado tipo de listas, un usuario de correo pueda suscribirse por sí mismo, sin necesidad de solicitarlo al administrador de la lista.
- Permite delegar la administración de una lista a personal ajeno al servicio de informática de la organización. Esta persona no tiene por qué tener

conocimientos informáticos para manejar la lista, le basta con saber utilizar unos pocos comandos.

- El administrador de la lista puede realizar su labor desde una ubicación remota, enviando los comandos en el cuerpo de un mensaje de correo electrónico. Majordomo le devuelve la respuesta también a través de e-mail.
- Soporta resúmenes (digests)
- Está escrito en Perl, fácil de adaptar y extender

2.7.3.2. Requerimientos

- Perl 4.036* o Perl 5.002* o superior
- Compilador de *C*.

2.8. WEBMAILS

Front-end para leer el correo electrónico del servidor de correos por medio de una pagina web. Comúnmente se usa front ends para leer correo como lo son Outlook, Kmail, Pine, etc. Estos son comúnmente llamados clientes de correo, pero estos deben de estar correctamente configurados en su maquina y solo puede leerlos desde su propia máquina. El webmail hace mas fácil la lectura del correo en cualquier parte donde se encuentre porque para leerlo solamente se necesita de un navegador web como lo son: Netscape, Mozilla, Nautilus, Opera, lynx, y hasta Internet Explorer.

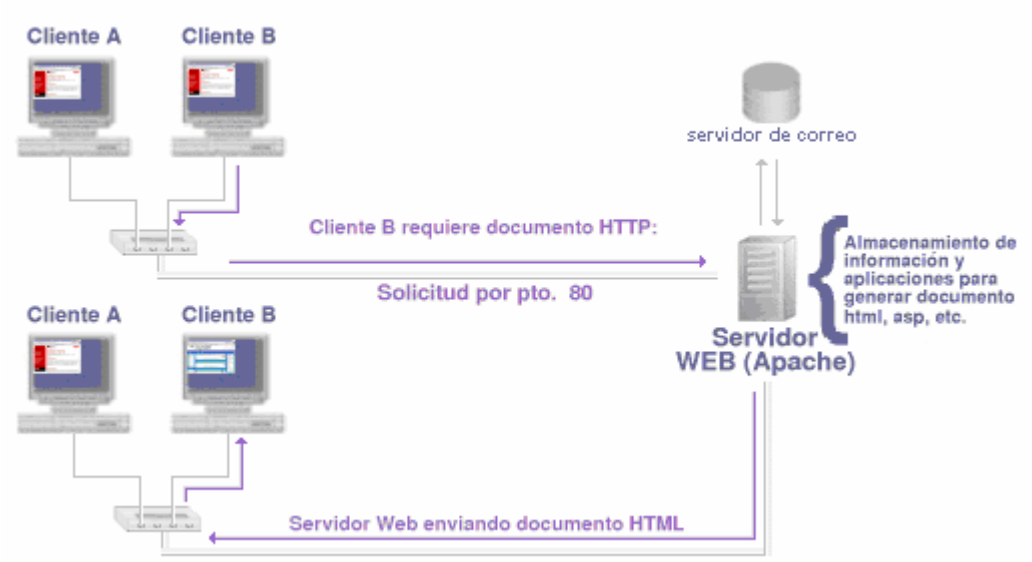

Figura 2.6. Representación de servidor Web

2.8.1. SQUIRRELMAIL

SquirrelMail es un paquete de correo por web basado en estándares y escrito en PHP 4. Incorpora soporte PHP para los protocolos IMAP y SMTP, y todas sus páginas se crean en puro HTML 4.0 (sin requerir el uso de JavaScript), de modo que se garantice la máxima compatibilidad entre navegadores. Tiene muy pocos requerimientos y es muy fácil de instalar y configurar. SquirrelMail tiene toda la funcionalidad que se espera de un cliente de correo electrónico, incluyendo soporte de MIME, agendas de contactos y gestión de carpetas.

2.8.1.1. Características

- Gestión de carpetas**.**

- Internacionalización.
- Libro de direcciones personal y acceso a otros servicios de LDAP. Permite hacer búsquedas de direcciones.
- Gestión de attachments**.**
- Servicio de búsqueda en emails.
- No necesita ninguna base de datos para funcionar (al contrario que muchos otros webmails que necesitan MySQL o PostgreSQL).
- Interfaz de usuario fácil y potente.
- Arquitectura de plug-in.
- Múltiples temas.
- Configuración de las vistas de mensajes: número de mensajes visibles en pantalla, campos visibles, orden, cada cuanto tiempo comprueba si hay nuevos mensajes, etc.
- Posibilidad de añadir direcciones de correo entrantes o contenidas en un email a la libreta de direcciones de forma automática.
- Auto completado de direcciones de correo cuando se escribe un email.
- Envio de páginas HTML comprimidas (en caso de ficheros largos).
- Filtros de mensajes según direcciones de correo o subject.
- Filtrado de spam.
- Descarga de correo de múltiples cuentas POP.
- Utilidad de corrección de correos en cualquier idioma. Esto me dejó con la boca abierta.
- Traducción de correos a diferentes lenguas.

2.8.1.2. Requerimientos

- PHP4.
- Servidor Web
- Servidor IMAP

2.8.2. NEOMAIL

Cliente de correo basado en Web que pude ser instalado en servidores Unix que ejecutan servicios Web.

2.8.2.1. Características

- Envía y recibe mensajes con múltiples attachments
- Despliega imágenes de los attachments
- Amigable, atractivo, basado en icono.
- Soporta múltiples lenguajes, incluyendo inglés, español, alemán, francés, húngaro, italiano, holandés, polaco, portugués, noruego, rumano, ruso, eslovaco, y más puede agregarse fácilmente.
- Configurable: limita en el tamaño de los attachments salientes, despliega el espacio del buzón, tamaño de la libreta de direcciones.
- Los usuarios pueden importar su libreto de direcciones de Outlook Express o Netscape Mail en el formato de CSV

2.8.3. OPEN WEBMAIL

Webmail basado en Neomail. Open Webmail está orientado a la operación con archivos de bandejas de gran tamaño con un uso eficiente de la memoria. También provee varias características para ayudar a los usuarios a migrar desde Microsoft Outlook sin problemas.

2.8.3.1. Características

- Acceso rápido a bandejas
- Movimiento de mensajes eficiente
- Poco uso de memoria
- Manejo apropiado de bandejas y mensajes
- Bloqueo de archivos elegante
- Relaying de SMTP remoto
- Virtual hosting
- Alias de usuarios
- Soporte de usuarios virtuales puros
- Capacidad de configuración por cada usuario
- Varios módulos de autenticación
- Búsqueda por contenido
- Soporte completo de MIME (en presentación y redacción)
- Soporte de bandeja de borradores

- Soporte de respuestas con membrete
- Soporte de verificación ortográfica
- Soporte de correo POP3
- Soporte de filtros de correo
- Previsualización de cantidad de mensajes
- Soporte de confirmación de lectura
- Conversión automática de conjuntos de caracteres
- Soporte de calendario con recordatorio/notificación
- Soporte de disco web
- Ejecución persistente a través de SpeedyCGI
- Soporte de compresión http

2.8.3.2. Requerimientos

- Servidor web Apache con cgi habilitado
- Perl 5.005 o superior
- CGI.pm-3.05 (requerido)
- MIME-Base64-3.01 (requerido)
- libnet-1.19 (requerido)
- Text-Iconv-1.2 (requerido)
- libiconv-1.9.1 (requerido si el sistema no soporta iconv)
- CGI-SpeedyCGI-2.22 (opcional)
- Compress-Zlib-1.33 (opcional)

- ispell-3.1.20.tar.gz (opcional)
- Quota-1.4.10 (opcional)
- Authen-PAM-0.14 (opcional)
- ImageMagick-5.5.3 (opcional)

2.9. SEGURIDAD Y CRIPTOGRAFÍA

2.9.1. CONCEPTOS BÁSICOS

2.9.1.1. Sistemas de Claves Públicas

Para poder entender mejor el sistema de codificación usado por los sistemas de claves asimétricas (claves públicas y privadas), es necesario entender las diferencias con los sistemas de claves simétricas (claves secretas).

Los sistemas de cifrado con clave simétrica son aquéllos en los que la clave que se usa para cifrar una serie de datos, es la misma que la que se usará para descifrar estos datos. En el caso del correo electrónico, el remitente cifraría el mensaje con una clave secreta, y para que el destinatario pueda descifrarlo, necesitaría haber obtenido previamente esta misma clave de un modo «seguro», o sea de modo que la clave no haya podido ser interceptada durante la entrega. Si no se tiene la completa seguridad de que el intercambio de la clave ha sido seguro, la validez de este sistema es nula.

Por el contrario, los sistemas de cifrado con claves asimétricas usan claves distintas para el cifrado y posterior descifrado de los datos. En un caso como el anterior, el remitente usaría la clave pública del destinatario para cifrar el mensaje, y el destinatario descifraría el mensaje con su propia clave privada. Así pues, la clave privada no debe ser accesible para nadie que no sea el propio dueño de la misma, mientras que la clave pública, puede ser entregada a cualquier persona. En un sistema de cifrado bien implementado, la clave privada no debe derivar nunca de la clave pública.

2.9.1.2. Firmas Digitales

El concepto de la firma digital se basa en la verificación de la autoría de un mensaje. Esto quiere decir que se puede comprobar que el destinatario del mensaje puede comprobar que el «supuesto» remitente es quien afirma ser. Para ello, el remitente, una vez compuesto el mensaje, lo firma usando su propia clave privada. El destinatario, una vez ha recibido el mensaje, comprobará la veracidad de éste, esto es, lo verificará usando la clave pública del remitente.

Este método es de especial utilidad para reducir riesgos de seguridad en nuestros sistemas (nos podrían enviar un supuesto parche para un programa, y éste en realidad ser un virus o un troyano); también podrían enviarnos información o datos, como provenientes de una fuente lícita o fiable. En ambos casos, no sería muy difícil

falsificar la dirección y nombre del remitente, pero sí imposible falsificar la firma digital de éste.

Como ya hemos dicho, la verificación de un mensaje firmado digitalmente se lleva a cabo mediante el uso de la clave pública del remitente sobre el texto del propio mensaje. De este modo no sólo podemos verificar la identidad del autor, sino que también podemos comprobar la integridad del mensaje, ya que la firma digital ha sido generada con el texto y la clave privada. Así pues, una alteración o modificación del texto «a posteriori», o cualquier manipulación del mensaje (especialmente si hacemos uso de las especificaciones MIME/PGP), daría como resultado un error en la verificación.

2.9.1.3. Anillos de Confianza

Un punto flaco en los algoritmos de clave asimétrica es la transmisión del código público. Es posible que una persona ponga en circulación código con un identificador de usuario falso. Si se codifican mensajes con este pseudo código, el intruso los puede descodificar y leerlos.

La solución PGP (y por consiguiente la solución GnuPG) está en firmar los códigos. La clave pública de un usuario puede estar firmada con las claves de otros usuarios. El objetivo de estas firmas es el de reconocer que el UID (identificador de usuario) de la clave pertenece al usuario a quien dice pertenecer. A partir de ahí es un

problema de cada usuario de GnuPG el decidir hasta qué punto se puede fiar de la firma. Una clave se puede considerar fiable cuando se confía en el remitente y cuando se sabe con seguridad que dicha clave pertenece a éste. Sólo cuando se puede confiar plenamente en la clave del firmante, se puede confiar en la firma que acompaña a la clave de un tercero. Para tener la certeza de que la clave es correcta hay que compararla con la huella digital por medio de canales fiables (por ejemplo, podríamos buscar el teléfono en la guía y llamarle, y que nos la dijera de palabra para poder compararla), antes de darle una confianza absoluta.

2.9.1.4. Límites de Seguridad

Si lo que se desea es mantener la confidencialidad de los datos que se poseen, no basta con determinar qué algoritmo de cifrado se va a usar; también es necesario pensar en la seguridad general del sistema. En principio, PGP está considerado como suficientemente seguro, y hasta el momento no se sabe que haya habido ningún incidente en el que una clave PGP haya sido descodificada. Pero eso no significa que todo lo cifrado sea seguro; si la NSA (Agencia de Seguridad Nacional de los EE.UU.) hubiera conseguido descodificar una clave PGP mediante criptoanálisis, analización del código, o cualquier otro modo, no es probable que lo hicieran público. Pero aún en el caso de que las claves PGP fueran a todas luces imposibles de descodificar, otros tipos de ataques a la seguridad pueden ser utilizados.

Otra posibilidad técnica, aunque más difícil, es la de los troyanos que recogen entradas de teclado y las transmiten al asaltante. También es posible, aunque muy difícil, pasar el contenido de una pantalla a otra. En este último caso no sería necesario ningún análisis sobre datos cifrados, ya que se obtendrían «pre-cifrados».

Por todo esto es necesaria una planificación de la seguridad que esté bien prevista y que minimice los riesgos.

La idea no es la de recrear una atmósfera de paranoia entre la gente, sino dejar claro que para implementar un sistema seguro no basta con la instalación de un programa criptográfico, que si bien es un paso hacia un sistema más seguro, no es una solución completa. Troyanos como el aparecido en Marzo de 1999 (Melissa) probaron que muchas compañías no se encuentran preparadas en temas de seguridad.

2.9.2. SASL (Simple Authentication and Security Layer)

Método para añadir soporte para la autenticación a protocolos basados en la conexión que ha sido estandarizado por la IETF (Internet Engineering Task Force). Se usa en servidores para manejar las peticiones de autenticación de los clientes. Para ello, el protocolo incluye un comando para identificar y autenticar un usuario contra un servidor y para, opcionalmente, negociar la protección de las subsiguientes interacciones del protocolo. Si se negocia su uso, una capa de seguridad es añadida entre el protocolo y la conexión.

La librería SASL de Cyrus también usa la librería OpenSSL para encriptar los datos.

2.9.3. PGP (Pretty Good Privacy)

Es un software que permite al usuario tener tres características en el envío de mensajes. Estas características son las siguientes:

- **Intimidad**. Sólo pueden leer el mensaje aquellas personas a quienes va dirigido.
- **Autentificación**. Los mensajes que parecen ser de una persona sólo pueden venir de esa persona.
- **Integridad**. Si un mensaje va firmado, no se puede modificar el contenido del mensaje.

Para enviar un mensaje de forma que nadie excepto la persona a la que va dirigido pueda leerlo, éste se encripta o cifra, esto es, se resuelve de forma que se hace ilegible. Cuando el mensaje llega al receptor, éste se descifra, es decir, se vuelve a hacer legible. De este modo se consigue intimidad.

La autentificación y la integridad se consiguen firmando el mensaje que será enviado.

PGP normalmente usa criptografía de clave pública, pero también puede utilizar criptografía convencional.

La criptografía de clave pública se basa en la utilización de dos claves para cada usuario: una clave pública (que es conocida por todos) y una clave secreta (que sólo es conocida por el propio usuario). Si una persona X desea enviar un mensaje a otra persona Y, la persona X pondrá en el mensaje su clave secreta y la clave pública de la persona Y. Cuando el mensaje sea recibido por la persona Y, esta utilizar su clave secreta para descifrarlo. Como la clave secreta de la persona Y solo es conocida por la persona Y, sólo ella podrá descifrar el mensaje. De este modo se consigue confiar.

