Instalación y configuración de un servidor de correo electrónico con filtro antispam y antivirus

Autor: Félix de Jesús Zamora Nava

Tesina presentada para obtener el título de:
Ing. En Sistemas computacionales

Nombre del asesor:
Juan Carlos Guzmán Contreras

UNIVERSIDAD VASCO DE QUIROGA

ESCUELA DE SISTEMAS COMPUTACIONALES

"INSTALACIÓN Y CONFIGURACIÓN DE UN SERVIDOR DE CORREO ELECTRÓNICO CON FILTRO ANTISPAM Y ANTIVIRUS "

TESINA

QUE PARA OBTENER EL TITULO DE:

INGENIERO EN SISTEMAS COMPUTACIONALES

PRESENTA:

FÉLIX DE JESÚS ZAMORA NAVA

ASESOR:

I.E. JUAN CARLOS GUZMÁN CONTRERAS

CLAVE: 16PSU0049F

MORELIA, MICHOACÁN, MÉXICO. ENERO DE 2006.

En la actualidad el ser humano ha sustituido medios de comunicación como el teléfono y el fax por el correo electrónico, ya que en la mejor de las situaciones es rápido y barato, es excelente para responder con rapidez a preguntas simples.

Los mensajes enviados por correo electrónico pueden ser de cualquier tipo ya sea a nivel personal, académico o empresarial. Ha evolucionado impresionantemente ya que ahora es posible anexar archivos a los mensajes y contenido multimedia, estas características han hecho que le den un uso para enviar mensajes de publicidad que en la mayoría de los casos no han sido pedidos por el usuario y es llamado correo basura.

Para solucionar el problema del correo basura existen programas que permiten el bloqueo de estos mensajes así como el bloqueo de virus que podrían ocasionar perdida de información de los usuarios del correo electrónico.

Instalar un servidor de correo electrónico con antivirus y antispam para evitar la entrada masiva de correo electrónico no deseado (*spam*), es la forma de evitar el problema de los mensajes basura, con esto se evitaría la pérdida de tiempo y dinero para las empresas que ofrecen este servicio tan útil hoy en día.

El *spam* puede ocasionar un sin fin de problemas para los administradores del servicio de correo electrónico, ya que puede verse en la necesidad de empezar a bloquear remitentes para evitar que el servidor se sature de correo basura, para cubrir esta necesidad se propone el uso de *sendmail* que es el software encargado de enviar y recibir correo junto con herramientas para evitar correo no deseado y virus.

En este proyecto se propone la instalación y configuración de *sendmail* con antispam y antivirus bajo la plataforma de Linux en su distribución *Fedora Core 3*, debido a que *Red Hat* dejo de producir versiones gratuitas por lo que

se creó la comunidad llamada *Fedora* la cual se encarga de publicar actualizaciones cada tres o cuatro meses. *Fedora Core 3* es compatible con las versiones *Enterprise* de *Red Hat*, también se puede usar *White Box* ó *Centos* sin ningún problema de compatibilidad.

Con la implementación de *sendmail* y las herramientas antes mencionadas se puede ofrecer un servicio de correo electrónico libre de correo basura y virus, con esto el administrador aligerará su carga de trabajo en cuanto a la limpieza de los buzones de correo de los usuarios pertenecientes a su empresa.

INDICE

INTRODUCCIÓN

En los últimos años en México el uso del Internet ha ido en aumento y a su vez los medios de comunicación como el teléfono y el telegrama han sido reemplazados poco a poco por los medios electrónicos como el correo electrónico.

Este trabajo de investigación esta orientado para ayudar a los administradores de un servidor de correo electrónico a resolver el problema que genera el exceso de correo basura que se manda sin que sea pedido, el cual es denominado como *spam* además de los virus transmitidos y que pueden llegar a afectar la productividad de la empresa dueña del servidor de correo.

En el primero capitulo llamado *Correo Electrónico* se da una explicación de lo que es, como es su estructura interna y como es su funcionamiento, esto es para poder entender el proceso de envío recepción de mensajes electrónicos, para poder ofrecer una verdadera solución al problema que se va atacar.

El segundo capitulo titulado *Spam*, explicara lo que es, sus características y algunas formas para evitarlo, este es el problema que se va a solucionar pero para eso se necesitara saber como se comporta para dar una solución definitiva.

El tercer capitulo *Análisis de herramientas*, trata de un análisis de las herramientas existentes para evitar el *Spam* y los virus en el Sistema Operativo Linux y se definen las utilidades que se van a emplear para la implementación del servidor de correo seguro.

El último capitulo *Sendmail con SpamAssassin y Clam Antivirus* es la instalación y configuración del software para poder tener un servidor de correo electrónico eficiente que evite a los usuarios el tener que estar

borrando correo indeseado de las bandejas de entrada así como la saturación de la misma por el s*pam*.

CAPITULO I
CORREO ELECTRONICO

1.1 Descripción

El correo electrónico es un servicio que permite a los usuarios comunicarse mediante el Internet, además ofrece la posibilidad de enviar y recibir archivos, dependiendo del proveedor del servicio puede variar el tamaño de estos archivos y también el tamaño de la bandeja de entrada o buzón de correo.

El correo electrónico fue creado en 1971 por *Ray Tomlinson*, que aunque en ese entonces ya existía un software capaz de comunicar a dos personas entre si, una caracteristica del sistema de *Tomlinson*, es la posibilidad de enviar el mensaje a cualquier red mientras que con *SNDMSG* solo permitía el envió y recepción de mensajes dentro de la misma red, es por esta razón que el *e-mail* revolucionó la forma de comunicación del ser humano mediante el Internet.

Existen diferentes protocolos que pueden intervenir para la transmisión de correo electrónico, a continuación se describen brevemente:

a) ***Simple Mail Transfer Protocol*** **(SMTP)** o protocolo simple de transferencia de correo electrónico, usado para transferir mensajes de correo electrónico, esta diseñado únicamente para el envío de mensajes de texto, incluso es usado en celulares o *PDA's* que son dispositivos que no tienen la capacidad para recibir correos con contenido multimedia.

   Este protocolo fue creado principalmente para *ARPANET* con el objetivo de intercambiar mensajes mediante el programa *SNDMSG* que posteriormente *Ray Tomlinson* usaría para la creación del correo electrónico como actualmente lo conocemos.

   Para más información sobre este protocolo consultar el RFC 821[1].

[1] http://www.ietf.org/rfc/rfc0821.txt

b) ***Post Office Protocol* (POP)** o protocolo de oficina de correo, este es más nuevo que el protocolo mencionado anteriormente, las diferencias son que este sirve solo para la recepción de correo electrónico, no necesita una conexión permanente a la red, ya que cuando hace la conexión es cuando solicita al servidor el envió de los mensajes recibidos, si se esta conectado permanentemente a la red con la ayuda de un programa como *Outlook, Eudora* o *Thunderbird* se puede estar monitoreando para que avise la llegada de nuevos mensajes.