La criptografía convencional sólo utiliza una clave. Si una persona X desea enviar un mensaje a una persona Y, ambas personas se pondrán de acuerdo y elegirán una clave que sólo ellos conocerán para cifrar el mensaje. La persona X pondrá dicha clave en el mensaje y la persona Y descifrará el mensaje con esa misma clave. El inconveniente de este sistema es la necesidad de un canal seguro para transmitir la clave y ponerse de acuerdo sobre ella.

2.9.4. GnuPG (Gnu Privacy Guard)

GnuPG es una herramienta que se usa para las comunicaciones seguras; es un reemplazo gratuito de la tecnología de encriptación PGP (Pretty Good Privacy, una aplicación de encriptación muy conocida). Con GnuPG, puede codificar sus datos y

su correspondencia y autentificar ésta con una firma digital. GnuPG es también capaz de descifrar y verificar PGP 5.x.

Debido a que la herramienta GnuPG es compatible con otros sistemas estándares, su correspondencia segura será también compatible con otras aplicaciones de correo electrónico en otros sistemas operativos, tales como Windows y Macintosh.

GnuPG usa la criptografía de clave pública para asegurar a los usuarios un intercambio de datos seguro. En un esquema de criptografía de clave pública, se tiene que crear dos claves: una pública y otra privada. La clave pública se intercambia con aquellas personas con las que se comunica o con el servidor de claves pero nunca debe revelar la clave privada.

La encriptación va a depender del uso de las claves. En criptografía tradicional, ambas partes tienen la misma clave que usan para descodificar cada una de las transmisiones de información. En la criptografía de clave pública, coexisten dos claves: una pública y otra privada. Normalmente, una persona o una organización dan a conocer su clave pública y se reserva la privada. Los datos codificados con la clave pública sólo pueden ser descifrados con la privada y viceversa.

2.9.5. LDAP (Lightweight Directory Access Protocol)

Conjunto de protocolos abiertos usados para acceder información guardada centralmente a través de la red. Está basado en el estándar *X.500* para compartir directorios, pero es menos complejo e intensivo en recursos. Por esta razón, a veces se habla de LDAP como X.500 Lite.

LDAP organiza la información en un modo jerárquico usando directorios. Estos directorios pueden almacenar variedad de información y se pueden incluso usar de forma similar a Network Information Service (NIS), permitiendo que cualquiera pueda acceder a su cuenta desde cualquier máquina en la red activa LDAP.

La mayor ventaja de LDAP es que información para toda una organización se puede consolidar dentro de un repositorio central. Por ejemplo, en vez de administrar listas de usuarios para cada grupo dentro de una organización puede usar LDAP como directorio central accesible desde cualquier parte de la red. Puesto que LDAP soporta Secure Sockets Layer (SSL) y Transport Layer Security (TLS), los datos delicados se pueden proteger de los curiosos.

LDAP también soporta un número de bases de datos back-end en las que se guardan directorios. Esto permite que los administradores tengan la flexibilidad para desplegar la base de datos más indicada para el tipo de información que el servidor tiene que diseminar. También, ya que LDAP tiene una interfaz de programación de

aplicaciones (API) bien definido, el número de aplicaciones activadas para LDAP son numerosas y están aumentando en cantidad y calidad.

En la parte negativa, LDAP puede ser complicado de configurar.

2.9.6. KERBEROS

Kerberos es un protocolo de seguridad creado por MIT que usa una criptografía de claves simétricas para validar usuarios con los servicios de red, evitando así tener que enviar contraseñas a través de la red. Al validar los usuarios para los servicios de la red por medio de Kerberos, se frustran los intentos de usuarios no autorizados que intentan interceptar contraseñas en la red.

2.9.6.1. Ventajas

La mayoría de las redes usan esquemas de autenticación basados en contraseñas. Tales esquemas requieren que cuando un usuario necesita una autenticación en un servidor de red, debe proporcionar un nombre de usuario y una contraseña. Lamentablemente, la información de autenticación para muchos servicios se transmite sin estar encriptada. Para que un esquema de este tipo sea seguro, la red tiene que estar inaccesible a usuarios externos, y todos los usuarios de la red deben ser de confianza.

Aún en este caso, una vez que la red se conecte a la Internet, ya no puede asumir que la red es segura. Cualquier intruso del sistema con acceso a la red y un analizador de paquetes pueden interceptar cualquier contraseña enviada de este modo, comprometiendo las cuentas de usuarios y la integridad de toda la infraestructura de seguridad.

El primer objetivo de Kerberos es el de eliminar la transmisión a través de la red de información de autenticación. Un uso correcto de Kerberos erradica la amenaza de analizadores de paquetes que intercepten contraseñas en su red.

2.9.6.2. Desventajas

A pesar de que Kerberos elimina una amenaza de seguridad común, puede ser difícil de implementar por una variedad de razones:

- La migración de contraseñas de usuarios desde una base de datos de claves estándar UNIX, tal como /etc/passwd o /etc/shadow, a una base de datos de contraseña Kerberos puede ser tediosa y no hay un mecanismo rápido para realizar esta tarea.
- Kerberos es sólo parcialmente compatible con los Pluggable Authentication Modules (PAM) usados por la mayoría de los servidores en Red Hat Linux.

- Para que una aplicación use Kerberos, el código debe ser modificado para hacer las llamadas apropiadas a las librerías de Kerberos. Para algunas aplicaciones, esto puede suponer un esfuerzo excesivo de programación.
- Kerberos presupone que se está utilizando hosts fiables en una red no fiable. Su primer objetivo es el de prevenir que las contraseñas en texto plano sean enviadas a través de la red.

2.9.7. ACL (Access Control Lists, Listas de Control de Acceso)

De manera tradicional, para cada objeto en Linux se definen tres grupos de permisos. Estos grupos reflejan los permisos de escritura (w), lectura (r) y ejecución (x) para las tres clases de usuarios: propietario del archivo (owner), grupo (group) y el resto (other). Además es posible definir los bits set user id, set group id y sticky.

Las ACLs intervienen en las situaciones en las que el concepto tradicional de permisos para archivos resulta insuficiente. Estas permiten asignar permisos a determinados usuarios o grupos, incluso cuando estos permisos no coinciden con los del propietario del archivo o su grupo.

Las listas de control de acceso son una característica del kernel de Linux y actualmente están soportadas por ReiserFS, Ext2, Ext3, JFS y XFS. Con su ayuda es posible llevar a la práctica complejos escenarios sin que sea necesario implementar complicados modelos de permisos a nivel de aplicaciones.

Access ACL. Los permisos de acceso de usuarios y grupos a cualquier objeto del sistema (archivos y directorios) se definen a través de las access ACLs (ACLs de acceso).

Default ACL. Las default ACLs (ACLs predeterminadas) sólo pueden aplicarse a directorios y definen los permisos que un objeto del sistema "hereda" del directorio superior al ser creado.

Entrada ACL. Una ACL está formada por una serie de entradas ACL (ACL entries). Una entrada ACL consta de un tipo, un indicador del usuario o el grupo al que se refiere la entrada, y los permisos en sí. En algunos tipos de entrada, el indicador para el usuario o el grupo está vacío.

2.9.7.1. Estructura de las entradas

Las ACLs pueden dividirse fundamentalmente en dos clases. Una ACL estándar consiste exclusivamente en las entradas de tipo owner (propietario), owning group (grupo propietario) y other (otros) y coincide con los bits de permisos tradicionales para archivos y directorios. Una ACL extendida (extended) contiene además una entrada mask (máscara) y puede incluir varias entradas del tipo named user (usuario identificado por el nombre) y named group (grupo identificado por el nombre). La siguiente tabla ofrece un resumen de los distintos tipos de entradas ACL.

Tipo	Formato en texto
Owner	user::rwx
named user	user:name:rwx
owning group	group::rwx
named group	group:name:rwx
Mask	mask::rwx
Other	other::rwx

Tabla 2.5. Tipos de entrada ACL

Los permisos definidos en las entradas owner y other siempre tienen vigencia. Excepto la entrada mask, el resto de entradas (named user, owning group y named group) pueden estar activadas o bien enmascaradas. Si se han definido permisos tanto en las entradas mencionadas en primer lugar como en las máscaras, tendrán validez. Los permisos que sólo han sido definidos en la máscara o en la propia entrada, no tienen validez.

Tipo	Formato en texto	Permisos
named user	User:jane:r-x	r-x
Mask	mask::rw-	rw-
		r--

Tabla 2.6. Enmascaramiento de permisos de acceso

2.9.7.2. Efecto de una acl predeterminada

Los permisos de acceso en la ACL predeterminada son heredados de forma distinta por archivos y subdirectorios:

- Un subdirectorio hereda la ACL predeterminada del directorio superior como propia default ACL y además como access ACL.
- Un archivo hereda la ACL predeterminada como propia access ACL.

Todas las llamadas del sistema (system calls) que crean objetos del sistema utilizan un parámetro mode. Este parámetro se encarga de definir los permisos de acceso sobre el nuevo objeto del sistema:

- Si el directorio superior carece de ACL predeterminada, los permisos resultantes son los introducidos en el parámetro mode menos los permisos asignados en umask.
- Si existe una ACL predeterminada para el directorio superior, se asignan al objeto los bits de permiso resultantes de la intersección de los permisos del parámetro mode y de los que contiene la ACL predeterminada. En este caso no se tiene en cuenta umask.

2.9.8. IPTABLES

Iptables es una extensión del kernel, es decir, el propio sistema se encarga de su gestión. Su función consiste básicamente en analizar todo el flujo de tráfico entrante y saliente hacia y desde él y tomar unas decisiones sobre cada paquete en base a las reglas definidas.

La estructura de Iptables es básicamente una cola: cuando un paquete llega, este es validado contra cada una de las reglas del firewall, en el momento que alguna regla coincide, se ejecuta la acción que haya sido definida en la regla (descartar el paquete, aceptarlo, enrutarlo, etc.).

2.9.8.1. Sintaxis básica de iptables

iptables −t [tabla] −[AIRDLFZNXP] [regla] [criterio] −j [acción]

Tablas para las diferentes funciones de un filtro de paquetes.

- **filter.** En esta tabla, que contiene la mayoría de reglas, se realiza el verdadero filtrado de paquetes y se definen las reglas para aceptar (ACCEPT) o rechazar (DROP) paquetes.

- **nat**. Esta parte define la modificación de las direcciones de origen y destino de los paquetes. El enmascaramiento o masquerading, que se utiliza para conectar una pequeña red privada a Internet, es un caso especial de NAT.
- **mangle.** Las reglas en este apartado permiten editar valores en el encabezamiento del paquete.

Dentro de las tablas mencionadas existen varias cadenas predefinidas por las que tienen que pasar los paquetes:

- **PREROUTING.** Esta cadena se aplica a paquetes que acaban de llegar al sistema.
- **INPUT.** Esta cadena se aplica a paquetes que se ocupan de procesos internos del sistema.
- **FORWARD.** Esta cadena se aplica a paquetes que atraviesan el sistema sin ser modificados.
- **OUTPUT.** Esta cadena se aplica a paquetes generados en el propio sistema.
- **POSTROUTING.** Esta cadena es para todos los paquetes que salen del sistema.

Ordenes básicas

- **A.** Añade (Append) una regla. Reglas válidas: INPUT, FORWARD y OUTPUT.
- **L.** lista las reglas.

- **F.** Borra todas las reglas en caso de INPUT, FORWARD o OUTPUT sean dados como argumento, se borraran las reglas asociadas solo a esa clase.
- **P.** Establece la política por defecto del firewall. Por defecto es aceptar todas las conexiones.

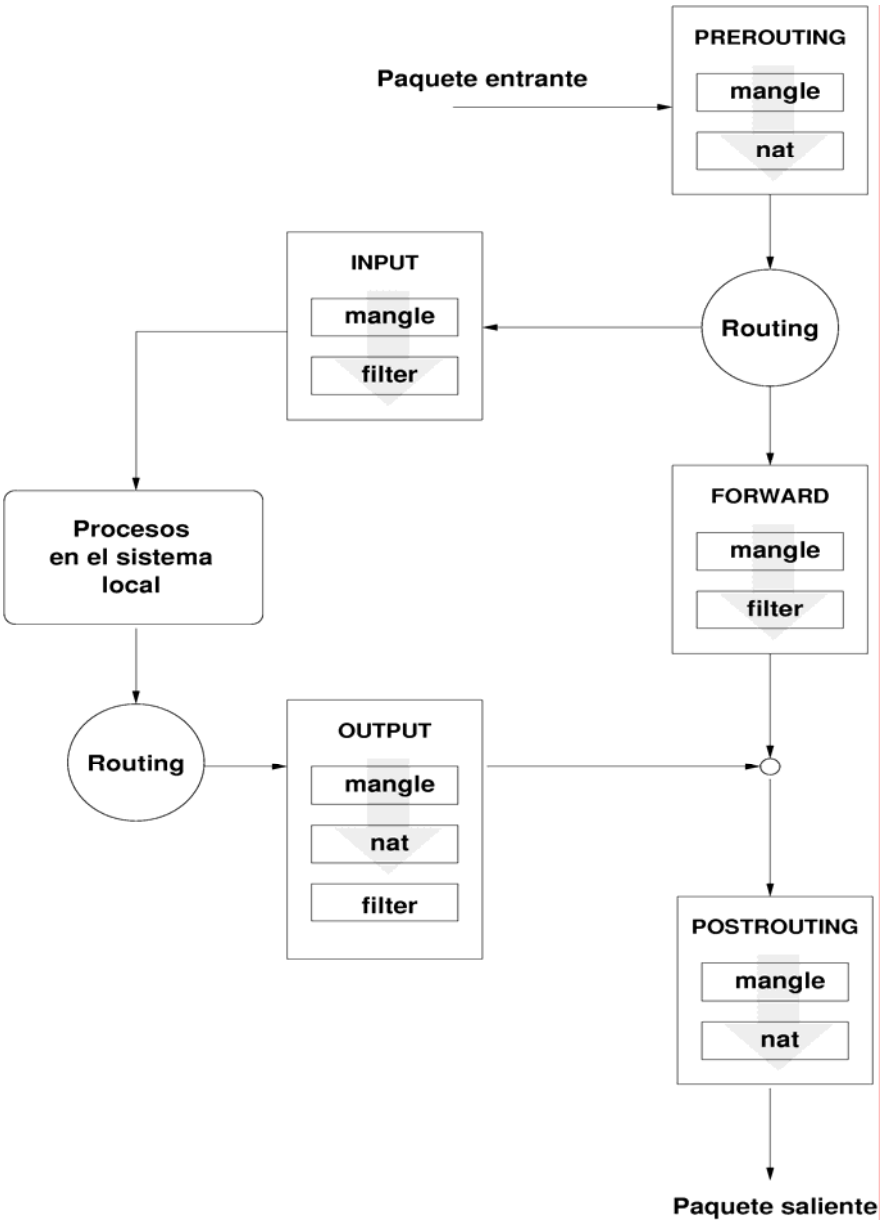

Figura 2.7. Rutas de un paquete por el sistema

Los parámetros utilizados para filtrar son los siguientes:

- **-t <tabla>** Especifica la tabla sobre la cual se trabajará. Por ejemplo: -t nat
- **-i** <interfaz> Especifica la interfaz de red por la que entra el paquete. Por ejemplo: -i eth0
- **-o** <interfaz> Indica la interfaz de red por la que sale el paquete. Por ejemplo: -o eth0
- **-p** <protocolo> Especifica el protocolo del paquete. Por ejemplo: -p tcp
- **-s** <ip> Especifica la ip de origen (o red de la que procede) del paquete. Por ejemplo: -s 192.168.0.2 para especificar una ip, o bien -s 192.168.0.0/24 para especificar una red de origen.
- **-d** <ip> Igual que en el caso anterior pero para la ip destinataria del paquete.
- **--dport** <puerto> Especifica el puerto al que va dirigido el paquete. Por ejemplo: --dport 22, o bien --dport 1:1024 (para especificar un rango de puertos).
- **-j** <accion> Se establece que es lo que hay que hacer con el paquete. Las posibles opciones son: ACCEPT, REJECT, DROP, REDIRECT, LOG (existen más, pero estas son las básicas).