Existe una nueva versión de este protocolo llamado *POP3*[2] la diferencia es que pueden acceder a una sola bandeja de entrada y los *IMAP* que proporcionan accesos a múltiples carpetas en los servidores.

c) ***Internet Message Access Protocol* (IMAP[3])** La diferencia de este protocolo con el *POP3* es que cualquier computadora con acceso a Internet puede configurar la cuenta de correo y se puede especificar que carpetas mostrar y cuales ocultar, este protocolo esta diseñado para consultar el correo electrónico en un servidor donde se almacenan los mensajes.

Para poder acceder a una cuenta de correo electrónico es necesario contar con el software adecuado que se ajuste al protocolo del servidor que se esta empleando, mas sin embargo existe otra forma de verificar correo nuevo por medio del *webmail* el cual consiste en utilizar cualquier navegador de Internet para consultar la bandeja de entrada.

Algunos de los proveedores de correo electrónico gratuitos mas usados son:

- *Gmail* http://www.gmail.com
- *Hotmail* http://www.hotmail.com

[2] http://www.faqs.org/rfcs/rfc1939.html
[3] http://www.faqs.org/rfcs/rfc3501.html

- *Yahoo!* http://www.yahoo.com
- *Xasamail* http://www.xasamail.com
- *Starmedia* http://www.starmedia.com
- *Lycos* http://www.lycos.com

Finalmente estas son las ventajas que tiene el correo electrónico y por lo cual es una herramienta tan popular en Internet:

a) Cada día existen nuevos proveedores de correo electrónico en la red y muchos ofrecen el servicio gratuitamente.
b) Es rápido, generalmente un correo electrónico puede tardar solo unos minutos en llegar a su destinatario, esto es una gran ventaja sobre el correo tradicional ya que se puede enviar en instantes un mail al otro lado del mundo y llegara sin contratiempos.
c) Un solo correo puede ser enviado a múltiples usuarios sin necesidad de pagar estampillas como en el correo tradicional.
d) Es ecológico, debido a que son solo bytes no contaminan y se evita la tala de árboles para la creación de hojas de papel que sirven para escribir y enviar cartas de la manera tradicional.
e) A diferencia del antiguo correo, en el correo electrónico se puede enviar mensajes multimedia gracias a las características que ofrece *MIME*[4] (*Multi-Purpose Internet Mail Extensions*, Extensiones de correo Internet multipropósito).
f) Se puede retransmitir los mensajes sin necesidad de volver a teclearlos ya que existe la posibilidad de guardarlos en el servidor.
g) Seguridad mediante cifrado lo cual permite tener la certeza de que el correo no ha sido alterado o modificado.

[4] http://www.rfc-es.org/getfile.php?rfc=2045

1.2 Estructura

El correo electrónico se conforma de una dirección que se tiene que especificar para saber a quien le va a llegar el mensaje que será transmitido, la estructura general es *usuario@maquina.dominio*, las partes de un correo electrónico se explican a continuación:

- ***Usuario:*** Consiste en el usuario al que le va a llegar el mensaje que se enviará, este es elegido por el propio dueño de la cuenta.
- **@:** Significa en. Este símbolo sirve para indicar hacia que máquina y dominio es dirigido el mensaje
- ***Máquina:*** Este es el nombre del servidor de correo al que pertenece el correo electrónico y es separado por un punto del dominio. Ejemplo: *pepe@hotmail.com*.
- ***Dominio:*** Es el tipo de dominio al que pertenece la máquina.

Los dominios más comunes son:

- *.com*: para las empresas.
- *.edu:* para las Instituciones educativas.
- *.org:* para las Organizaciones no comerciales.
- *.gov ó .gob:* para las Instituciones gubernamentales.
- *.mil:* para una Institución militar.
- *.net:* para una red específica.
- *.aero:* para la industria del transporte aeréo.
- *.info:* para cualquier uso.
- *.museum:* para uso de museos.

Algunas veces después del dominio se les puede agregar otras dos letras para definir el país como por ejemplo:

- *.mx:* para México.
- *.us:* para los Estados Unidos.
- *.it:* para Italia
- etcétera

La estructura que todo mensaje de correo electrónico debe tener, se especifica enseguida (observe la figura 1):

- ***From o De:*** En este renglón se especifica quien es el autor del mensaje recibido.
- ***To o Para:*** Aquí se especifica quien va a recibir el correo pueden ser uno o varios destinatarios.
- ***CC (Con Copia):*** El mensaje puede ser enviado con copia para varios destinatarios.
- ***CCO (Con Copia Oculta):*** La diferencia con el anterior es que con esta opción no se mostraran todas las direcciones a las cuales fue enviado el correo.
- ***Fecha y hora:*** Los datos de cuando y a que hora fue recibido el mensaje en nuestro buzón.
- ***Subject o Asunto:*** En este renglón se especifica el tema a tratar en el mensaje para que el destinatario pueda identificar de que se trata.
- ***Attach o Archivo adjunto:*** Se puede enviar un archivo adjunto en el correo electrónico, con esta característica es fácil enviar documentos con un formato distinto al que el correo electrónico permite como por ejemplo archivos de *Microsoft Word* o de *Acrobat Reader*.
- ***Cuerpo:*** En esta parte va el texto que escribió quien envía el mensaje.
- ***Firma:*** Esta va después del cuerpo pero no siempre aparece ya que es opcional, aunque es recomendable ponerla por buenos modales.

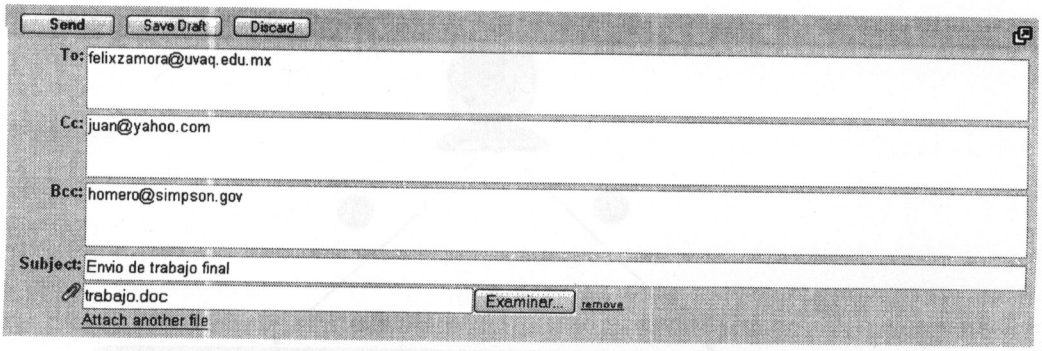

Figura 1 Ejemplo del encabezado de un e-mail en una aplicación de correo electrónico

1.3 Funcionamiento

Ahora que ya se sabe que es un correo electrónico, sus ventajas y como esta conformado solo falta saber como se comporta realmente en la red. En este apartado se explica cual es el camino que debe seguir un mensaje para llegar a su destino, en la figura 2 se muestra el proceso.

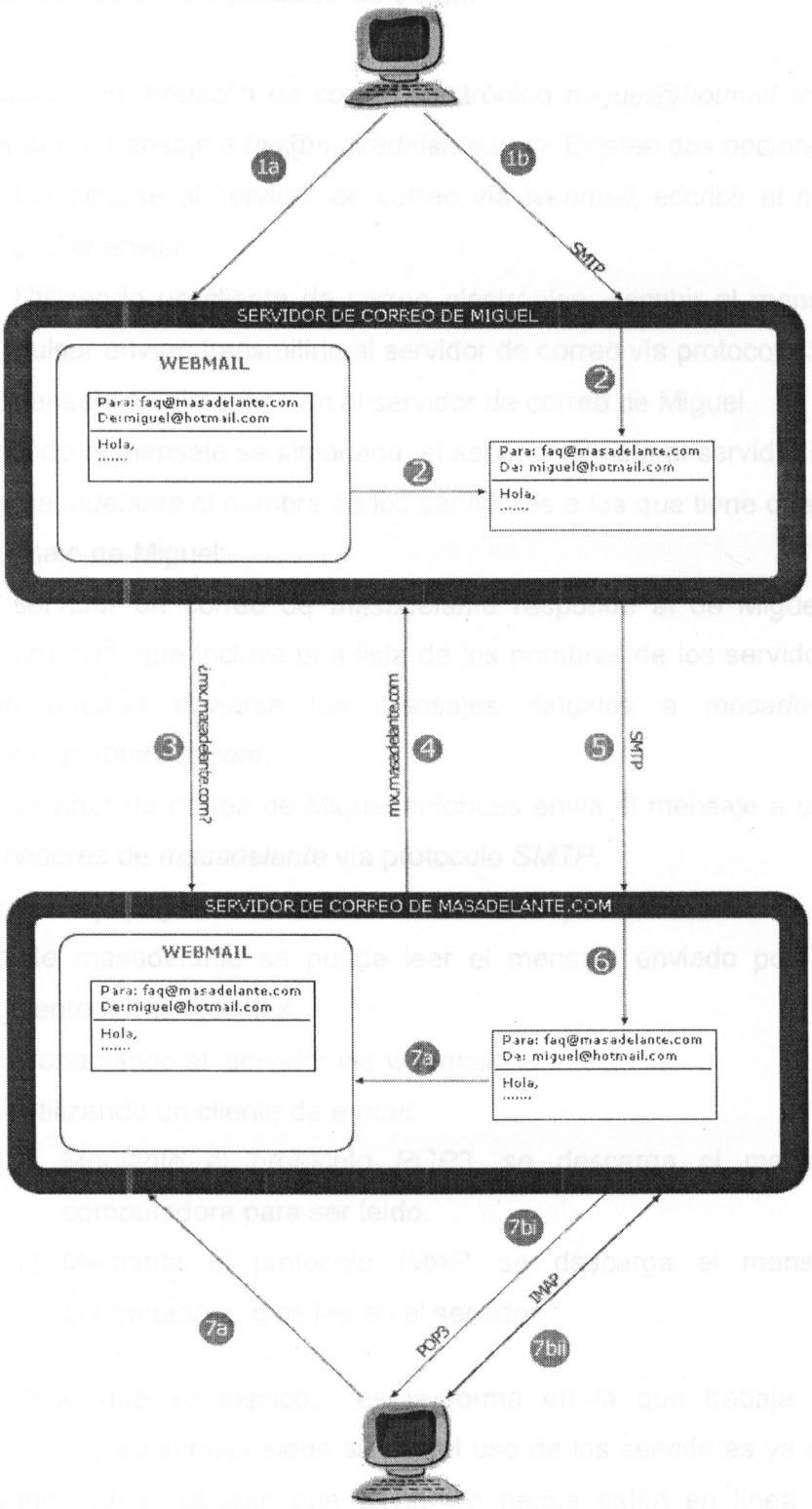

Figura 2 Funcionamiento del correo electrónico

1.3.1 Descripción del proceso de e-mail

1) Miguel, con dirección de correo electrónico *miguel@hotmail.com* quiere enviar un mensaje a *faq@masadelante.com*. Existen dos opciones:
   a) Conectarse al servidor de correo vía *webmail*, escribir el mensaje y pulsar enviar.
   b) Utilizando un cliente de correo electrónico, escribir el mensaje y, al pulsar enviar, transmitirlo al servidor de correo vía protocolo *SMTP*.
2) El mensaje se almacena en el servidor de correo de Miguel.
3) Cuando el mensaje se almacena, el servidor solicita al servidor de correo de *masadelante* el nombre de los servidores a los que tiene que enviar el mensaje de Miguel.
4) El servidor de correo de *masadelante* responde al de Miguel con un *record mx*[5], que incluye una lista de los nombres de los servidores a los que pueden enviarse los mensajes dirigidos a *masadelante.com* (*mx.masadelante.com*).
5) El servidor de correo de Miguel entonces envía el mensaje a uno de los servidores de *masadelante* vía protocolo *SMTP*.
6) El mensaje llega al servidor de *masadelante* y se almacena.
7) Desde masadelante se puede leer el mensaje enviado por Miguel la siguiente forma distintas:
   a) Conectando al servidor vía *webmail*.
   b) Utilizando un cliente de *e-mail*:
      i) Mediante el protocolo *POP3*, se descarga el mensaje a la computadora para ser leído.
      ii) Mediante el protocolo *IMAP*, se descarga el mensaje a la computadora o se lee en el servidor.

El proceso que se explicó, es la forma en la que trabaja el correo electrónico, y su ventaja sigue siendo el uso de los servidores ya que no es necesario que el usuario que envía y/o recibe estén en línea al mismo

[5] Mail Exchange Record, identificador de un servidor de correo en la base de datos del DNS.

tiempo para poder intercambiar sus mensajes ya que estos podrán ser leídos a cualquier hora y día del año.

CAPITULO 2

SPAM

2.1 ¿Qué es el SPAM?

SPAM son mensajes electrónicos no solicitados y en cantidades masivas. Aunque se pueden enviar por distintas vías, la más utilizada es la basada en el correo electrónico.

La palabra *SPAM* tiene su origen en el año de 1926 cuando una empresa productora de alimentos empacados sacó al mercado un jamón con especias al que llamaron "*Spiced Ham*" (Figura 3), al ser este un producto que no necesitaba estar refrigerado se volvió cada vez mas popular e hizo que estuviera en todos los comercios, en todas las casas e incluso en los regimientos de los soldados americanos y rusos de la segunda guerra mundial.

Figura 3 Spiced Ham

Un grupo de comediantes británicos llamados *Monthy Phyton* crearon una escena en un restaurante en el cual el mesero lee el menú y todos los platillos contenían el tan popular *spam* de tal manera que unos vikingos que estaban sentados en una mesa empiezan a cantar *spam, spam, spam*!, lo

que hace que sea muy molesto para las personas a las que les estaban ofreciendo el menú.