    - **ACCEPT** aceptará el paquete.
    - **REJECT o DROP** lo desecharán, la diferencia entre ellos reside en que DROP descartará el paquete silenciosamente y REJECT emitirá un paquete ICMP Port Unreachable, indicando que está cerrado.

- **REDIRECT** redirigirá el paquete a donde se indique en el criterio del comando y por último...
- **LOG** lo logeará para su posterior análisis.

2.10. SPAM

El spam es el hecho de enviar mensajes electrónicos (habitualmente de tipo comercial) no solicitados y en cantidades masivas. Aunque se puede hacer por distintas vías, la más utilizada entre el público en general es la basado en el correo electrónico. Otras tecnologías de Internet que han sido objeto de spam incluyen mensajes, grupos de noticias UseNet, motores de búsqueda y blogs. El spam también puede tener como objetivo los teléfonos celulares (a través de mensajes de texto) y los sistemas de mensajería instantánea.

2.10.1. TÉCNICAS DE SPAM

2.10.1.1. Spam en buzones de correos electrónico.

En este contexto un spam es un mensaje publicitario, casi siempre de carácter comercial, no solicitado y que se manda al buzón electrónico, también conocido como UCE (Unsolicited Commercial Email).

Estos mensajes tienen índole diversa. Puede ser que inviten a visitar una web, generalmente porno, que indique cómo conseguir dinero navegando, que oferte una serie de equipos informáticos a buen precio.

Lo que todos esos mensajes tienen en común, y es lo que los define como spam, es el no haber sido solicitados, el no poder rechazarse, y el buscar un beneficio por parte de la persona, empresa, o institución que lo manda. Es decir, que son anuncios publicitarios.

No parece necesario que ese beneficio sea de carácter económico. Podría darse la situación en la que una organización religiosa, política, o de otro tipo, bombardeara los buzones de correos en un intento de hacer seguidores, ganar votos, o conseguir directamente dinero. En cualquier caso sí parece haber en todo spam ese denominador común que es el beneficio, sea directamente económico o de otro tipo.

A veces este spam se presenta hábilmente ambiguo. Es típico el mensaje, con información comercial, que se presenta como dirigido a otra persona aunque, por error, haya llegado a tu dirección e-mail.

El spam significa un incremento innecesario, y a veces desproporcionado, del tráfico de red. Ese incremento se termina pagando en tanto que al ocupar recursos y tiempos, impide que los usuarios legales y normales puedan ejercerlos con la misma eficacia y normalidad de la que haría uso si el spam no existiera.

Un spammer puede mandar miles de mensajes spam a través de un servidor de correo que, en general, ni siquiera es el de su proveedor (lo hace así para dificultar su identificación) pues bien, eso supone un gasto en tiempo y recursos de un servicio que ni ha pagado ni legalmente se le permite. Por eso, en algunos países, la identificación del spammer que utilizó de forma fraudulenta un servidor de correo que no es el suyo para lanzar su spam, puede acabar en los tribunales donde ya ha habido sentencias judiciales condenatorias.

2.10.1.2. Spam en grupos de noticias a nivel de servidor.

Lo que en el presente los servidores de noticias consideran mensaje spam es el mismo mensaje, o copias sustancialmente idénticas del mismo, que se postean un número excesivo de veces, ya sea de modo individual (EMP), o colectivamente (ECP), a varios grupos.

En el primer caso se denomina EMP, que son las siglas inglesas de Excessive MultiPosting, y entonces el spammer mandará copias, sustancialmente idénticas, del mismo mensaje, en distintos posteos, a los distintos grupos. La forma de determinar si un mensaje es "sustancialmente idéntico" a otro está sometida a reglas.

En el segundo ECP, que son las siglas inglesas de Excessive CrossPosting, y entonces el spammer introducirá en la cabecera el nombre de los distintos grupos de modo que, con un sólo posteo, se anuncie en todos ellos.

A esta denominación para spam se llegó con el uso extendido del término, ya que inicialmente era sinónimo del EMP. Hoy, sin embargo, engloba al conjunto de ambas categorías. Con todo, en el presente, no hay software capaz de superar el índice BI únicamente realizando ECP; habría que mandar el mismo mensaje a 400 grupos distintos.

Por tanto, desde el punto de vista de los servidores, el mensaje spam no se determina por el contenido sino, exclusivamente, por el número de mensajes enviados en un tiempo determinado.

El número concreto de mensajes que adquiere la categoría de spam se basa en un índice propuesto por Seth Breidbart, y que tenía la intención de medir la "malicia" que un spam concreto podía alcanzar en función del número de veces que se enviaba.

El índice Breidbart (BI) consiste en la raíz cuadrada de un número N, siendo N la suma del número de grupos en el que el mensaje, y sus copias, fueron enviados.

Por ejemplo, si una copia de un mensaje es crossposteada en 9 grupos y otra copia, sustancialmente idéntica, lo es en 16, entonces el BI de ese mensaje sería la suma de las raíces cuadradas de 9 y 16; es decir, la suma de 3 y 4, por tanto el BI sería 7.

Un mensaje se considera spam, para los servidores, cuando su BI es igual, o superior, a 20 en un periodo de 45 días.

No todos los servidores, ni jerarquías de grupos, aceptan este umbral. Hay jerarquías que funcionan de modo interno de modo más estricto, como por ejemplo bofh, net y free. Por otro lado hay servidores concretos que han decidido no aceptar los mensajes de cancelación que los demás servidores aceptan y emiten, o, por el contrario, establecen una política más restrictiva que la aquí indicada.

2.10.1.3. Spam en grupos de noticias a nivel de usuario.

La mayoría de los usuarios de las news consideran que, esos mismos mensajes que eran spam en el buzón e-mail, lo son también en los grupos de noticias. Es decir, para el usuario de news existen mensajes que son identificables como spam no por su frecuencia, sino por su contenido comercial ajeno a la temática del grupo en el que aparece.

La diferencia entre un mensaje off topic, es decir un mensaje que se encuentra fuera de la temática del grupo al que se postea, y un mensaje spam es que éste último, además de off topic, es comercial y es a eso a lo que los usuarios denominamos spam en news.

Son mensajes no queridos en un grupo de noticias y que, sin embargo, uno suele verse en la necesidad de bajar, ya que la alternativa de conectarse y bajar primero las cabeceras, limpiar lo que es spam, y conectarse después para bajar lo seleccionado,

casi se hace más caro, en dinero y, desde luego tiempo y comodidad, que bajarse todo y luego limpiar.

Son mensajes que gastan tiempo y atención, del mismo modo que el spam en el correo electrónico; análogamente a éstos no han sido solicitados por parte de los receptores del grupo de noticias, de hecho se encuentran claramente fuera del interés del grupo.

Por otro lado gastan recursos de los servidores de noticias. Eso repercute en una demora en los tiempos de conexión y, en general, en una pérdida en la eficacia y utilidad de los propios grupos de noticias, además de los posibles trastornos económicos en los servidores de noticias en los que se recibe y mantiene.

Hay que señalar, sin embargo, que éste punto de vista no es el tradicional. Desde la perspectiva tradicional los mensajes comerciales off topic a un grupo de noticias se han considerado como un tipo de off topic, más molesto que el normal pero a fin de cuentas sólo off topic. Este modo de ver el asunto está extendido, aunque no universalmente extendido, entre los ISP y los administradores de los servidores de noticias. Con todo, el punto de vista normal entre los usuarios actuales de los grupos de noticias parece ser el de considerar spam a los mensajes comerciales off topic.

2.10.2. ANTISPAM

2.10.2.1. Spamassassin

Filtro de correo que trata de identificar el spam mediante el análisis del texto y el uso en tiempo real de algunas listas negras a través de Internet.

A partir de su base de datos de reglas, utiliza un amplio abanico de pruebas heurísticas en las cabeceras y el cuerpo de los correos para identificar el spam. Una vez identificado, el correo puede ser opcionalmente marcado como spam o más tarde filtrado usando el cliente de correo del usuario.

Spamassassin normalmente identifica acertadamente entre un 95 y un 99% del spam, dependiendo del tipo de correo que se reciba. También incluye soporte para informar de mensajes de spam, automática o manualmente, a bases de datos como Vipul's Razor.

2.10.2.2. MailScanner

Mail Scanner es sistema antispammer y scanner de virus para el correo electrónico. Es capaz de detectar un gran número de tipos de correos electrónicos comerciales de publicidad (spam). No solo posibilita el escaneo de virus conocidos sino que también amplía su protección a los no conocidos, chequeando los archivos

adjuntos o attachments y rechazando los que contengan una serie de patrones que MS tiene predeterminados como no aceptados. En los mencionados patrones se puede destacar el de extensión de fichero mediante el cual rechaza mails que contengan una serie de extensiones (p.e. ".txt, .vbs").

MailScanner procesa cada mensaje que se recibe en el servidor antes de colocarlo en el fichero correspondiente al buzón del usuario de correo. Si encontrara cualquier tipo de virus, eliminaría el fichero adjunto y daría aviso al emisor, al destinatario y al postmaster.

Otra de sus funciones básicas es el evitar los ataques de tipo DoS. Una herramienta muy útil para Administradores de servidores de Internet bajo Linux.

2.11. VIRUS

Los virus son programas de computadora, que cuando se ejecutan, se propagan por si mismos a otros programas o archivos, no infectados. Tienen como objetivo causar alteraciones en un sistema de cómputo. Pueden causar desde una simple broma, hasta la pérdida total de programas, datos y algunos llegan a formatear el disco duro o a comprometer la integridad del sistema de cómputo.

Un virus típico ejecuta 2 funciones:

- Se copia a si mismo a un programa no infectado.
- Ejecuta cualquier instrucción que el autor incluyó en él

2.11.1. CLASIFICACIÓN DE LOS VIRUS

2.11.1.1. Caballos de Troya

Los caballos de troya no llegan a ser realmente virus porque no tienen la capacidad de autoreproducirse. Se esconden dentro del código de archivos ejecutables y no ejecutables pasando inadvertidos por los controles de muchos antivirus. Existen diferentes caballos de troya que se centrarán en distintos puntos de ataque. Su objetivo será el de robar las contraseñas que el usuario tenga en sus archivos o las contraseñas para el acceso a redes, incluyendo a Internet. Muchos caballos de troya utilizados para espionaje industrial están programados para autodestruirse una vez que cumplan el objetivo para el que fueron programados, destruyendo toda la evidencia.

2.11.1.2. Camaleones

Son una variedad de similar a los Caballos de Troya, pero actúan como otros programas comerciales, en los que el usuario confía, mientras que en realidad están haciendo algún tipo de daño. Cuando están correctamente programados, los

camaleones pueden realizar todas las funciones de los programas legítimos a los que sustituyen (actúan como programas de demostración de productos, los cuales son simulaciones de programas reales). Un software camaleón podría, por ejemplo, emular un programa de acceso a sistemas remotos (rlogin, telnet) realizando todas las acciones que ellos realizan, pero como tarea adicional (y oculta a los usuarios) va almacenando en algún archivo los diferentes logins y passwords para que posteriormente puedan ser recuperados y utilizados ilegalmente por el creador del virus camaleón

2.11.1.3. Virus polimorfos o mutantes

Los virus polimorfos poseen la capacidad de encriptar el cuerpo del virus para que no pueda ser detectado fácilmente por un antivirus. Solo deja disponibles unas cuantas rutinas que se encargaran de desencriptar el virus para poder propagarse. Los métodos básicos de detección no pueden dar con este tipo de virus. Muchas veces para virus polimorfos particulares existen programas que se dedican especialmente a localizarlos y eliminarlos.

2.11.1.4. Virus sigiloso o stealth

El virus sigiloso posee un módulo de defensa bastante sofisticado. Este intentará permanecer oculto tapando todas las modificaciones que haga y observando cómo el sistema operativo trabaja con los archivos y con el sector de booteo. La técnica

stealth de ocultamiento de tamaño captura las interrupciones del sistema operativo que solicitan ver los atributos del archivo y, el virus le devuelve la información que poseía el archivo antes de ser infectado y no las reales.

2.11.1.5. Virus lentos

Los virus de tipo lento hacen honor a su nombre infectando solamente los archivos que el usuario hace ejecutar por el SO, simplemente siguen la corriente y aprovechan cada una de las cosas que se ejecutan

2.11.1.6. Retro-virus o Virus antivirus

Un retro-virus intenta como método de defensa atacar directamente al programa antivirus incluido en la computadora. Generalmente los retro-virus buscan el archivo de definición de virus y lo eliminan, imposibilitando al antivirus la identificación de sus enemigos.

2.11.1.7. Virus multipartitos

Atacan a los sectores de arranque y a los ficheros ejecutables. Su nombre está dado porque infectan las computadoras de varias formas. No se limitan a infectar un tipo de archivo ni una zona de la unidad de disco rígido.

2.11.1.8. Virus voraces

Alteran el contenido de los archivos de forma indiscriminada. Generalmente uno de estos virus sustituirá el programa ejecutable por su propio código. Son muy peligrosos porque se dedican a destruir completamente los datos que puedan encontrar.

2.11.1.9. Bombas de tiempo

Son virus convencionales y pueden tener una o más de las características de los demás tipos de virus pero la diferencia está dada por el trigger de su módulo de ataque que se disparará en una fecha determinada.

2.11.1.10. Conejo

El programa se coloca en la cola de espera y cuando llega su turno se ejecutaba haciendo una copia de sí mismo, agregándolo también en la cola de espera. Los procesos a ser ejecutados van multiplicándose hasta consumir toda la memoria de la computadora central interrumpiendo todos los procesamientos.

2.11.1.11. Macro-virus

Los macro-virus representan una de las amenazas más importantes para una red. Actualmente son los virus que más se están extendiendo a través de Internet. Los macro-virus son pequeños programas escritos en el lenguaje propio (conocido como lenguaje script o macro-lenguaje) propio de un programa. Este tipo de virus alteran de tal forma la información de los documentos infectados que su recuperación resulta imposible.

2.11.1.12. Gusanos

Tienen la capacidad de desparramar un segmento de él o su propio cuerpo a otras computadoras conectadas a una red.

Hay dos tipos de gusanos:

- **Host Computer Word**. Son contenidos totalmente en una computadora, se ejecutan y se copian a si mismo vía conexión de una red. Los Host Computer Worm, originalmente terminan cuando hicieron una copia de ellos mismos en otro host. Entonces, solo hay una copia del gusano corriendo en algún lugar de una red. También existen los Host Computer Worm, que hacen una copia de ellos mismos e infectan otras redes, es decir, que cada máquina guarda una copia de este Gusano.

- **Network Worms**. Consisten en un conjunto de partes (llamadas segmentos), cada una corre en una maquina distinta y usando la red para distintos propósitos de comunicación. Los Network Worm tienen un segmento principal que coordina el trabajo de los otros segmentos, llamados también octopuses.

2.11.2. ANTIVIRUS

2.11.2.1. AMAVISD-NEW

Interfaz de alto rendimiento y fiabilidad entre el MTA y uno o más filtros de contenidos: antivirus o el módulo Mail::SpamAssassin de Perl. Está escrito en Perl, asegurando alta fiabilidad, portabilidad y facilidad de mantenimiento. Se comunica con el MTA vía (E)SMTP o LMTP, o mediante el uso de otros programas. No existen problemas de sincronización en su diseño que pudieran causar pérdidas de correos.