El termino de *spam* se uso posteriormente en el Internet para calificar al correo no deseado ya que genera un descontento entre los usuarios por que pueden saturar el buzón de entrada y dejar de recibir sus correos que tienen un mayor significado en cuanto al contenido del mismo.

El *spam* es el equivalente electrónico del correo basura que llega al buzón de las casas, pero para los que se dedican a enviarlo es mucho mas rentable hacerlo por medios electrónicos ya que un mensaje puede llegar a millones de personas por un costo insignificante para ellos pero muy alto para los proveedores de Internet que al ver que su ancho de banda se satura, necesitan contratar mas ancho de banda y esto indirectamente afecta a los usuarios domésticos al incrementarse los costos por mes del Internet.

Algunas otras definiciones de *spam*:

- ***Spam CSS:*** Mensaje que utiliza estilos en cascada (*Cascading Style Sheets* o CSS), que se usa para controlar el aspecto de páginas Web, y ocultar mensajes de *spam*. Los grupos de spammers también usan CSS para reciclar antiguos trucos basados en lenguaje *HTML (Hypertext Markup Language)* que engañan a los filtros de antispam que no entienden *CSS*.
- ***Spam NDR:*** Mensaje que utiliza una falsa notificación estándar de entrega fallida (*non-delivery report* o *NDR*) que el destinatario considerará auténtica, hecho que le llevará a abrir un adjunto que es *spam*. Los *spammers* pueden enviar una *NDR* directamente o hacer que un servidor legítimo la envíe, de manera que parece aún más creíble.

El *spam* no supone una amenaza para los datos como son los virus pero como se menciono anteriormente las empresas son las más afectadas por algunas razones como:

- El *spam* es una pérdida de tiempo para el personal. Los usuarios sin protección antispam deben comprobar si un mensaje es *spam* antes de eliminarlo de su buzón de entrada.
- Los usuarios pueden fácilmente pasar por alto o incluso eliminar mensajes con contenido realmente importante y que es confundido por *spam*.
- El *spam* como los virus usan ancho de banda y llegan a saturar las bases de datos.
- Algunos mensajes de *spam* pueden contener mensajes ofensivos para el usuario. Se puede responsabilizar al empresario ya que en teoría debe ofrecer un ambiente de trabajo seguro.
- Los *spammers* a menudo usan computadoras de otros usuarios para practicar sus actividades ilegales.

Los *spammers* quieren saber quien recibe sus mensajes y quien no, de modo que puedan seguir utilizando las direcciones activas para pode llegar cada vez mas a un número mayor de usuarios de correo electrónico.

Aunque no se responda al remitente del mensaje de *spam*, el *spammer* tiene algunas formas de saber si se ha recibido su mensaje o no.

- Si el programa de correo electrónico del usuario le permite previsualizar los mensajes (muestra una parte del contenido del mensaje, siempre y cuando se use software que permita interpretar código *HTML* como *Outlook* ó *Eudora*), el *spammer* puede ver que el correo ha sido recibido sin problemas.

- Si se hace clic en algún enlace dentro del mensaje de *spam* que permita dar de baja la dirección de correo electrónico, lo único que se hará es avisar al *spammer* que esta dirección es valida.
- Los mensajes de *spam* pueden incluir *scripts* que son invisibles para el usuario y sirven para conectarse a la página *Web* del *spammer* para notificarle en cuanto el *spam* sea leído o previsualizado.

Por si fuera poco al problema del correo no solicitado se le agrega uno más que son los virus informáticos ya que los *spammers* y programadores de virus pueden crear sociedades para darles más problemas a los usuarios de correo electrónico.

Los virus pueden constituir nuevas oportunidades para el *spam*. Un programador de virus que permite a otros usuarios poder controlar una maquina remotamente sin que el dueño de la misma se de cuenta; si el virus que llega por medio de un correo electrónico infecta la computadora, enviara un mensaje al atacante y este hace una lista de computadoras infectadas que después le venderá al *spammer* para que realice sus actividades ilícitas desde estos equipos.

Actualmente, el *spam* se envía mediante equipos infectados, de esta manera es más difícil rastrear la pista de su origen.

Del mismo modo, las técnicas de *spam* pueden ayudar a los creadores de virus, un programador puede enviar un virus a un gran número de usuarios utilizando la lista de direcciones del *spammer*. Con infinidad de destinatarios, es probable que un gran número de ellos active el virus, garantizando así su rápida propagación.

Por lo anterior es importante que se conozcan las características del *spam*.

2.2 Características del SPAM

Los mensajes de *spam* generalmente anuncian un sitio *Web* con contenido pornográfico, explican alguna "formula" para ganar dinero rápidamente, o simplemente muestran un gran numero de artículos para su venta, que en la mayoría de los casos no son de gran uso. Aunque el *spam* puede llegar de diferentes maneras generalmente llega con ofrecimiento de promociones, ventas con descuentos, etc.

El *spam* usa técnicas para mejorar su clasificación dentro de los buscadores, estas técnicas están prohibidas por la mayoría de los motores de búsqueda, ya que causan daños contra los algoritmos que usan, algunas de estas técnicas son:

- ***Meta-refresh:*** algunos propietarios de sitios crean páginas objetivo (*target pages*) que automáticamente conducen a los visitantes a diferentes páginas dentro del sitio. El índice *meta-refresh* es una de las formas de hacerlo. Algunos motores de búsqueda no indexarán páginas con un rango de meta refrescantes alto, o redireccionadas.
- ***Texto Invisible:*** es la técnica de colocar texto en una página del mismo color que el fondo, haciéndolo invisible a la vista humana.
- ***Texto pequeño:*** es una técnica que coloca texto sobre una página en un tamaño de fuente muy pequeño. Las páginas donde predomina el texto pequeño se confunden con *spam*, o en su defecto no se indexa el texto pequeño.
- ***Spam con imágenes:*** es una forma muy fácil de evitar ser detectado por los filtros antispam, aunque su defecto es que solo puede ser leido cuando se usa algún programa que pueda leer código *HTML*, la forma de evitar estos mensajes sería deshabilitando la opción de interpretar *HTML* en el programa de correo electrónico.

Algunos de los contenidos más comunes dentro de los correos que contiene *spam* son los siguientes temas:

- Cadenas de cartas. Los clásicos correos electrónicos que al final contienen una nota como "reenvíelo a 10 personas y sobrevivirá".
- Negocios piramidales, incluido el Multilevel Marketing (MLM).
- Otros mensajes tipo "Hágase rico rápidamente".
- Anuncios de *Web* pornográficos o líneas eróticas.
- Programas para acumular direcciones y hacer *spamming*.
- Oferta de acciones de empresas desconocidas.
- Remedios milagrosos.
- Niños/Ancianos muy enfermos. Penurias y tragedias difíciles de creer.
- Falsas amenazas de virus.
- Programas piratas (*warez*).

Los *spammers* buscan constantemente formas de camuflar sus mensajes y tratar de engañar a los programas antispam, a continuación se listan algunos de los trucos más utilizados:

a) ***Perdido en el espacio:*** El *spammer* pone espacios entre las letras de las palabras que se quiere ocultar, por ejemplo "m e d i c a m e n t o", creyendo que el software antispam no leerá las letras como una sola palabra, este método no es muy eficiente y es fácil de detectar.

b) ***El agujero negro:*** Los *spammers* usan código *HTML*, que es el código con el cual se crean las paginas *Web*, para insertar un espacio entre las letras, aunque también reduce el tamaño del espacio a cero, por ejemplo:

   Lo que ve el programa antispam:

V i a g r a

Lo que se muestra en el contenido del correo electrónico: Viagra

c) ***Tinta invisible:*** A veces, los *spammers* trataran de engañar al programa antispam de una manera ingeniosa haciéndole creer que es un texto inofensivo y por otro lado oculta el mensaje que quiere que vea el destinatario, esto se logra poniendo el mensaje de *spam* del mismo color del fondo, en este caso hacen uso otra vez del código HTML.

Lo que ve el programa antispam:

<body bgcolor=white> Viagra ¡Hola hijo! Ayer me la pase muy bien en la cena. ¡Hasta pronto! Con cariño, Mamá. </body>

Lo que se muestra en el contenido del correo electrónico: Viagra

d) ***El micro punto****:* El *spammer* inserta una letra adicional en medio de una palabra que quiere ocultar, pero utiliza un tamaño de fuente muy pequeño, esto hace que el programa antispam vea la letra y lea la palabra de forma incorrecta evitando ser detectado como correo no deseado pero el destinatario no tiene problema al leer el mensaje.

e) ***Devolución al remitente:*** El *spammer* envía deliberadamente un correo electrónico a una dirección no válida, pero pone una dirección válida en el campo "De". El correo no puede entregarse, pero el proveedor de correo devuelve el mensaje a la dirección válida del campo "De".

f) ***El juego de números:*** En este método se utiliza el valor ASCII de los caracteres y el código HTML en lugar de escribir las letras

como comúnmente se hace, en el ejemplo se puede ver la forma de crear los mensajes con código *HTML* y *ASCII*.

Lo que ve el programa antispam:

V i a &#103 r &#97

Lo que se muestra en el contenido del correo electrónico: Viagra

g) ***Trituradora:*** En este método se usan tablas *HTML* para "triturar" el texto en columnas verticales, como si el mensaje hubiera pasado por una trituradora de papel logrando así engañar al programa antispam.

Lo que ve el programa antispam:

```
<table border="0">
 <tr>
  <td>V</td><td>i</td><td>a</td><td>g</td><td>r</td><td>a</td>
 </tr>
 <tr>
  <td>C</td><td>r</td><td>&eacute;</td><td>d</td><td>i</td>
  <td>t</td><td>o</td>
 </tr>
 <tr>
  <td>G</td><td>r</td><td>a</td><td>t</td><td>i</td><td>s</td>
 </tr>
</table>
```

Lo que se muestra en el contenido del correo electrónico:

Viagra

Crédito

Gratis

Como se puede apreciar cada día hay nuevas técnicas para crear *spam* y poder engañar a los programas antispam, es por eso que se ha convertido en una guerra entre creadores y bloqueadores de correo no deseado, esto poco a poco va creando un gran problema para los usuarios como son los virus.

2.3 Como evitar el SPAM

En este tema se explicarán algunas de las formas para evitar el *spam*, ya que es muy complicado librarse de este mal que afecta cada día más a los usuarios del correo electrónico. Si el proveedor de correo no tiene activado un filtro antispam entonces se tendría que recurrir a algunas herramientas para hacer más fácil la tarea.

Los siguientes programas son *shareware* (versión de prueba por 30 días) o *freeware* (software completamente libre), estos programas ayudaran para evitar el *spam* y el bombardeo de correo basura al buzón de entrada, estos programas realizan diferentes técnicas como enviar un mensaje al *postmaster* con las direcciones de donde proviene el correo basura o en su defecto eliminar el correo no deseado de una manera transparente para el usuario.

- **Antispam 1.0**[6]: Este tiene la ventaja de que su base de datos de *spammers* se actualiza constantemente desde su sitio *Web*.

[6] http://www.xde.net/antispam

- ***Bounce Spam Mail 1.8***[7]***:*** Este programa envía un mensaje falso a los *spammers* indicándoles que la dirección de correo no existe y así se logra que la dirección de correo sea borrada de su lista.
- ***E-Mail Remover 2.4***[8]***:*** Permite ver las cabeceras de los mensajes en el servidor antes de descargarlos, permitiendo su eliminación.
- ***MailShield 1.0***[9]***:*** Una barrera que filtra el *spam* y el bombardeo de correo en el servidor antes de que el usuario entre a su buzón.
- ***SpamBuster 1.61***[10]***:*** Funciona con el tradicional sistema de filtros, bajando únicamente las cabeceras de los mensajes para evitar perder tiempo.

Es muy difícil encontrar un método que sea totalmente seguro para evitar recibir *spam*, ni siquiera el software especializado puede garantizar esto, por lo que el evitarlo estará prácticamente en las manos del usuario ya que el es el único que maneja su cuenta de correo y sabe a quien la ha dado y en donde la ha publicado, por lo que se pueden seguir los siguientes consejos que ayudaran para caer en el mundo del *spam*.

a) **Usar un programa antispam:** Un programa de este tipo puede reducir considerablemente el volumen de mensajes no deseados, sobre todo si tiene la capacidad de "aprender" a distinguir los mensajes de *spam* de los que no.
b) **No comprar en los sitios publicados por spam:** Si el usuario realiza una compra de un sitio que ha sido publicado por medio del *spam* solo estará contribuyendo a su crecimiento, además de que puede convertirse en el objetivo de otras empresas ya que al comprar en línea es un cliente en potencia.

[7] http://ay.home.ml.org/bsm
[8] http://home.pacific.net.sg/~thantom/eremove.htm
[9] http://www.mailshield.com
[10] http://www.contactplus.com

c) **Si no se conoce la dirección de origen, borrar el mensaje:** Esto es para evitar perder el tiempo revisando correo basura que puede llegar a contener un virus.

d) **No responder mensajes de spam ni hacer clic en sus enlaces:** El problema si se responde un mensaje de este tipo el usuario simplemente le esta avisando al *spammer* que su cuenta de correo esta activa, esto puede ocasionar el recibir mas *spam*.

e) **Desactivar opciones para recibir información y ofertas:** En ciertos formularios de la red muchas veces a la hora del registro se pide que se confirme esta opción lo cual no se debe aceptar ya que solo enviaran ofertas muchas veces inútiles para el usuario.

f) **No usar el método de "previsualización" del programa de correo:** Muchos programas que sirven para revisar el correo pueden tener la opción de previsualización, los *spammers* tienen técnicas para saber cuando un mensaje ha sido visto aunque sea parcialmente, la mejor opción es basarse en el asunto del correo para detectar si es o no *spam*.

g) **Usar el campo "Cco" para mensajes a más de una persona:** Cuando se tiene la necesidad de enviar un mensaje a más de una persona se debe usar el campo "Cco" (con copia oculta) para evitar que sean interceptadas por los *spammers*, es por eso que no es recomendable el estar enviando cadenas.

h) **No proporcionar la cuenta de correo en sitios públicos:** Los *spammers* tienen herramientas que se dedican a buscar cuentas de correo electrónico en los sitios de Internet como pueden ser los foros.

i) **Proporcionar dirección electrónica solo a gente de confianza:** Con la finalidad de tener un control sobre quien tiene la dirección de correo, aunque siempre se corre el riesgo de que incluyan la cuenta en algún mensaje en cadena.

j) **Tener una o dos direcciones electrónicas "alternativas":** Una dirección alternativa será útil para darla en sitios públicos donde sea requerida, como son en encuestas publicas, foros, chats etc., con esto

se logra que la dirección electrónica personal la sepa solo la gente de confianza.