Normalmente se posiciona dentro o cerca del gestor de correo principal, no necesariamente donde se ubiquen las cuentas de correo de los usuarios. Si se está buscando una solución que soporte configuración por usuario y ratios de mensajes pequeños que se ubique al final del proceso de envío (llamado desde procmail o en sustitución de un agente local de envío), posiblemente puedan encontrarse otras soluciones más apropiadas.

Cuando está habilitado el uso de Mail::SpamAssassin (SA), se llama a SA una sola vez por mensaje (independientemente del número de destinatarios). Amavisd-new se beneficia del uso del módulo de Perl Net::Server, el cuál ofrece un rápido entorno multihilo. Esto lo hace adecuado para múltiples analizadores de virus y de correo publicitario en plataformas de correo donde la fiabilidad y el cumplimiento de los estándares son importantes.

2.11.2.2. CLAM ANTIVIRUS

ClamAV es una herramienta antivirus GPL para UNIX. El propósito principal de este software es la integración con los servidores de correo (escaneo de datos adjuntos). El paquete proporciona un servicio multihilo flexible y escalable, un analizador de línea de comandos y una utilidad para la actualización automática vía Internet. Los programas están basados en una librería distribuida con el paquete Clam AntiVirus, la cual puede ser usada por su propio software. Y lo más importante, la base de datos se mantiene actualizada constantemente.

Otras características destacables son el soporte de firmas digitales en la actualización de la base de datos, el análisis durante el acceso bajo Linux y FreeBSD, la detección de más de 20000 virus, gusanos y troyanos, el soporte integrado para archivos comprimidos con Rar, Zip, Gzip y Bzip2 y formatos de correo Mbox, Maildir y ficheros crudos de correo.

2.11.2.2.1 Características

- Escanea archivos y ficheros comprimidos Zip, Rar (2.0), Tar, Gzip, Bzip2, MS OLE2, archivos MS Cabinet, MS CHM, formatos de compresión MS SZDD, UPX (todas las versiones), FSG (1.3, 1.31, 1.33), Petite (2.x)
- Archivos de correo
- Detecta más de 25000 virus, gusanos, y troyanos, incluyendo macro virus de Microsoft Office y MacOffice
- Soporta sistemas Linux y FreeBSD
- Actualización avanzada de la base de datos con soporte para firmas digitales y DNS.

2.11.2.2.2 Requerimientos

- Paquetes zlib y zlib-devel
- compilador gcc 2.9x o superior

2.11.2.3. F-PROT ANTIVIRUS FOR LINUX MAIL SERVERS

Es una protección antivirus de alta velocidad que escanea mensajes de correo electrónico y archivos adjuntos y detecta, desinfecta, borra programas maliciosos, como mass-mailers, gusanos, virus de macros y caballos de Troya.

Soporta los más populares servidores de correo en Linux incluyendo Sendmail, Postfix y Qmail.

2.11.2.3.1 Características

- Escanea más de 123000 virus conocidos y sus variantes.
- Remueve los virus en forma segura sin dañar el archivo original.
- Escanea todos los sistemas de archivos montados, directorios y archivos específicos.
- Escanea archivos y archivos empaquetados.
- Incluye actualización automática de la base de datos de virus.
- Puede ser configurado con la utilidad cron.
- Escanea correo electrónico en tránsito con los tres sistemas del correo electrónico ampliamente usados: Sendmail, Postfix, y Qmail.

2.11.2.4. SOPHOS ANTIVIRUS

Proporciona protección antivirus integrada multiplataforma, detecta y desinfecta virus en servidores, estaciones y portátiles. Ofrece escaneado en acceso, programado y en demanda. Su arquitectura única permite determinar de forma inteligente los archivos a escanear, maximizando la transparencia ante el usuario y minimizando el gasto de recursos. Sophos Antivirus incorpora potentes herramientas administrativas que permiten la instalación, actualización, configuración y notificación centralizadas.

2.11.2.4.1 Características

- Monitoriza todos los posibles puntos de entrada de virus, incluyendo disquetes, programas, documentos, unidades de red, CD-ROM, así como correo electrónico y descargas de Internet.
- Detecta y desinfecta virus de macro, sector de arranque y ciertos archivos ejecutables.
- Detecta virus en archivos comprimidos, incluyendo compresión recursiva.
- Detecta virus polimórficos gracias al avanzado lenguaje de descripción de virus (VDL) de Sophos y al emulador de código integrado.
- Bajo consumo de recursos al escanear gracias al sistema de reconocimiento inteligente de tipos de archivo.
- Permite el escaneado programado automático de los diferentes recursos.
- Dispone de modo inmediato para el escaneado en demanda.
- Ofrece notificación centralizada automática de cada incidente vírico.
- Puede instalarse de forma automática en múltiples estaciones.
- Permite la actualización y administración remotas a través de la red con SAVAdmin.
- Permite la actualización constante y automática con descargas desde Internet mediante Enterprise Manager.
- Permite la actualización remota de portátiles a través de Internet desde la Web de la compañía.

CAPÍTULO III. CONFIGURACIONES

3.1. CONFIGURACIÓN DE SERVIDOR DE CORREO

3.1.1. Verificar los parámetros de red.

Debe definirse el nombre de la máquina que funcionará como servidor de correo. Normalmente se utiliza el esquema host.dominio. Así que se debe asegurar de que esto se encuentra perfectamente definido en /etc./sysconfig/network y /etc./hosts

/etc/sysconfig/network

NETWORKING=yes
HOSTNAME=srvFedora.gguerrero.com
GATEWAY=100.100.100.1

/etc/hosts

127.0.0.1 localhost.localdomain localhost
100.100.100.1 srvFedora.gguerrero.com srvFedora gguerrero.com

/etc./resolv.conf

search gguerrero.com
nameserver 100.100.100.1

3.1.2. Configuración de DNS

Si el servidor DNS se localiza en otro servidor y es administrado por otras personas, solo bastará con informar al administrador de dicho servidor de nombres la existencia del nuevo servidor de correo electrónico, a fin de que se dé de alta la entrada correspondiente en el DNS y a su vez a fin de que el NIC lo tome en cuenta en el siguiente ciclo de refresco.

Si se desea configurar DNS propio, y dar éste de alta con el NIC, se necesitará tener instalados los siguientes paquetes: bind, bind-utils y caching-nameserver.

Editar el fichero named.conf en RedHat 7.x, RedHat 8, RedHat 9, Fedora Core 1 en el directorio /etc., en Fedora Core 2 y Fedora Core 3 en /var/named/chroot/etc. Adicionar la zona directa y la zona inversa del DNS que se necesite.

Si se trata de un servidor de nombres de dominio para uso exclusivo en red local, y se quiere evitar problemas de seguridad de diferente índole, puede utilizarse el parámetro allow-query, el cual servirá para especificar que solo ciertas direcciones podrán realizar consultas al servidor de nombres de dominio. Se pueden especificar directamente direcciones IP, redes completas o listas de control de acceso que deberán definirse antes.

acl "redlocal" { 100.100.100.0/24; 100.100.102.0/24; }

La declaración options toma la siguiente forma:

```
options {
   option ;
   [ option ; ...]
};
```

En esta declaración, las directivas option son reemplazadas con una opción válida. Las siguientes son opciones usadas a menudo:

- **allow-query**. Especifica que hosts tienen permitido consultar el servidor de nombres. Por defecto, todos los hosts tienen derecho a consultar. Una lista de control de acceso, o una colección de direcciones IP o redes se puede usar aquí para sólo permitir a hosts particulares hacer consultas al servidor de nombres.
- **allow-recursion**. Parecida a la opción allow-query, salvo que se aplica a las peticiones recursivas. Por defecto, todos los hosts están autorizados a presentar peticiones recursivas en un servidor de nombres.
- **blackhole**. Indica que hosts no tienen permitido consultar al servidor de nombres.
- **directory**. Reemplaza el directorio de trabajo named en vez del directorio predeterminado /var/named.
- **forward.** Controla el comportamiento de reenvío de una directiva forwarders.

```
options {
   # directorio donde se copiaran las zonas DNS
   directory "/var/named";
   # no permitir la transferencias de zonas a nadie
   allow-transfer { any; };
   # tamaño del buffer de datos del DNS
   datasize 20M;
   # escuchando por las direcciones, localhost. más otros IP's
   listen-on { 127.0.0.1; 100.100.100.1;; };
   # a través de qué interfaces se permitirán solicitudes
   allow-query { redlocal; localhost; };
   # a través de qué interfaces se escucharán solicitudes recursivas
   allow-recursion { localhost; };
   # tamaño de la caché del DNS
   max-cache-size 20M;
   # a través de que puerto escuchará solicitudes
   query-source address * port 53;
   transfer-source * port 53;
   notify-source * port 53;
   zone-statistics yes;
   # Donde se guardarán las estadisticas
   statistics-file "named.stats";
};
```

Interfaces de control

```
controls {
   # Interfaz donde se escucha las actualizaciones de zona
   inet 127.0.0.1 port 953 allow { 127.0.0.1; } keys { rndckey; };
   # a través de estas interfaces se escucha cualquier otro tipo de solicitud:
   inet 100.100.100.1 allow { any; };
};
```

Zona Directa de gguerrero.com

```
zone "gguerrero.com" IN {
        allow-update { none; };
        type master;
        file "/var/named/gguerrero.com.zone";
```

```
};
```

Zona Inversa de la interfaz local 100.100.100.in-addr.arpa

```
zone "100.100.100.in-addr.arpa"  IN {
        allow-update { none; };
        type master;
        file "/var/named/100.100.100.zone";
};
```

Crear o editar los archivos que se hace referencia en named.conf. /var/named/gguerrero.com.zone para RedHat 7.x, RedHat 8, RedHat 9. /var/named/chroot/var/mened/gguerrero.com.zone para Fedora Core x

```
$TTL 86400
@       IN      SOA     gguerrero.com.  gguerrero.gguerrero.com. (
                        2004071302 ; serial
                        28800 ; refresh
                        14400 ; retry
                        3600000 ; expire
                        86400 ; ttl
                        )
        IN      NS      gguerrero.com.
        IN      MX      5       gguerrero.com.
srvFedora               IN      A       100.100.100.1
gguerrero.com.          IN      A       100.100.100.1
mail            IN      CNAME           srvFedora
www             IN      CNAME           srvFedora
```

No se deben implementar registros del estilo:

```
*.gguerrero.com.        IN      MX      10      mail.gguerero.com.
```

Sólo se deben reflejar direcciones susceptibles a recibir de correo. Estas entradas son invalidadas en el caso de existir registros de tipo Address (A) intentando establecerse conexiones a través del puerto 25.

/var/named/100.100.100.zone para RedHat 7.x, RedHat 8, RedHat 9. /var/named/100.100.100.zone para Fedora Core x

```
$TTL 86400
@       IN      SOA     gguerrero.com.  gguerrero.gguerrero.com. (
                        2004071303 ; serial
                        28800 ; refresh
                        14400 ; retry
                        3600000 ; expire
                        86400 ; ttl
                        )
        IN      NS      gguerrero.com.
1       IN      PTR     gguerrero.com.
```

Al terminar de editar todos los ficheros involucrados, iniciar el servicio.

/sbin/service named start

3.1.3. Configuración de Sendmail

Editar el fichero /etc./mail/local-host-names, en él enumerar todos y cada uno de los aliases que tenga el servidor que se este configurando, así como los posibles dominios.

gguerrero.com
srvFedora.gguerrero.com
mail.gguerrero.com

Es recomendable realizar una copia del fichero /etc./mail/sendmail.mc y luego editarlo de la siguiente manera:

a. Deshabilitar las funciones que definen trabajar sobre la interfaz 127.0.0.1 y recibir correo de dominios inexistentes precediendo con 'dnl ' en las siguientes líneas:

dnl *DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')dnl*

b. Para filtrar spam de manera eficiente, la mejor manera de empezar a hacerlo es rechazando correo proveniente de dominios no resueltos, es decir dominios que no están registrados en un DNS y que por lo tanto son inválidos. Para tal fin, a menos que se requiera lo contrario, es necesario mantener comentada la siguiente línea:

dnl *FEATURE(`accept_unresolvable_domains')dnl*

c. Habilitar las siguientes líneas y adapte valores para definir la máscara que utilizará el servidor:

MASQUERADE_AS(`gguerrero.com')dnl
FEATURE(masquerade_envelope)dnl
FEATURE(masquerade_entire_domain)dnl

d. Si se desea cargar listas negras para mitigar el spam, puede añadirse las siguientes líneas justo arriba de MAILER(smtp)dnl:

FEATURE(dnsbl, `blackholes.mail-abuse.org', `Rejected - see www.mail-abuse.org/rbl/')dnl
FEATURE(dnsbl, `dialups.mail-abuse.org', `Rejected - see www.mail-abuse.org/dul/')dnl
FEATURE(dnsbl, `relays.mail-abuse.org', `Rejected - see work-rss.mail-abuse.org/rss/')dnl
FEATURE(dnsbl, `sbl-xbl.spamhaus.org', `Rejected - see http://www.spamhaus.org/SBL/')dnl
FEATURE(dnsbl, `bl.spamcop.net', `Rejected - see http://spamcop.net/')dnl

Debido a la naturaleza del correo electrónico, es posible para un atacante inundar fácilmente el servidor y desencadenar en una denegación de servicio. Fenómenos como el Spam no hacen las cosas más fáciles y la administración de un servidor de correo puede tornarse una pesadilla. Añadir opciones avanzadas de seguridad se convierte en algo indispensable.

confMAX_RCPTS_PER_MESSAGE. Este parámetro sirve para establecer un número máximo de destinatarios para un mensaje de correo electrónico. De modo predeterminado sendmail establece un máximo de 256 destinatarios.

define(`confMAX_RCPTS_PER_MESSAGE', `20')dnl

confBAD_RCPT_THROTTLE. Sirve para establecer el tiempo de letargo que se utilizará por cada destinatario que sobrepase el límite establecido por confMAX_RCPTS_PER_MESSAGE. De modo predeterminado Sendmail no establece tiempo de letargo.

define(`confBAD_RCPT_THROTTLE', `2')dnl

confPRIVACY_FLAGS. Cuando se establece como valor `goaway', se deshabilitan varios comandos SMTP como EXPN y VRFY, los cuales pudieran ser utilizados para revelar los nombres de usuarios locales a un spammer. También deshabilita las notificaciones de entrega, el cual es un mecanismo comúnmente utilizado por los spammers para verificar la existencia de una cuenta, y hace que el sistema solicite obligatoriamente HELO o EHLO antes de utilizar el comando MAIL. Muchos programas utilizados para enviar correo masivo no solicitado ni siquiera se molestan en utilizar HELO o EHLO. Por defecto los valores de confPRIVACY_FLAGS son `authwarnings,novrfy,noexpn,restrictqrun', cambie por lo siguiente:

define(`confPRIVACY_FLAGS',`goaway')dnl

confMAX_HEADERS_LENGTH. Esté parámetro se utiliza para definir el tamaño máximo permitido para la cabecera de un mensaje en bytes. Algunos programas utilizados para enviar spam tratan de impedir que los MTA puedan registrar transacciones generando cabeceras muy grandes. Limitar el tamaño de las cabeceras hace más difícil la ejecución de guión que explote vulnerabilidades recientes (desbordamientos de búfer) en UW IMAP, Outlook y Outlook Express. La mayor parte de los mensajes de correo electrónico tendrán cabeceras de menos de 2 Kb (2048 bytes).