A veces es complicado evitar el *spam* pero con las sugerencias anteriores seguramente se vera reducido el correo basura en el buzón de correo. Existe otra opción para tratar de evitarlo, como el poner un filtro antispam en el servidor de correo pero esto sólo lo puede poner el administrador del servidor de correo.

CAPITULO 3
ANÁLISIS DE HERRAMIENTAS

3.1 Herramientas antispam en Linux

Una de las principales características que ofrece el Sistema Operativo Linux es que su libre distribución motiva a los desarrolladores a crear aplicaciones para su uso libre y sin costo alguno en algunos casos, esto representa una ventaja sobre otros sistemas operativos que pueden ofrecer las mismas herramientas pero siempre tendrán un costo por lo cual aquellas pequeñas empresas que necesiten un servidor de correo electrónico con antispam siempre puede optar por usar Linux en cualquiera de sus distribuciones y hacer uso del software libre.

Existe software antispam tanto a nivel de usuario o complementos para los servidores de correo electrónico, a continuación se describirán algunos de los programas para Linux:

a) **Bogofilter**[11]**:** Es un filtro de correo electrónico que lo clasifica como *spam* o *ham* (no *spam*) por medio de un análisis estadístico de las cabeceras y el contenido de los mensajes. El programa es capaz de aprender la clasificación y correcciones que haga el usuario. Bogofilter provee procesamiento para texto plano y *HTML*. Esta escrito en lenguaje C y es extremadamente rápido comparado con otros filtros similares, el único fallo que puede tener es que ignora archivos adjuntos e imágenes.

b) **BlacMail**[12]**:** Es un antispam usado por algunos de los *MTA* (Agentes Transportadores de Correo) mas usados como son *Sendmail, Qmail y SMail.* Es altamente configurable, poderoso y muy rápido, puede ser ejecutado en una larga variedad de plataformas.

[11] http://bogofilter.sourceforge.net
[12] http://www.jsm-net.demon.co.uk

c) **SpamAssassin**[13]**:** Es un filtro de correo que usa un amplio rango de pruebas heurísticas en las cabeceras y el cuerpo del correo para identificar el *spam*. Una vez identificado el correo puede ser marcado opcionalmente como *spam* para que mas tarde el usuario lo clasifique usando su propia aplicación de correo electrónico. Provee una herramienta de línea de comandos para realizar el filtrado.

d) **GNU SAUCE**[14]**:** Software en contra del correo electrónico no solicitado, es un servidor *SMTP* y se sitúa entre el Internet y el software de mail del usuario, este software hace chequeos extremadamente agresivos en el correo de entrada y sus orígenes, si descubre algún problema el correo es rechazado en el acto, puede agregar las direcciones falsas inmediatamente a su lista negra y nunca deniega la entrada a correos de sitios que se sabe no envían *spam* para hacerse publicidad.

e) **DSPAM**[15]**:** Es un servidor estadístico que se encuentra del lado del agente de los servidores de correo Unix. Es muy eficiente pues se dice que ha rendido cuentas de éxito en el mundo real de una exactitud de 99.9% de efectividad y un rango menor al 0.01% de errores.

f) **Clamfilter**[16]**:** Es un pequeño, seguro y eficiente filtro contenedor hecho para *Postfix* esta diseñado para filtrar mensajes eficientemente por medio del demonio *clamd*.

Como se pudo observar existe una gran variedad de software antispam para Linux, aunque no se han analizado todas las opciones estas son algunas de las mas usadas por los *postmasters* para mantener sus sitios libres del *spam* y evitar la saturación del ancho de banda de cada servidor.

---

[13] http://www.spamassasin.org
[14] http://www.chiark.greenend.org.uk
[15] http://www.nuclearelephant.com/projects/dspam
[16] http://www.ensita.net/products/clamfilter

3.2 Opciones de herramientas antivirus en Linux

Cuando ya se cuenta con un filtro antispam en el servidor de correo electrónico es importante que también se instale un antivirus ya sea en la parte del servidor o por parte del usuario, por que aunque el filtro antispam puede evitar el arribo de correo basura al buzón de entrada no puede evitar que pasen los virus.

Generalmente los correos que traen archivos adjuntos en su cuerpo ya sean imágenes, archivos comprimidos, documentos de *Word* puede que estén infectados con algún virus o troyano, esto puede ocasionar que tanto el servidor o la computadora del usuario queden expuestas a ciertos ataques por medio de la red.

Aunque existen muy pocos virus para Linux seguramente en un futuro no muy lejano habrá una gran cantidad de variaciones de virus y troyanos que afecten los sistemas basados en Unix, es por esta razón que en este apartado se hará un breve análisis sobre los diferentes tipos de antivirus existentes para Linux tanto para el Sistema Operativo como para los agentes de transporte de correo electrónico ya que combinando ambos se obtiene un mejor resultado.

a) **Clam Antivirus**[17]**:** Clam Antivirus es una herramienta antivirus para Unix basado en la licencia *General Public License* (GPL). El propósito principal de este software es la integración con los servidores de correo para el escaneo de archivos adjuntos. El paquete provee un demonio multi-hilo flexible y escalable, un escáner para línea de comandos y una herramienta para que se actualice la base de datos de virus mediante Internet. Lo más importante de este software es que la base de datos siempre se mantiene actualizada.

[17] http://www.clamav.net

b) **Qmail-scanner**[18]: Este software escanea el correo electrónico para que este libre de virus. Ha sido escrito específicamente para *Qmail MTA*, reemplazando el proceso *qmail-queue*. Escanea los archivos adjuntos por medio del escáner de virus instalado en el servidor y genera respuestas de correo cuando un virus es encontrado.

El *postmaster* es el indicado para decidir cual de los antivirus anteriores es el que se adecua mejor a las necesidades del servidor que administra, independientemente de cual elija sabrá que estará más protegido en contra de los ataques de virus.

3.3 SpamAssassin como filtro antispam

SpamAssassin es una herramienta confiable para crear una buena estrategia de filtrado de *spam*. Su combinación de reglas estáticas para el reconocimiento de correo no deseado, su habilidad para adaptarse y aprender dinámicamente nuevas características del correo basura y de los *spammers*, lo hacen atractivo en muchos entornos.