define(`confMAX_HEADERS_LENGTH', `5128')dnl

confMAX_MESSAGE_SIZE. Se utiliza para especificar el tamaño máximo permitido para un mensaje de correo electrónico en bytes. Puede especificarse lo que el administrador considera apropiado.

define(`confMAX_MESSAGE_SIZE', `3072000')dnl

confMAX_DAEMON_CHILDREN. Este parámetro sirve para especificar cuantos procesos hijos se permitirán simultáneamente en el servidor de correo. De modo predeterminado Sendmail no establece límites para este parámetro. Si se sobre pasa el límite de conexiones simultáneas, el resto serán demoradas hasta que se terminen las conexiones existentes y dejen lugar para nuevas conexiones.

define(`confMAX_DAEMON_CHILDREN', `5')dnl

confCONNECTION_RATE_THROTTLE. Establece el número de conexiones máximas por segundo. De modo predeterminado sendmail no establece límites para este parámetro.

define(`confCONNECTION_RATE_THROTTLE', `4')dnl

confSMTP_LOGIN_MSG. Este parámetro permite establecer el mensaje de bienvenida al establecer la conexión al servidor. Es posible ocultar el nombre y al versión de sendmail, esto con el objeto de agregar seguridad por secreto. Funciona simplemente haciendo que quien se conecte hacia el servidor no pueda saber que software y versión del mismo se está utilizando y con ellos dificultar a un delincuente o abusador de servicio el determinar que vulnerabilidad específica explotar.

define(`confSMTP_LOGIN_MSG',`$j ; $b')dnl

e. Generar /etc/mail/sendmail.cf:

m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf

f. Habilitar los servicios ipop3 (POP3 tradicional, autenticación en texto plano), pop3s (POP3 seguro, autenticación con criptografía), imap (IMAP

tradicional, autenticación en texto plano) e imaps (IMAP seguro, autenticación con criptografía). Utilizar los más apropiados para la red local de acuerdo a las capacidades de los clientes de correo electrónico utilizados. Tomar en cuenta que la autenticación por medio de texto plano es definitivamente un método inseguro, y siempre serán mejor usar los servicios que permitan establecer conexiones seguras:

- Fedora core 1, RedHat 9, RedHat 8, RedHat 7.x

```
/sbin/chkconfig imap on
/sbin/chkconfig ipop3 on
```

También puede habilitarlos manualmente con un editor de texto, lo cual es sugerido a fin de habilitar opciones adicionales, como direcciones IP específicas a las cuales se les estaría permitido cierto servicio. Acceder a al directorio /etc/xinet.d/ y edite los fichero ipop3, pop3s, imap e imaps, según se requiera.

```
service pop3
{
   socket_type = stream
   wait = no
   user = root
   server = /usr/sbin/ipop3d
   log_on_success += USERID
   log_on_failure += USERID
   disable = no
   only_from = 100.100.100.1 100.100.1002  localhost
}
```

Lo mismo se aplica para el protocolo IMAP e IMAPS.

- Fedora Core 2, Fedora Core 3 es necesario editar el archivo /etc./dovecot.conf para habilitar los protocolos imap, pop, imaps y/o pop3s así como la ubicación de los ejecutables.

protocols = imap imaps pop3 pop3s
imap_executable = /usr/libexec/dovecot/imap
pop3_executable = /usr/libexec/dovecot/pop3

Habilitar dovecot para que inicie con el sistema.

/sbin/chkconfig dovecot on

- Definir los dominios para los cuales se estará permitiendo enviar correo electrónico. Esto se hace generando el fichero /etc./mail/relay-domains:

srvFedora.gguerrero.com
gguerrero.com

- Abrir el fichero /etc./mail/access y agregar algunas líneas para definir quienes podrán hacer uso del servidor de correo para poder enviar mensajes y agregar las direcciones de correo electrónico de aquellos a quienes se considere indeseables, o se quiera bloquear.

Para aceptar que un usuario, una máquina, o un conjunto de máquinas, puedan enviar mensajes para el servidor, independiente de otras reglas definidas, adicionar host OK.

gguerrero.com *OK*

Para aceptar mensajes de un usuario, una máquina o un conjunto de máquinas para reenvío, adicionar host RELAY.

100.100.100.1 *RELAY*
100.100.100 *RELAY*

Para evitar que un usuario, una máquina o un conjunto de máquinas puedan enviar mensajes para este servidor, adicionar host REJECT.

1ppmgfh@ohta-hp.or.jp *REJECT*
12ak12@hotmail.com *REJECT*
1234d3dw@Flashmail.com *REJECT*

Para descartar mensajes de un usuario, una máquina o un conjunto de máquinas, adicionar host DISCARD.

bangou@letrera.net *DISCARD*
barros@sprintmail.com *DISCARD*

Compilar este archivo para generar otro en formato de base de datos a fin de ser utilizado por Sendmail.

cd /etc./mail
make

O bien ejecutar lo siguiente:

makemap hash /etc/mail/access.db < /etc/mail/access

- Será de utilidad designar un alias a la cuenta de correo de root a fin de recibir los mensajes generados por el sistema en una cuenta común de usuario. Abrir el archivo /etc./aliases, en donde al final se encuentra la siguientes líneas:

#root: *jperez*

Descomentar esta línea y asignar el nombre de la cuenta de usuario que se utiliza normalmente.

root: *gguerrero*

A fin de que este nuevo alias surta efecto y pueda ser utilizado por Sendmail utilizar el comando newaliases

/sbin/newaliases

- Reiniciar el servicio de Sendmail

/sbin/service sendmail restart

Para verificar si el servicio de correo esta funcionando correctamente se envía un correo a un usuario del sistema con el comando mail.

 # mail gguerrrero
Subject: hola
este es un correo enviado con el servidor local

Luego verificar si el correo fue recibido en forma correcta ejecutando el comando mail. El resultado debe ser parecido al siguiente:

From root@srvFedora.gguerrero.com Thu Feb 17 00:47:02 2005
Date: Thu, 17 Feb 2005 00:46:33 -0500
From: root <root@srvFedora.gguerrero.com>
To: gguerrero@srvFedora.gguerrero.com
Subject: hola
X-MailScanner: Found to be clean
X-Spam-Checker-Version: SpamAssassin 3.0.0 (2004-09-13) on
srvFedora.gguerrero.com
X-Spam-Level:
X-Spam-Status: No, score=-2.8 required=5.0 tests=ALL_TRUSTED,AWL
autolearn=ham version=3.0.0

este es un correo enviado con el servidor local

Para configurar smtpauth con sendmail de debe realiazar lo siguiente:

a. Configurar PAM para utilizar los mecanismos de validación. No es complicado, simplmente editar fichero llamado /etc./pam.d/smtp con el siguiente contenido:

auth required /lib/security/pam_stack.so service=system-auth
account required /lib/security/pam_stack.so service=system-auth

b. Editar el fichero /usr/lib/sasl2/Sendmail.conf

pwcheck_method:saslauthd

c. Editar /etc./mail/sendmail.mc para que acepte SASL

define(`confAUTH_OPTIONS', `A')dnl
TRUST_AUTH_MECH(`EXTERNAL DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
define(`confAUTH_MECHANISMS', `EXTERNAL GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl

d. Regenerar sendmail.cf y reiniciar el servicio sendmail

3.2. CONFIGURACIÓN DE FETCHMAIL

Fetchmail puede ser utilizado sin archivo de configuración, simplemente con los parámetros que le demos en la línea de comandos. Pero lo más útil es indicar las

opciones de fetchmail en su fichero de configuración, un fichero llamado .fetchmailrc.

```
defaults                # Comandos comunes a todos los servidores
        fetchall        # - Recoge todos los mensajes en el buzón
        flush           # - Borra todos los mensajes ya recogidos en
                        #anteriores llamadas al servidor
```

Configurar las cuentas de correo que se desea depositar en buzón de correo local. Se puede indicar el protocolo que utilizará la cuenta de correo o dejar que fetchmail busque el protocolo apropiado para que lea el correo.

```
poll pop.mail.yahoo.com with proto POP3
user login_remoto_s with password "contraseña" is gguerrero here

poll www.mail.com with
user login_remoto with password "contraseña" is gguerrero here

poll mail.lycos.es with
user login_remoto is gguerrero here
```

3.3. CONFIGURACIÓN PROCMAIL

Para empezar a usar procmail para filtrar el correo crear un archivo /etc./procmailrc.

La sintaxis básica de una regla es:

```
:0:
* condicion_1
* condicion_2
...
acción
```

a. Esta regla enviará todo el correo recibido a la dirección especificada.

```
:0
! guillermo@gguerrero.com
```

b. Esta regla es para enviar las copias de los mensajes a otra dirección.

```
:0c
! guillermo@gguerrero.com
```

c. Guardar los mensajes de una lista en una carpeta

```
:0:
* ^TO_lista@gguerrero.com
lista_mails

:0:
* ^Subject: lista:
lista_mails
```

d. Borrar ciertos mensajes que contenga ciertas palabras

```
:0
* ^Subject: .*Viagra
/dev/null
```

e. Descartar el correo spam, guardandolo en el fichero /var/log/spam_log

```
:0fw
| spamc
:0
* ^X-Spam-Flag: YES
      /var/log/spam_log
```

3.4. CONFIGURACIÓN CLAM ANTIVIRUS

Si se va ha instalar ClamAV por primera vez para, se necesita agregar un nuevo grupo y usuario

```
groupadd clamav
# useradd -g clamav -s /bin/false -c "Clam AntiVirus" clamav
```

La opción –s /bin/false indica que no se podrá iniciar una sesión con el usuario clamav.

Una vez creado el grupo y el usuario clamav, desempaquetar el archivo:

```
zcat clamav-0.80.tar.gz | tar xvf -
cd clamav-0.80
```

Para instalar los archivos de la configuración en /etc, configurar el paquete como sigue:

./configure --sysconfdir=/etc.

Para habilitar clamav-milter se requiere libmilter y sus archivos de desarrollo. Configurar clamav con:

./configure --enable-milter

Compilar e instalar clamav. Para realizar esta tarea es necesario el compilador gcc

make
make install

Terminada la ejecución de los comandos es necesario editar el archivo /etc./clamd.conf. Comentar la línea “Example” adicionando el carácter “#” al inicio de la misma.

clamav-milter es un escáner de correo electrónico muy rápido diseñado para Sendmail. Para conectarlo con Sendmail agregar las siguientes líneas a /etc./mail/sendmail.mc:

INPUT_MAIL_FILTER('clmilter','S=local:/var/run/clmilter.sock,F=, T=S:4m;R:4m')dnl
define('confINPUT_MAIL_FILTERS', 'clmilter')

Regenerar el archivo /etc./mail/sendmail.cf y reinizar el servicio de sendmail.

Para las pruebas escanear el directorio de las fuentes guardando un log en el archivo scan.txt, esta ejecución tiene que detectar 1 archivo infectado con el virus test eicar (no es un virus de verdad, solo se usa para probar los antivirus).

clamscan −r −l scan.txt clam−0.80

El programa encargado de las actualizaciones es el freshclam. Este programa puede ser ejecutado de 3 formas diferentes y básicamente lo que hace es conectarse a 2 servidores diferentes (por motivos de seguridad) para comprobar si ha cambiado la base de datos de virus, y en caso afirmativo descargarla, comprobar su firma e instalarla.

- **Modo interactivo**. Siendo root desde una consola ejecutamos freshclam.

  # freshclam

- **Como demonio.** Para ejecutar freshclam como demonio lo único que tenemos que añadir es el parámetro −d seguido del parámetro −c X, siendo X el número de comprobaciones al día que queramos que haga.

Evidentemente para que esto funcione bien debería estar en los scripts de inicio.

 # freshclam -d

- **daemon del cron.** Agregar la línea siguiente al el crontab del root o usuarios del clamav.

*N * * * * /usr/local/bin/freshclam --quiet*

3.5. CONFIGURACIÓN MAILSCANNER

Editar /etc./mailscanner/MailScanner.conf a fin configurar los siguientes parámetros:

a. Definir antivirus a utilizar, se puede utilizar más de un antivirus. Solo se necesita instalar las versiones apropiadas para el sistema operativo que se utiliza y añadir como lista separada por ",".

Virus Scanner = sophos, f-prot, clamav

Si se quiere usar los múltiples antivirus, separar los comandos con "," en el mismo orden de la lista de antivirus.

Sweep = /usr/local/Sophos/bin/sophoswrapper, /usr/local/f-prot/f-protwrapper, /usr/local/bin/clamscan

b. Control de Spam

Spam Checks = yes
Spam Header = X-MailScanner-SpamCheck:
Spam Modify Subject = yes
Spam Subject Text = {SPAM?}
Use SpamAssassin = yes

c. Editar el fichero /etc./mail/spamassassin/local.cf.

- **required_hits.** Puntos son necesarios para considerar a un correo spam.
- **rewrite_subject.** Subject del mail si se detecta spam.
- **subject_tag.** Subject que se añade al asunto del mensaje si se marca como spam.
- **report_safe.** Adjuntar el mensaje marcado como SPAM para evitar posibles mensajes malévolos.
- **hitelist_from.** Para decir que una dirección de correo no es SPAM.

required_hits 5.0
rewrite_subject 1
*subject_tag *** ES UN SPAM ****
report_safe 1
use_terse_report 0
use_bayes 1
skip_rbl_checks 0
use_razor2 1
use_dcc 1

use_pyzor 1
ok_languages es
whitelist_from ofertas@push.infojobs.net

3.6. SQUIRRELMAIL

Squirrelmail viene incluido en los cds de instalación de la mayoría de las distribuciones Linux. Para instalarlo ejecutar los siguientes comandos:

apt-get install squirrelmail

ó

yum squirrelmail

ó

rmp –i squirrelmail.x.x.rpm

La configuración de squirrelmail se la puede realizar en forma directa en el archivo /etc/squirrelmail/config.php o ejecutando el comando conf.pl.

Editar el archivo de /etc/squirrelmail/config.php:

```
global $version;
$config_version = '1.4.0';
$config_use_color = 2;
$org_name      = "Gguerrero";
$org_logo      = SM_PATH . 'images/sm_logo.png';
$org_logo_width  = '308';
$org_logo_height = '111';
$org_title     = "SquirrelMail $version";
```

```
$signout_page  = '';
$frame_top     = '_top';
$provider_uri     = 'http://mail.gguerrero.com/';
$provider_name     = 'gguerrero.com';
$motd = "Bienvenido a su cuenta de correo";
$squirrelmail_default_language = 'es_ES';
$domain                 = 'gguerrero.com';
$imapServerAddress      = 'gguerrero.com';
$imapPort               = 143;
$useSendmail            = true;
$smtpServerAddress      = 'gguerrero.com';
$smtpPort               = 25;
$sendmail_path          = '/usr/sbin/sendmail';
$pop_before_smtp        = false;
$imap_server_type       = 'uw';
$invert_time            = false;
$optional_delimiter     = '/';
$default_folder_prefix          = 'mail/';
$trash_folder                   = 'Trash';
$sent_folder                    = 'Sent';
$draft_folder                   = 'Drafts';
```

Con el comando ./conf.pl, en algunas distribuciones el comando es ./configure que se encuentra en el directorio donde se instalo squirrelmail. En el menú escoger que configurar:

```
SquirrelMail Configuration : Read: config.php (1.4.0)
---------------------------------------------------------
Main Menu --
1.  Organization Preferences
2.  Server Settings
3.  Folder Defaults
4.  General Options
5.  Themes
6.  Address Books (LDAP)
7.  Message of the Day (MOTD)
8.  Plugins
9.  Database

D.  Set pre-defined settings for specific IMAP servers
```

C. Turn color on
S Save data
Q Quit

Command >>

La opción 1 permite configurar el nombre de la organización, la imagen que se desea que muestre, el idioma por defecto a utilizar.

SquirrelMail Configuration : Read: config.php (1.4.0)

Organization Preferences
1. Organization Name : Gguerrero
2. Organization Logo : ../images/sm_logo.png
3. Org. Logo Width/Height : (308/111)
4. Organization Title : SquirrelMail $version
5. Signout Page :
6. Default Language : es_ES
7. Top Frame : _top
8. Provider link : http://mail.gguerrero.com/
9. Provider name : gguerrero.com

R Return to Main Menu
C. Turn color on
S Save data
Q Quit

Command >>

Presionar R para regresar al menú anterior y escojer la opción 2 para realizar la configuración del servidor de correo, tales como, dominio, el protocolo de transporte.

SquirrelMail Configuration : Read: config.php (1.4.0)

Server Settings

General

----
1. Domain : gguerrero.com
2. Invert Time : false
3. Sendmail or SMTP : Sendmail

A. Update IMAP Settings : gguerrero.com:143 (uw)
B. Change Sendmail Config : /usr/sbin/sendmail

R Return to Main Menu
C. Turn color on
S Save data
Q Quit

Command >>

Reiniciar el servicio httpd.

service httpd restart

Iniciar el navegador web para probar el servicio.

Figura 3.1. Inicio squirrelmail

3.7. MAILMAN

Mailman se puede instalar con los paquetes rpm o con las fuentes del mismo. Para el presente caso se va ha utilizar las fuente de mailman.

Antes de iniciar la instalación de mailman es necesario crear el grupo y usuario mailman.

groupadd mailman
useradd mailman –g mailman

Descomprimir las fuentes de mailman en un directorio

zcat mailman-<version>.tar.gz | tar xvf –

Cambiar al directorio mailman-<versión> y ejecutar el comando ./configure para que el sistema realice las configuraciones necesarias para su instalación.

cd mailman-<version>
./configure
./make
./make install

Después de la ejecución de “make install", se puede verificar que su instalación tiene todos los permisos correctos y propiedades del grupo ejecutando el comando check_perms que se encuentra en la carpeta donde se instalo mailman.

```
cd /usr/local/mailman
bin/check_perms
```

Repitir el paso anterior hasta que ningún error se reporte.

A continuación abrir el archivo de configuración de mailman mm_cfg.py. Este archivo se encuentra en mailman/Mailman.

Modificar las siguientes líneas de acuerdo a sus necesidades:

```
DEFAULT_HOST_NAME       = 'gguerrero.com'
DEFAULT_URL             = 'http://mail.gguerrero.com/mailman/'
IMAGE_LOGOS             = '/icons/'
```

Para probar Mailman, crear una lista llamada test. Esta lista debería ser borrada después de que hacer uso de ella. En caso de problemas durante la prueba, arréglarlos y repitir el procedimiento de prueba desde el principio.

Primero ejecutar el proceso /usr/local/mailman/bin/newlist y responder como sigue a estas entradas:

[root@srvFedora bin]# ./newlist test
Indique la dirección de correo de la persona que gestionará la lista:
Clave inicial de test:
Para terminar de crear su lista de distribución
tiene que editar el fichero /etc./aliases (o equivalente)
añadiendo las siguientes líneas y ejecutando posiblemente el
programa `newaliases':

A continuación agregar las líneas especificadas por newlist en el archivo de /etc./aliases y regenerar con newaliases. Estas deberían verse así:

test: "|/usr/local/mailman/mail/mailman post test"
test-admin: "|/usr/local/mailman/mail/mailman admin test"
test-bounces: "|/usr/local/mailman/mail/mailman bounces test"
test-confirm: "|/usr/local/mailman/mail/mailman confirm test"
test-join: "|/usr/local/mailman/mail/mailman join test"
test-leave: "|/usr/local/mailman/mail/mailman leave test"
test-owner: "|/usr/local/mailman/mail/mailman owner test"
test-request: "|/usr/local/mailman/mail/mailman request test"
test-subscribe: "|/usr/local/mailman/mail/mailman subscribe test"
test-unsubscribe: "|/usr/local/mailman/mail/mailman unsubscribe test"

Esperar a que mailman envíe al propietario de la lista un e-mail. Este e-mail empezará con algo como el siguiente texto:

----- start of excerpt -----
The mailing list `test' has just been created for you. The following
is some basic information about your mailing list.

Your mailing list password is:

{password}

You need this password to configure your mailing list. You also need
it to handle administrative requests, such as approving mail if you
choose to run a moderated list.

You can configure your mailing list at the following web page:

http://mail.gguerrero.com/mailman/admin/test

The web page for users of your mailing list is:

http://mail.gguerrero.com/mailman/listinfo/test

You can even customize these web pages from the list configuration page. However, you do need to know HTML to be able to do this.

There is also an email-based interface for users (not administrators) of your list; you can get info about using it by sending a message with just the word `help' as subject or in the body, to:

test-request@gguerrero.com

To unsubscribe a user: from the mailing list 'listinfo' web page, click on or enter the user's email address as if you were that user. Where that user would put in their password to unsubscribe, put in your admin password. You can also use your password to change member's options, including digestification, delivery disabling, etc.

Please address all questions to mailman-admin@gguerrero.com.
----- end of excerpt -----

Después de recibir este e-mail, ir a la página web http://mail.gguerrero.com/mailman/admin/test e iniciar la sesión con la contraseña enviada. Ahora, revisar la sección de Opciones Generales de la página web Administrativa. Agregar algún texto en la sección de descripción para familiarizarse con la forma en que las actualizaciones son hechas. Guardar las actualizaciones y terminar la sesión.

A continuación ingresar a la página web "http://mail.gguerrero.com/mailman/listinfo/test" y revizar la página. Usar esta página para suscribirse a la lista.

Debería en breve recibir un e-mail de solicitud de confirmación. Siguir las instrucciones en este e-mail para confirmar la solicitud de suscripción.

3.8. INN

INN posee un número de parámetros que son de naturaleza global; estos afectan a todos los grupos de noticias que maneja.

a. El archivo inn.conf

El archivo principal de configuración de INN es inn.conf. En medio de otras cosas, éste determina como es conocida la computadora en Usenet. La versión 2.x de INN posee un número desconcertante de parámetros. Afortunadamente, la mayoría de estos tienen valores por defecto, que son razonablemente compatibles para diferentes situaciones.

```
mta:                "/usr/sbin/sendmail -oi -oem %s"
# moderatormailer:      %s@uunet.uu.net
organization:         "Servidor de noticias gguerrero.com"
ovmethod:             tradindexed
hismethod:            hisv6
pathhost:             news.gguerrero.com
pathnews:             /usr/local/news

# General Settings

domain:                     news.gguerrero.com
#innflags:
```

mailcmd: /usr/local/news/bin/innmail
server: srvFedora.gguerrero.com

Server le dice a rnews y a inews cuál es el servidor al que deben contactar para entregar los artículos. Esta entrada es absolutamente crucial; para pasarle artículos a innd, se debe establecer una conexión NNTP con el servidor.

El campo domain debe contener el dominio del servidor que se encuentra completamente calificado. Algunos programas necesitan este dominio; si la librería que resuelve los nombres, solamente retorna nombres no calificados, el nombre dado en el campo domain es derivado hacia ella. No es un problema configurar este modo, pero es mejor definir un dominio en domain.

pathhost, define el nombre del servidor que INN agregará a la cabecera Path: cuando quiera recibir un artículo. El la mayoría de los casos, Ud. querrá utilizar el nombre del dominio de su servidor de noticias; si éste es el caso, puede omitir esta línea ya que por defecto se utiliza este nombre. Ocasionalmente, puede utilizar el nombre genérico, como por ejemplo news.gguerrero.com, para dar servicio a un dominio grande. Haciendo esto, se puede mover el sistema de noticias fácilmente hacia un servidor diferente, cuando se requiera.

La clave organization le permite saber a inews que texto debe ingresar en el campo Organization: de los artículos publicados por los usuarios locales. Formalmente, este es el lugar donde debe ir una descripción de su organización, o el

nombre extendido de la misma. Si no desea ser tan formal, está muy de moda, que las organizaciones con un poco de humor lo expresen aquí.

El campo mta es obligatorio y especifica la ruta de acceso y el nombre del agente de transporte de los mensajes, usado para enviarle mensajes al moderador. %s es reemplazado por la dirección de mail del moderador.

La línea que contiene la entrada moderatormailer define la dirección por defecto que es utilizada cuando un usuario intenta dejar un mensaje en un grupo de noticias que se encuentra moderado. Las direcciones de los moderadores de cada grupo usualmente son guardadas en un archivo por separado, pero toma mucho tiempo seguirles los pasos a todos ellos. La entrada moderatormailer es, por consiguiente, consultada como último recurso

Finalmente, cada una de las entradas restantes, especifica la ubicación de algún componente o archivo perteneciente a INN. Si instaló INN desde los paquetes, estas ubicaciones han sido creadas por usted. Por el contrario, si se decidió compilar el sistema, debe asegurarse que estas entradas reflejen las ubicaciones donde se encuentra INN.

b. Grupos de Noticias

El administrador del sistema de noticias, es capaz de controlar que usuarios tienen acceso a los grupos. INN provee dos archivos de configuración los cuales

dejan al administrador decidir cuáles son los grupos de noticias a los cuales se les da soporte, y además proveen una descripción de cada uno de ellos.

Los archivos active y newsgroups son usados para guardar y describir los grupos de noticias hospedados en el servidor. En ellos se encuentran los grupos de noticias en los que se tiene interés en publicar y recibir artículos, y además, algo de información administrativa. Estos archivos se pueden encontrar en el directorio /var/lib/news/.

El archivo active determina a que grupos de noticias se le da soporte. Su sintaxis es lineal. Cada línea del archivo active contiene cuatro campos delimitados por un espacio en blanco:

name himark lomark flags

El campo name es el nombre del grupo. El campo himark es el mayor número que se ha usado para un artículo en ese grupo. lomark es usado para guardar el número más bajo de un mensaje activo.

El campo flag, debe contener alguno de estos parámetros:

- **y**. Permite la publicación de forma directa en el servidor.

- **n.** Publicar directamente en el servidor no esta permitido. Esto previene que los lectores de noticias publiquen de forma directa los artículos en el servidor. Los artículos nuevos, deben venir de otros servidores de noticias.
- **m**. El grupo está moderado. Cualquier artículo publicado en este grupo es desviado hacia la dirección del moderador, para su aprobación antes de ser publicado. La mayoría de los grupos, no están moderados.
- Los artículos en estos grupos no son almacenados, solamente son pasados a otro servidor. Esto causa que el servidor de noticias acepte los artículos, pero todo lo que hace es reenviarlos al siguiente servidor que se encuentra más alto en la cadena de flujo. Esto no permite que los artículos estén disponibles para lectura por parte de los usuarios de ese servidor.
- **x**. Este grupo de noticias no acepta artículos. La única forma de que los artículos sean recibidos por este servidor, es que provengan de otro servidor de noticias. Los lectores de noticias, no podrán acceder para publicar artículos.
- **=foo.bar**. Los artículos son guardados en el servidor local con el nombre de grupo foo.bar.

control 0000000000 0000000001 y
junk 0000000000 0000000001 y
rec.crafts.brewing 0000000000 0000000001 y

El archivo newsgroups no es muy sofisticado. Solamente provee una breve descripción (de una sola línea) de los grupos de noticias. Algunos lectores son

capaces de leer este archivo y presentarle la información al usuario para ayudarlo a decidir si quiere suscribirse al grupo descrito.

name description

El campo name es el nombre del grupo de noticias, y el campo description la descripción del mismo.

rec.crafts.brewing.ales Elaboración casera de cerveza negra y rubia
rec.crafts.brewing.badtaste Elaboración casera de cerveza adulterada

c. Proveedores de Noticias

En el archivo newsfeeds se encuentran determinados los artículos que serán enviados. El formato de newsfeeds puede parecer un poco complicado al principio. El formato es el siguiente:

```
site:pattern:flags:param
site2:pattern2\
      :flags2:param2
```

El campo site nombra el sitio al cual ese alimentador relaciona. El nombre del sitio puede ser codificado de la forma que uno quiera y no tiene que ser el nombre del dominio del sitio. Este nombre será usado posteriormente y se referirá a una

entrada en una tabla que provee el nombre del servidor al programa innxmit que transmite los artículos a través de NNTP hacia el servidor remoto. Se puede tener múltiples entradas para cada sitio; cada entrada será tratada individualmente.

El campo pattern especifica que grupos son enviados a ese servidor. Por defecto, son enviados todos los grupos. Si lo desea, solamente deje este campo en blanco. Este campo es usualmente una lista de expresiones que corresponden a un patrón de búsqueda, delimitado por comas. El carácter * equivale a cualquier carácter, incluyendo al cero. El carácter "." no tiene ningún significado especial, el carácter "!" realiza la operación lógica NOT, y el carácter "@" al comienzo del nombre de un grupo significa que no se envíen o reenvíe ningún articulo publicado en el grupo. Esta lista, es leída y analizada gramaticalmente de izquierda a derecha, así que asegúrese de ingresar las reglas específicas al principio.

El campo flags controla y restringe los artículos que van al proveedor de noticias. Este campo se encuentra delimitado por comas y contiene una lista de cualquiera de los siguientes comandos

- **<size.** El tamaño del artículo debe ser menor que lo expresado, en bytes.
- **Aitems.** Los artículos serán verificados. items puede ser uno o más de d (deberá contener cabecera de distribución) o p (no se verificará el destino en la cabecera path).
- **Bhigh/low.** Define el tamaño del buffer antes de escribirlo en la salida.

- **H[count].** El artículo deberá tener por lo menos count saltos; por defecto, es 1.
- **Isize.** Tamaño del buffer interno (para el archivo de salida).
- **Mpattern.** Solo los grupos moderados pueden hacer uso del patrón.
- **Npattern.** Solo los grupos sin moderar pueden hacer uso del patrón.
- **Ssize.** Iniciar la cola de mensajes si el tamaño especificado en bytes es alcanzado.
- **Ttype.** Tipo de alimentación con el proveedor: f (archivo), m (canalizar; el campo param contiene el nombre al cual serán suministrados los artículos), p (tubería (pipe) que apunta a un programa), c (envía al canal de stdin los parámetros en param), y x (parecido al parámetro c pero manejando los comandos de stdin).
- **Witems.** Que se escribirá: b (el tamaño del artículo en bytes), f (la ruta de acceso completa), g (el primer grupo de noticias), m (el identificador de artículo), n (la ruta de acceso relativa), s (origen del artículo), t (antigüedad), * (nombre del canal alimentador o todos los lugares donde llegará el artículo), N (cabecera del grupo de noticias), D (cabecera de distribución), H (todas las cabeceras), O (datos de información general), y R (datos de réplica).

El campo param tiene una codificación especial que es dependiente del tipo de suministro. En las configuraciones más comunes es donde se especificará el nombre del archivo de salida donde se escribirá el suministro de salida. En otras

configuraciones, puede dejarlo fuera. También, dependiendo de la configuración, puede tener otro significado.

El programa nntpsend maneja la transmisión de los artículos usando NNTP como protocolo invocando al comando innxmit. El programa nntpsend tiene un archivo de configuración llamado nntpsend.ctl.