Existen muchos sistemas antispam. SpamAssassin se ha vuelto popular por varias razones.

a) Usan un gran numero de reglas y los valora de acuerdo a su diagnostico. Las reglas que han demostrado ser mas efectivas diferenciando *spam* de no *spam* tiene un valor mayor.
b) Es muy fácil de ajustar los marcadores asociados con cada regla o agregar nuevas reglas basadas en expresiones normales.
c) SpamAssassin se puede ejecutar en diferentes sistemas operativos, aprendiendo a reconocer cual envío será confiable e identificar nuevos tipos de *spam*.

[18] http://qmail-scanner.sourceforge.net/

d) Puede reportar *spam* a los sitios encargados de almacenar direcciones *Web* de empresas dedicadas a hacer *spam* y es capaz de crear trampas para las direcciones de correo electrónico que se dedican al envió de este.

e) Es un software libre, distribuido bajo la licencia pública GNU o bajo la licencia artística. Cualquiera de estas licencias permite al usuario modificar libremente el software y redistribuirlo pero siempre bajo los mismos términos mencionados anteriormente.

Hay muchas maneras de que spamassassin se de cuenta si un mensaje es *spam*.

a) El encabezado del mensaje puede ser revisado para verificar que tenga consistencia y adherencia de acuerdo a los estándares de Internet (ejemplo: si la fecha tiene un formato correcto).
b) Los encabezados y el cuerpo pueden ser revisados para ver si contienen frases que comúnmente son encontrados en el *spam* (ejemplo: "Haga dinero fácil").
c) Los encabezados y el cuerpo pueden ser verificados en muchas bases de datos en línea que siguen el rastro de mensajes verificados como *spam*.
d) La dirección IP del sistema que envía el mensaje puede ser revisada también en bases de datos en línea y verificar si no pertenecen a los *spammers* o son sospechosas.
e) Direcciones específicas, host o dominios pueden ser inscritos en una lista negra o una lista blanca. Una lista blanca puede ser construida automáticamente en base al historial del sistema de envío.
f) SpamAssassin puede ser entrenado para reconocer los tipos de *spam* que se reciben aprendiendo de un grupo de mensajes que el usuario considere *spam* y de un grupo que no considere *spam*.
g) La dirección IP del sistema del remitente puede ser comprobada a

nombre del dominio del remitente usando el protocolo *Sender Policy Framework* (SPF) para determinar si el sistema tiene permitido enviar mensajes a usuarios de ese dominio.

h) Métodos probabilísticos como el filtrado de *Bayes*.

SpamAssassin combina el formato de validación de mensajes, filtro de contenido y la posibilidad para consultar listas negras basadas en red. Filtrar el sistema requiere un poco de intervención del usuario y da un poco de retraso en el proceso de enviar y recibir mensajes. Hay otros métodos para prevenir *spam*, cada uno de ellos conlleva a sus propias ventajas y desventajas.

En un sistema de desafío y respuesta, el sistema retiene todos los mensajes de origen desconocido y se les envía un mensaje de respuesta con un código único o conjunto de instrucciones (el desafío). Los remitentes deben de responder a ese desafío de una manera que verifique esa dirección de correo probando de cierta manera que son "seres humanos" y no un programa de correo masivo (la respuesta). Después de una respuesta satisfactoria, el sistema permite que el remitente sea aceptado en vez de retenerlo.

En sistemas de listas grises, el servidor de correo inicialmente retorna un código de falla temporal a los mensajes de nuevos remitentes o sistemas de envío. Si el sistema intenta reenviar el mensaje después de un periodo razonable de tiempo, el servidor de correo acepta el mensaje y los siguientes mensajes del host remitente, debido a que los *spammers* tratan cualquier falla temporal como una falla permanente o intentan enviar mensajes continuamente durante el periodo de tiempo de las listas grises sus mensajes no son recibidos.

En sistemas de direcciones de tiempo limitado, los usuarios generan variaciones únicas de sus direcciones de correo para incluirlas en los

formularios *Web*, correos de mensajes, foros etc. Las direcciones solo serán validas por un límite de tiempo o podrán ser validas hasta que el usuario las anule. En estos sistemas si un usuario recibe *spam* a una de sus direcciones puede identificar a la compañía que le esta enviando *spam* o le vendió su dirección a los *spammers*, con esto puede actuar rápidamente e invalidar la dirección para no seguir recibiendo correo basura.

3.4 ClamAV como herramienta antivirus

ClamAV es un proyecto abierto a colaboraciones de todo tipo, liberado bajo la GPL[19] (*General Public License*), cuyo objetivo es el de programar un motor antivirus y las bases de datos de firmas actualizadas con un escáner en la línea de comandos, interfaz para sendmail y soporte para escanear archivos comprimidos (.rar, .zip, .chm) entre otras facilidades.

Aunque nació inicialmente como antivirus para sistemas Linux y BSD con tareas principales para servidor, ahora se puede encontrar para otros entornos como Windows, Mac OS X, Solaris e incluso BeOS.

Sobre su efectividad basta decir que durante la expansión de diversas variantes del virus Bagle a finales del 2004, ClamAV fue el tercer antivirus de una larga lista de estos programas en incorporar las firmas necesarias para detectarlas.

Gracias a que es un sistema sin costo y tiene disponible el código fuente, junto a su integración a nivel de servidor, ClamAV es una buena opción para los desarrolladores e integradores de cortafuegos por hardware, por poner un ejemplo. Además, a su alrededor se han desarrollado una serie de herramientas proporcionadas por terceras partes como interfaces gráficas de usuario que facilitan su empleo en computadoras de usuario final.

---

[19] http://www.gnu.org/copyleft/library.html

Pese a todo esto ClamAV no se considera en muchas ocasiones en las comparativas de antivirus, centradas en productos comerciales, lo cual genera que el usuario final no confié en este producto ya que siempre se guían en base a estas comparativas.

A pesar de no ser un antivirus muy conocido ClamAV ofrece las siguientes ventajas:

- Es gratuito, si una empresa tiene un presupuesto bajo, usar ClamAV es una excelente opción.
- Se actualiza diariamente la base de datos de antivirus, y además, esta recibe sus datos de otras muy conocidas, como son *F-Secure*, *Symantec*, *Sophos*, *F-Prot*, *Norman* y *Vexira*.
- Se puede programar para que realice escaneos en el disco duro cuando el usuario lo desee.
- Puede detectar gusanos y troyanos que se pueden considerar como "viejos"
- Utiliza pocos recursos de CPU.

Como todo el software también cuenta con algunas desventajas como pudieran ser:

- No puede detectar virus polimórficos, es decir los virus que pueden estar variando su firma cada vez que se copia o hace una nueva infección para hacer más difícil su detección.
- No puede escanear el sector de arranque del disco duro.
- ClamAV no puede limpiar archivos infectados, es decir no puede quitar los virus de los archivos, solo hace la detección.
- Puede tardar mucho tiempo al analizar archivos de gran tamaño o discos duros grandes lo cual puede ser una de sus mayores desventajas frente a otros antivirus.