El archivo nntpsend.ctl le permite asociar un nombre de dominio completo, algunas restricciones acerca del tamaño de los suministros, y un número de parámetros acerca de las transmisiones de un sitio en particular. El nombre del sitio significa excepcionalmente un suministro lógico de los artículos.

sitename:fqdn:max_size:[args]

- **sitename.** El nombre del sitio escrito en el archivo newsfeeds.
- **fqdn.** El nombre de dominio completo del servidor de noticias que será suministrado con los artículos.
- ***max_size.*** El máximo volumen de artículos a suplir en una sola transferencia.
- **Args.** Argumentos adicionales que serán pasados el comando innxmit.

d. El archivo readers.conf

Determina quién puede acceder al servidor como cliente para leer noticias. Se puede especificar que los clientes se identifiquen por varios métodos, qué grupos puede leer qué cliente o grupo de clientes y muchas cosas más.

El archivo por defecto deja leer sólo a clientes que se conectan desde localhost y prohíbe cualquier otro acceso.

```
auth "localhost" {
   hosts: "127.0.0.1"
   default: "<localhost>"
}
```

Adicionar la red que se desee tenga acceso al servidor:

```
auth "localhost" {
   hosts: "100.100.100.*, 127.0.0.1"
   default: "<localhost>"
}
```

Esto dará acceso de lectura y escritura (incluyendo la posibilidad de publicar mensajes de moderación en grupos moderados) a toda la red local.

Para casos más complicados habría que crear otras entradas.

e. Puesta en marcha y manejo del servidor

Verificar que todos los archivos y directorios implicados tengan los permisos correctos. Al ejecutar el programa /usr/local/news/bin/inncheck se muestra todos los

errores en permisos de archivos, directorios, y algún otro error de sintaxis que se haya cometido al editar los archivos de configuración.

Es importante que se corrija todos los errores (principalmente de permisos incorrectos) que indique este programa antes de arrancar el servidor; de no hacerlo, podría negarse a iniciar o funcionar incorrectamente.

Corregido todos los errores que inncheck indique, Iniciar el servicio como usuario news:

su news

/usr/local/news/bin/rc.news start

Para verificar si el servicio está funcionando realizar un telnet al puerto 119:

telnet localhost 119

Si funciona bien se mostrará algo así:

Trying 127.0.0.1...
Connected to localhost (127.0.0.1).
Escape character is '^]'.
200 srSin nombre 1vFedora InterNetNews server INN 2.4.1 ready

La gestión de grupos se realiza mediante el programa /usr/local/news/bin/ctlinnd.

Para crear un grupo

/usr/lib/news/bin/ctlinnd newgroup varios

Borrar un grupo

/usr/lib/news/bin/ctlinnd rmgroup varios

Convertir un grupo normal en moderado

/usr/lib/news/bin/ctlinnd changegroup varios m

Convertir un grupo moderado en normal

/usr/lib/news/bin/ctlinnd changegroup varios y

3.9. IPTABLES

Un firewall debe permitirte acceder a todos los servicios ofrecidos en internet, pero protegiendo el sistema y los datos.

Ejecutar el siguiente comando:

```
cat > /etc/rc.d/init.d/firewall << "EOF"
#!/bin/sh

# Inicio de $rc_base/init.d/firewall

# Inserta los módulos de seguimiento de la conexión (no es necesario si
# se compilaron en el kernel).

modprobe ip_tables
modprobe iptable_filter
modprobe ip_conntrack
modprobe ip_conntrack_ftp
modprobe ipt_state
modprobe iptable_nat
modprobe ip_nat_ftp
modprobe ipt_MASQUERADE
modprobe ipt_LOG
modprobe ipt_REJECT

# Permitir conexiones estrictamente locales
iptables -A INPUT  -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

# Permitir reenvío
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A FORWARD -m state --state NEW -i ! ppp+       -j ACCEPT

# Hacer enmascaramiento (no es necesario si la red interna no usa
# direcciones ip privadas)
iptables -t nat -A POSTROUTING -o ppp+ -j MASQUERADE

# Registra todo para el depurado (la última de todas las reglas, pero
# antes de DROP/REJECT)
iptables -A INPUT   -j LOG --log-prefix "FIREWALL:INPUT  "
iptables -A FORWARD -j LOG --log-prefix "FIREWALL:FORWARD"
iptables -A OUTPUT  -j LOG --log-prefix "FIREWALL:OUTPUT "

# Establece una política sensata
iptables -P INPUT   DROP
iptables -P FORWARD DROP
iptables -P OUTPUT  DROP
```

```
# Redirecciona todas las peticiones hacia puerto el 80 del
# exterior para pasar a través del puerto donde escucha el Proxy
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-
port 80


# Muestra más información para direcciones ip dinámicas (no es necesario
# en el caso de IP estáticas)
echo 2 > /proc/sys/net/ipv4/ip_dynaddr

# Desactiva ExplicitCongestionNotification (Notificación Explícita de
# Congestión)
echo 0 > /proc/sys/net/ipv4/tcp_ecn

# Activa TCPsyncookies
echo 1 > /proc/sys/net/ipv4/tcp_syncookies

# Activa Verificación de ruta = Protección contra engaños IP (IP spoofing)
for f in /proc/sys/net/ipv4/conf/*/rp_filter; do
      echo 1 > $f
done

# Activa el reenvío IP (IP forwarding)
echo 1 > /proc/sys/net/ipv4/ip_forward
EOF
```

CAPÍTULO IV. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

- Se ha desarrollado una guía metodológica para la administración de servidores de correo en plataformas GNU/Linux, con la cual se obtiene beneficios en seguridad, administración, escalabilidad, automatización, reducción de costos.
- Los programas de correo Sendmail, Postfix, Qmail, etc. pueden ser configurados para dar servicio de correo electrónico en una red local
- Los programas de correo permiten limitar el tamaño máximo que un mensaje de correo tendrá en el momento de ser enviado. El tamaño puede ser limitado tanto en el encabezado del mensaje como en el cuerpo del mismo.
- El spam es una plaga en Internet. Una de las formas de lidiar con él desde el lado del servidor es configurando correctamente procmail o utilizando programas especializados (Spamassassin, MailScanner, etc.) para estos fines.
- La configuración de los programas antivirus para que busquen cualquier virus en los archivos adjuntos de los correos reducen el riesgo de infección de virus.
- La implementación de firewalls reduce el riesgo de acceso al sistema por parte de usuarios no deseados.

- Las distribuciones de Linux, Sendmail, openldap, innd, mailman, dovecot, imapd, ipop3d, bind, etc. se distribuyen bajo licencia Gnu, es decir, son de libre distribución por lo tanto no influye en gastos.

4.2. RECOMENDACIONES

- En entornos de alto riego a ataques es necesario conseguir que la intervención del operador sea mínima.
- Seleccionar el programa de correo que se ajuste a las necesidades de la organización donde se desee configurar el servidor de correo.
- El tamaño de los encabezados no deben ser demasiado grandes, para evitar desbordamientos de búfer.
- El parámetro required_hits en spamassassin/local.cf debe contener un valor 5 debido a que muchos programas firman sus mensajes con un valor de 5.
- Combinar los programas spamassassin, amavis-new, mailscanner, etc. para tener una mejor protección antivirus y antispam
- Mantener actualizadas la bases de datos de los programas antivirus.
- Definir las reglas de firewalls solo para aquellos servicios que se desee que los usuarios tengan acceso.
- Mantener actualizada las versiones de los programas que se estén utilizando en especial las opciones de seguridad.
- Continuar la investigación, para de esta manera seguir aportando con nuevas ideas en la administración de los servidores de correo.

BIBLIOGRAFÍA

LIBROS

MOHR, James. LINUX, Recursos para el usuario. 1ra ed. México, 1997. Prentice Hall. 796p.

TACKETT, Jack Jr. USING LINUX Special Edition. 1ra ed. USA, 1995. Macmillan Computer Publishing. 862p.

ZIEGLER, Robert L. 2000. Guía avanzada Firewalls Linux. 1ra ed. Trad por José I. Sánchez. Madrid – España. Prentice-Hall. 456p.

HATCH, B; LEE, J; KURTZ G. 2001. Hackers en Linux. Secretos y soluciones para la seguridad en Linux. 1ra ed. Trad por Gregorio Carabaldo Montaño. Madrid – España. McGraw-Hill. 268p.

SERY, Paul G. 1998. Linux Network Toolkit. IDG Books Worldwide, Inc.

HUNT, Craig. 2003. Sendmail cookbook. 1ra ed. California**,** United States of America. O'Reilly. 408 p.

STANGER, James; LANE, Patrick. 2001. Hack proofing Linux: a guide to open source security. 1 ed. United States of America. Syngress. 704p.

RED HAT, Inc. 2003. Red Hat Enterprise Linux 3 Manual de referencia. 1 ed. United States of America. 296p.

MCCARTY, Hill. 2004. Learning Red Hat Enterprise Linux & Fedora, 4th ed. United States of America. O'Reilly. 352p.

MOORE, S.; FOX, T.; FULLER, J.; HA, J.; BAILEY, E. 2003. Official Red Hat Linux User's Guide. Indianapolis, United States of America. Wiley Publishing. 422p.

EILERT, J.; EISENHAENDLER, M.; MATTHAEUS, D.; SALM, I. 2003. Linux on the Mainframe. New Jersey, United States of America. Prentice Hall PTR. 464p

RFCs

RFC2821: Simple Mail Transfer Protocol (SMTP).

RFC2060: Internet Message Access Protocol (IMAP) Version 4rev1.

RFC1939: Post Office Protocol Version 3 (POP3).

RFC822: Estándar para el formato de mensajes de texto de Internet.

RFC819: Convenciones de nombres de dominio para aplicaciones de usuarios de Internet (Domain Naming Convention for Internet User Applications)

RFC1123. Es una extensión del RFC821 y del RFC822

RFC1651. ESMTP (Extended SMTP)

RFC1652. Extensión 8-BIT-MIME

RFC1653. Extensión para declaración de tamaños de mensajes

DIRECCIONES DE INTERNET

FAQ de sendmail: http://www.his.com/~brad/sendmail/index.html

Sendmail-mini-COMO, por Ignacio Llona: http://www.infor.es/LuCAS/Otros/html/sendmail-minicomo/

Infovía-HOWTO, por Francisco José Montilla: http://www.infor.es/LuCAS/COMO-INSFLUG/html/Infovia-Howto/Infovia-Howto.html

Páginas man de sendmail, fetchmail y procmail, postfix

Hatch, Brian; Filtering E-Mail with Postfix and Procmail; Security Focus; http://www.securityfocus.com/infocus/1593

Ben Koette, Patrick; Postfix SMTP AUTH (and TLS) HOWTO; Postfix Howtos, Guides and Tips by Ralf Hildebrandt and Patrick Koetter; http://postfix.state-of-mind.de/patrick.koetter/smtpauth/

GLOSARIO DE TERMINOS

Clave privada. Es la clave que tan sólo nosotros conocemos y que utilizamos para desencriptar el mensaje que nos envían encriptado con nuestra clave pública. Este sistema de clave pública y clave privada se conoce como sistema asimétrico.

Clave pública. Es la clave que hacemos que esté al alcance de todo el mundo para que nos puedan enviar un mensaje encriptado. También con ella pueden desencriptar lo que les enviemos encriptado con nuestra clave privada.

Clave secreta. Es el código básico utilizado para encriptar y desencriptar un mensaje. Cuando se utiliza la misma para las dos funciones, estamos ante un sistema simétrico.

DNS (Domain Name Service) Base de Datos distribuida que mapea nombres de sistemas con direcciones IP y viceversa.

Dominio. Conjunto de computadoras que comparten una característica común, como el estar en el mismo país, en la misma organización o en el mismo departamento. Cada dominio es administrado un servidor de dominios.

Header (cabecera) Parte inicial de un paquete, que precede a los datos propiamente dichos y que contiene las direcciones de origen y destino, control de

errores y otros campos. Una cabecera es también la porción de un mensaje de correo electrónico que precede al mensaje propiamente dicho y contiene, entre otras cosas, el emisor del mensaje, la fecha y la hora.

Host (sistema central) Computador que permite a los usuarios comunicarse con otros sistemas centrales de una red. Los usuarios se comunican utilizando programas de aplicación, tales como el correo electrónico, Telnet y FTP.

IMAP. Protocolo de Acceso a Mensajes de Internet (Internet Message Access Protocol). Protocolo diseñado para permitir la manipulación de mailboxes remotos como si fueran locales. IMAP requiere de un servidor que haga las funciones de oficina de correos pero en lugar de leer todo el mailbox y borrarlo, solicita sólo los encabezados de cada mensaje. Se pueden marcar mensajes como borrados sin suprimirlos completamente, pues estos permanecen en el mailbox hasta que el usuario confirma su eliminación.

Intranet. Una red privada dentro de una compañía u organización que utiliza el mismo software que se encuentra en Internet, pero que es solo para uso interno.

IP address (Dirección IP) Dirección de 32 bits definida por el Protocolo Internet en STD 5, RFC 791. Se representa usualmente mediante notación decimal separada por puntos.

Local Area Network (LAN) (Red de Area Local) Red de datos para dar servicio a un área geográfica pequeña, un edificio por ejemplo, por lo cual mejorar los protocolos de señal de la red para llegar a velocidades de transmisión de hasta 100 Mbps (100 millones de bits por segundo).

Mail gateway (pasarela de correo) Máquina que conecta entre sí a dos o más sistemas (incluso diferentes) de correo electrónico y transfiere mensajes entre ellos. A veces, la transformación y traducción pueden ser muy complejas.

MAN: Metropolitan Area Network. Red de Area Metropolitana.

MIME. Extensiones de Correo de Internet de Múltiples propósitos (Multipurpose Internet Mail Extensions) Técnica para codificar archivos y anexarlos a un mensaje de correo electrónico. Permite principalmente enviar archivos binarios como parte de un mensaje.

MTA. Agente para el transporte de correo electrónico (Mail Transport Agent) son programas que se encargan de distribuir los mensajes generados en el sistema. El más popular es el llamado sendmail, distribuido con sistemas UNIX.

MTU: Maximum Transmission Unit. Unidad Máxima de Transmisión. Tamaño máximo de paquete en protocolos TCP/IP como el PPP.

PGP: Pretty Good Privacity. Paquete de encriptación basado en clave publica escrito por Phil Zimmerman.

POP. Protocolo de Oficina de Correos (Post Office Protocol) Programa cliente que se comunica con el servidor, identifica la presencia de nuevos mensajes, solicita la entre de los mismos y utiliza al servidor como oficina despachadora de correo electrónico cuando el usuario envía una carta.

SMTP: Simple Mail Transfer Protocol. Protocolo de Transferencia Simple de correo. Es el protocolo usado para transportar el correo a través de Internet.

TCP: Transmission Control Protocol. Protocolo de control de Transmisión. Uno de los protocolos más usados en Internet. Es un protocolo de capa de transporte.

TCP/IP (Transmission Control Protocol/Internet Protocol) Arquitectura de red desarrollada por la "Defense Advanced Research Projects Agency" en USA, es el conjunto de protocolos básicos de Internet o de una Intranet.

Trojan Horse (Caballo de troya) programa informático que lleva en su interior la lógica necesaria para que el creador del programa pueda acceder al interior del sistema que lo procesa.

ANEXOS

A. POSTFIX

f. Instalación

La instalación en las distribuciones actuales se limita a usar el paquete suministrado en el formato adecuado (.deb, .rpm).

rpm –i postfix<version>.rpm

g. Configuración

La configuración de postfix se encuentra en el directorio /etc./postfix.

El archivo de configuración fundamental de Postfix es un archivo de texto plano llamado main**.**cf. En él residen los cientos de parámetros de configuración que se le pueden suministrar a este servidor de correo. Afortunadamente, la mayoría de dichos parámetros son opcionales o vienen de serie ya configurados con unos valores adecuados. Así que para una configuración básica es necesario cambiar algunos valores.

A los parámetros se le asigna un valor con el simple método:

parametro = valor

Dentro de la parte valor se puede utilizar el valor de otro parámetro mediante de la utilización del carácter "$" delante del nombre del parámetro:

miparametro = mivalor
otroparametro = $miparametro

Los parámetros que casi cualquier configuración debe modificar son los que indican el nombre y el dominio de la máquina donde está ubicada el servidor. Muchos otros parámetros (incluidos parámetros que vienen configurados por defecto) utilizan estos valores, así que es fundamental su correcta configuración.

- **myhostname** sirve para indicar a Postfix el nombre de la máquina donde reside.
- **mydomain** le indica al servidor el nombre de dominio de la máquina (normalmente sufijo del parámetro anterior).

myhostname = srvLinux.ejemplo.com
mydomain = ejemplo.com

Para configurar el nombre con el que saldrán los correos enviados desde el sistema se utiliza el parámetro *myorigin*. Por defecto el parámetro myorigin tiene el valor $myhostname. Esto significaría que los correos saldrían con origen

usuario@srvLinux.ejemplo.com. Para no incluir el nombre de la máquina del servidor del correo en las direcciones, sino únicamente el dominio: usuario@ejemplo.com es necesario modificar este parámetro:

myorigin = $mydomain

El parámetro mydestination especifica que dominios entregar localmente, en vez de enviarlo a otras maquinas. mydestination es una lista de nombres de dominios separados por comas y espacios en blanco. También se pueden incluir nombres de ficheros donde exista una lista de dominios y tablas de búsqueda

Por defecto, Postfix sólo acepta correo dirigido a sí mismo. Esto es tanto como decir que la variable está configurada con el siguiente valor:

mydestination = $myhostname, localhost.$mydomain, $mydomain

La configuración correcta de este parámetro es fundamental. Si se olvida de alguno de los nombres posibles de su máquina de correo, parte del correo enviado entrará en un ciclo sin fin hasta que el mensaje sea descartado por excesivo número de saltos de correo.

Es bastante probable que en sistemas reales esta lista sea todavía mayor, teniendo en cuenta que la lista de servicios o alias puede ser mayor.

Para saber por que no se entrego correo, la directiva notify_classes indica el nivel de error a notificar. Los valores que puede tener son:

- **bounce**: envía a postmaster copias de los correos no entregados, pero estas copias son modificadas para proteger la privacidad del mensaje.
- **2bounce**: envía dos copias del mail que rebota
- **policy**: informa a postmaster las peticiones rechazadas por políticas UCE de otros servidores. Llega una copia de la transacción
- **protocol**: informa a postmaster cualquier error de protocolos, cliente o servidor, o intentos de algún cliente de ejecutar comandos no implementados. Se recibe una copia de la transacción completa
- **resource**: informa a postmaster de los mail no entregados por algún problema de recursos (errores read/write, queue, etc.)
- **software**: informa a postmaster de problemas de software

Cualquiera de estas opciones son combinables.

notify_clases = resource, software

La directiva mynetworks permite que una red se considere local para Postfix. Esto es para distinguir entre maquinas conocidas de las extrañas (fuera de la red). Las maquinas consideradas como locales pueden usar a MAILSVR como un open relay incluso.

mynetworks = 200.200.200.0/28, 127.0.0.0/8

El parámetro inet_interfaces indica que interfaces de red debe escuchar MAILSVR. Los correos enviados a usuario@ejemplo serán entregados localmente, y direccionados a un dominio que este listado en $mydestination.

El valor por defecto es all (todas las interfaces). Si se tienen interfaces virtuales, se debe indicar cuales de las interfaces escuchar.

inet_interfaces = all
inet_interfaces = virtual.host.name # dominio virtual
inet_interfaces = $myhostname localhost.$mydomain # mailer no virtual

La opción relay_domains restringe los dominios donde los clientes usan a MAILSVR para enviar correo o que destinos va a servir MAILSVR. Por defecto, Postfix relega correo a: clientes confiables que su dirección esta en $mynetworks, clientes confiables que estén en $relay_domains o algún subdominio, clientes no confiables los cuales el destino sea $relay_domains o algún subdominio de él.

h. Bloqueo de SPAM mediante postfix

Se puede seguir varias técnicas para bloquearlo, o bien una combinación de todas ellas.

- Bloqueo de mensajes según remitente.

- Bloqueo de mensajes según listas negras de dominios.
- Bloqueo de mensajes según el contenido de las cabeceras.

i. Bloqueo de mensajes según remitente

Para bloquear los mensajes por un remitente concreto, dominio o parte del dominio, es necesario crear o editar un archivo llamado access en el directorio de postfix, como por ejemplo:

kornet.net 517 Delivery not authorized, message refused
kr 517 Delivery not authorized, message refused
goodlook@korea.net 517 Delivery not authorized, message refused

Tras lo cual es necesario crear el fichero con extensión *db* correspondiente, mediante el comando “postfix access” en el directorio del postfix.

Por último, hay que añadir la siguiente línea en el fichero /etc./postfix/main.cf:

smtpd_sender_restrictions = hash:/etc/postfix/access

j. Bloqueo de mensajes según listas negras de dominios

Otro de los sistemas para bloquear SPAM es el de no aceptar correo si viene de un "relay" abierto. Para ello, existen iniciativas en Internet tales como www.ordb.org,

que contienen una base de datos de IPS desde las cuales es posible enviar SPAM ya que permiten hacer "relay" sin control del remitente.

Para activar esta opción, basta editar el fichero /etc./postfix/main.cf para que contenga las siguientes líneas:

maps_rbl_domains = relays.ordb.org
smtpd_recipient_restrictions =
permit_mynetworks, check_relay_domains

k. Bloqueo mediante chequeo de cabeceras

Cuando todo falla, siempre se puede bloquear correo mediante una regla que lee la cabecera del mensaje y lo rechaza o acepta en caso de que encuentre un patrón. Esto se consigue teniendo en el archivo /etc./postfix/main.cf una línea como la que sigue:

header_checks = regexp:/etc/postfix/header_checks

Y creando un fichero header_checks en la misma localización.

*/Content-[[Tt]]ype:.*charset="big5"/ REJEC*
*/Content-[[Tt]]ype:.*charset="ks_c_.*"/ REJECT*
*/Content-[[Tt]]ype:.*CHARSET="KS_C_.*"/ REJECT*
*/Content-[[Tt]]ype:.*CHARSET=KS_C_.*/ REJECT*

B. QMAIL

INSTALACIÓN DE QMAIL

a. Directorio de qmail

Crear el directorio de trabajo para qmail. La sugerencia de los creadores es el directorio /var/qmail.

 # mkdir /var/qmail

b. Usuarios para qmail

Qmail requiere la creación de diversos usuarios para su correcta ejecución. Estos son: alias, qmaild, qmaill, qmailp, qmailq, qmailr, y qmails. Si por algún motivo no se puede emplear estos pseudos usuarios, entonces se deberá especificar los nuevos valores en el archivo conf-users. Igualmente se requiere de la creación de dos grupos (especificados en el archivo conf-groups.)

Para crear los usuarios y grupos usar:

 # 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
```

c. Compilación y configuración el directorio de trabajo

En este paso se generan los ejecutables de qmail y se prepara el directorio de trabajo de qmail:

```
# make setup check
```

d. Especificación del nombre del host

Especificar el nombre del host (incluyendo el dominio completo) mediante el comando config-fast del siguiente modo:

```
# ./config-fast vmdebian.correo.com
```

e. Alias del sistema

En qmail, el correo para los usuarios especiales postmaster, MAILERDAEMON y root, es redirigido hacia el pseudo-usuario alias. Esto requiere de la existencia de ciertos archivos en el "home directory" del pseudo-usuario alias:

```
# (cd ~alias; touch .qmail-postmaster \
.qmail-mailer-daemon .qmail-root)
# chmod 644 ~alias/.qmail*
```

f. Especificación del agente de procesamiento de correo

El correo dirigido a los usuarios locales debe ser almacenado en algún archivo o directorio (el mailbox.) Esto normalmente no lo realiza el MTA, sino que lo delega a un programa auxiliar. Sendmail normalmente emplea a procmail para este fin. qmail-local es el agente alternativo que proporciona qmail.

```
# cp /var/qmail/boot/proc /var/qmail/rc
```

Sin embargo, procmail en este caso será ejecutado mediante un usuario no privilegiado, por lo que es menester cambiar los permisos del directorio de los mailbox (que se mantendrá en /var/spool/mail.)

```
# chmod 1777 /var/spool/mail
```

g. Eliminación Sendmail

Determinar si sendmail está en ejecución ejecutando la siguiente instrucción:

 # ps ax | grep sendmail
698 ? S 0:00 sendmail: accepting connections
787 pts/0 S 0:00 grep sendmail

La primera línea indica que sendmail está en ejecución, por lo que se debe hacer que termine. Para esto usar cualquiera de los siguientes comandos en orden de preferencia:

 # service sendmail stop
 # /etc/rc.d/init.d/sendmail stop
 # kill 698

El "698" del último comando corresponde al PID del proceso y se obtiene del comando anterior.

Eliminar el paquete sendmail del sistema:

 # rpm -e sendmail

Probablemente se necesite desinstalar otros paquetes (como fetchmail y mutt.) Para forzar la desinstalación de sendmail:

 # rpm -e -nodeps sendmail

El archivo REMOVE.sendmail muestra otras maneras de trabajar sin necesidad de eliminar el paquete sendmail, aunque lo anterior es más recomendable.

h. Instalacíón del reemplazo de sendmail

Diversos programas asumen la existencia de sendmail y lo invocan ciegamente. Por esto, qmail proporciona un reemplazo básico para sendmail, a fin de mantener operativas a las aplicaciones mencionadas.

```
# ln -s /var/qmail/bin/sendmail /usr/lib/sendmail
# ln -s /var/qmail/bin/sendmail /usr/sbin/sendmail
```

i. Instalación de los manuales

Qmail proporciona páginas de manual para diversas utilidades. Estas se instalan en /var/qmail/man. Sin embargo, el sistema man debe ser configurado para acceder a éste. Para esto, añadir el directorio de los manuales mediante la directiva MANPATH en el archivo /etc./man.config:

```
MANPATH /usr/share/man
MANPATH /usr/man
MANPATH /usr/X11R6/man
MANPATH /usr/lib/perl5/man
MANPATH /usr/kerberos/man
MANPATH /usr/local/man
# Añadido para qmail
MANPATH /var/qmail/man
```

Luego, probar algo como man qmail-send.

INSTALACIÓN DE TCPSERVER

Qmail necesita de un mecanismo que lance el demonio qmail-smtpd cada vez que llega un intento de conexión SMTP del exterior del mailserver. Esto se puede hacer de diversas maneras; sin embargo, los creadores recomiendan el uso del programa tcpserver que está disponible como parte del paquete ucspi-tcp de D.J. Bernstein. Es posible configurar inetd para este fin e incluso xinetd.

a. Descargar y desempacar ucspi-tcp

En el site de ucspi-tcp (http://cr.yp.to/ucspi-tcp.html) se puede encontrar la última versión del paquete. Este viene en un archivo TAR comprimido.

Desempaquetar en un lugar razonable mediente un comando como:

 # cd /usr/local
 # tar xvzf /ruta_al_empaquetado/ucspi-tcp-0.88.tar.gz

b. Compilación e instalación de tcpserver

Compilar los programas del paquete. Para esto ejecutar:

 # cd ucspi-tcp-0.88
 # make

Y tras unos momentos se deberá tener una serie de ejecutables en el mismo directorio. Copiar los ejecutables tcpserver y tcprules a un directorio en el PATH, como /usr/sbin o /usr/local/bin:

 # cp tcpserver tcprules /usr/sbin

Sólo estos dos ejecutables son necesarios, por lo que se puede eliminar el directorio ucspi-tcp-0.88.

CONFIGURAR EL INICIO AUTOMÁTICO DE QMAIL

Configurar el sistema para que siempre se ejecute qmail al reiniciarse el computador.

Averiguar el UID y el GID del usuario "qmaild" y del grupo "nofiles" respectivamente:

 # id qmaild
uid=502(qmaild) gid=502(nofiles) groups=502(nofiles)

El número 502 es el UID del usuario "qmaild", y el número asociado al grupo "nofiles" es 502. Estos valores serán distintos en otros sistemas.

Al final del archivo /etc./rc.d/rc.local añadir los siguientes comandos:

csh -cf '/var/qmail/rc &'
/usr/sbin/tcpserver -u 502 -g 502 0 smtp /var/qmail/bin/qmail-smtpd &

PROBAR QMAIL

Para las pruebas que siguen, se recomienda disponer de un computador auxiliar configurado para enviar y recibir correo.

Qmail enviará mensajes de diagnósitco a syslog, por lo cual normalmente se debería buscar en el archivo /var/log/maillog que es donde syslog imprime los mensajes de e-mail.

a. Iniciando qmail

Se ha configurado qmail para que se ejecute cada vez que el computador es reiniciado.

Para analizar si los procesos de qmail están en ejecución, lanzar el siguiente comando:

 # ps axu | grep qmail
qmails 3727 1392 pts/2 S Jan13 0:00 qmail-send
qmaill 3728 1360 pts/2 S Jan13 0:00 splogger qmail
root 3729 1348 pts/2 S Jan13 0:00 qmail-lspawn
qmailr 3730 1348 pts/2 S Jan13 0:00 qmail-rspawn
qmailq 3731 1340 372 pts/2 S Jan13 0:00 qmail-clean

Lo importante es el nombre de los procesos, y de los usuarios dueños de los mismos.

Para no reiniciar el computador, iniciar qmail manualmente:

 # csh -cf '/var/qmail/rc &'
 # /usr/sbin/tcpserver -u 502 -g 502 0 smtp /var/qmail/bin/qmail-smtpd &

b. Probar el delivery

Observar si los mensajes de qmail pueden ser distribuidos, es decir, pueden ser enviados a los usuarios del sistema o de otro.

En primer lugar, enviar un mensaje con destino local, para lo cual se debe indicar el nombre de un usuario común del sistema (distinto del administrador.) Esta prueba se debería efectuar con un usuario normal.

$ echo to: usuario | /var/qmail/bin/qmail-inject

Escribir con cuidado el "to:" separando el nombre de usuario. Probar el envío a un usuario local inexistente:

$ echo to: inexistente | /var/qmail/bin/qmail-inject

Probar ahora el envío a un computador remoto:

$ echo to: usuarioremoto@remoto.correo.com | /var/qmail/bin/qmailinject

Esto requiere que qmail se conecte al puerto SMTP de remoto y que allí exista el usuario especificado.

c. Probar la recepción

Para esto, tcpserver debe estar escuchando en el puerto SMTP (25). Esto puede analizarse fácilmente con netstat:

 # netstat -a -inet | grep smtp
*tcp 0 0 *:smtp *:* LISTEN*

Si esto tarda mucho, o no funciona bien, usar:

 # netstat -an -inet | grep 25
tcp 0 0 0.0.0.0:25 0.0.0.0: LISTEN*

Desde el computador remoto.correo.com, probar el envió de un mensaje a cualquier usuario (no root) del computador local vmdebian.correo.com, y observar los mensajes del log de ambos sistemas.