CAPITULO 4
SENDMAIL CON SPAMASSASSIN Y CLAMAV

4.1 Que es Sendmail

Existe una gran variedad de programas de correo electrónico que proveen al usuario de una aplicación para la creación y el envío de correo. Estos programas son llamados *MUA* (Agentes de Usuario), y su propósito es el aislar al usuario de los *MTA* (Agentes de Transporte), que son los encargados de enviar los correos a su destino correcto.

Sendmail es el agente de transporte de correo más común de Internet. Las misiones básicas son las siguientes:

a) Recibir los correos provenientes de un Agente de Usuario como puede ser *Eudora* o *Pine*; o provenientes de un Agente de Transporte como puede ser el propio sendmail.
b) Elección de la estrategia de reparto del correo, basándose en la información del destinatario contenida en la cabecera:
    a. Si el mail es local en el sistema, enviará el correo al programa de reparto local de correo electrónico.
    b. Si el mail no es local, sendmail utilizará el *DNS* (*Domain Name System*) del sistema para determinar el host al que debe ser enviado. Para transferir el mensaje, iniciará una sesión *SMTP* con el *MTA* de dicho host.
    c. Si no es posible mandar el correo a su destino, sendmail almacenará estos correos en una cola y volverá a intentar el envió un tiempo después. Si el correo no puede ser enviado después de un tiempo razonable, será devuelto a su autor con un mensaje de error. Sendmail debe garantizar que cada correo llegue a su destino o si hay un error este debe ser notificado.
c) Reformatear el correo antes de pasarlo a la siguiente máquina, según unas reglas de reescritura. Según el tiempo de conexión que se tenga con una determinada máquina, o según el agente de

transporte al que vaya dirigido el correo, se necesitara cambiar los formatos de las direcciones del remitente y del destinatario, algunas líneas de la cabecera del correo o incluso puede que se necesite agregar una línea a esta. Sendmail debe realizar estas tareas para conseguir la máxima compatibilidad entre usuarios distintos.

d) 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.
e) La ejecución como agente de usuario. Aunque no posee interfaz de usuario, sendmail también permite el envío directo de correo a través de su ejecutable.

Todas estas características y muchas otras de sendmail deben ser configuradas y varían dependiendo de las necesidades del sistema donde se este ejecutando.

A continuación se presenta la descripción y ubicación típica de los programas y archivos de soporte usados por sendmail.

- */usr/sbin/sendmail o /usr/lib/sendmail* Ejecutable de sendmail.
- */etc/sendmail.cf o /usr/lib/sendmail.cf* Archivo de configuración de sendmail.
- */etc/aliases* Archivo donde se almacenan los alias del sistema.
- */etc/mail/sendmail.mc* Archivo de configuración del servidor de correo.
- */etc/mail/access* Archivo que contiene la base de datos de acceso al servidor de correo electrónico.

* Alias es un nombre que puede acompañar o reemplazar el nombre de una persona, ejemplo Pepe es un alias de José.

- */usr/bin/newaliases* Reconstruye la base de datos de los alias usados por sendmail a partir del fichero aliases. Es un enlace simbólico a *sendmail* en el modo -bi.
- */var/spool/mqueue* Directorio donde se almacena la cola de correo.
- */usr/bin/mailq* Muestra por pantalla el contenido de la cola de correo. Es un enlace simbólico a sendmail en el modo -bp.

4.2 Funciones de Sendmail

4.2.1 Sendmail como agente de usuario

Como agente de usuario, sendmail lee por defecto su entrada estándar hasta encontrar un fin de archivo (EOF) o una línea con un punto. En ese momento manda una copia de ese mensaje a cada una de las direcciones destino. Sendmail determinará la ruta a seguir, basándose en el contenido de la dirección destinataria.

Sendmail determinará como enrutar un mensaje, de acuerdo a la información que posee en su archivo de configuración.

Ejemplo:

Este comando manda un mail a "*pepe@xyz.com*", sin título, el cuerpo del correo solo contiene la palabra "hola".

$sendmail pepe@xyz.com

Hola

.

$

4.2.2 Sendmail como agente de transporte

La utilización de sendmail como demonio en el sistema permite enviar y recibir correo *SMTP*. Para esto, se queda como un proceso residente escuchando el puerto 25, admitiendo y realizando conexiones *SMTP* cuando sea necesario. Cuando reciba una petición de conexión, creará un proceso hijo que se encargara de ello, mientras el proceso padre seguirá escuchando el puerto 25.

Para utilizar sendmail como demonio se debe iniciar en el arranque del sistema añadiendo al script de inicio la siguiente línea: *sendmail -bd -q15m.*

Con esta orden, sendmail actuara en modo *background* admitiendo conexiones por el puerto 25. La opción *-q15m* indica que actualice la cola de correo cada 15 minutos, el parámetro *-bd* es igual a *-bg*, pero actuando en *foreground.*

4.2.3 La cola de correo

Cuando el envío de un correo no puede alcanzar su destino por que es rechazada la conexión, este debe ser almacenado en una cola de la máquina que manda el correo, para intentar enviarlo mas tarde.

La cola de correo se encuentra en *"/var/spool/mqueue"*, en este directorio se crean archivos temporales para cada correo que se almacena, el formato del nombre de los archivos tienen el prefijo siguiente:

- *df* Archivos donde se guardan los cuerpos de los mensajes, sin las cabeceras.
- *qf* Archivos donde se guarda la información necesaria para procesar los trabajos.

- *tf* Archivos temporales, imagen de los archivos qf cuando estos están siendo reconstruidos.
- *xf* Archivos donde se almacena toda la información transmitida durante la apertura y cierre de una sesión.

Para visualizar el contenido de la cola se usa el comando *"mailq"*, que es un enlace simbólico hacia el comando *"sendmail -bp"*. Este producirá una lista con los identificadores de los mensajes, su tamaño, la fecha en la que el mensaje entro en la cola, el remitente y el destinatario.
Para procesar la cola de correo, se utiliza el comando: *sendmail -q<tiempo>*, que procesa la cola de correo dependiendo del tiempo que se le ponga.

<tiempo> es un número seguido de caracteres. El carácter "s" significa segundos, "m" minutos, "h" horas, "d" días y "w" semanas. Si se omite el parámetro de tiempo, sendmail procesara la cola en ese instante.

Ejemplo:

$sendmail –q1h30m

Actualizara la cola de correo cada hora y media.

4.2.4 Los alias en Sendmail

El uso de alias en correo electrónico permite:

- Tener nombres alternativos (*nicknames*) para usuarios individuales.
- Envío de correo a otras máquinas, aunque la dirección sea local.
- Listas de correo.

Existe un archivo de texto *"/etc/aliases"* donde es posible ver o modificar la base de datos actual de alias del sistema. Este archivo no es usado

directamente por sendmail, sino que es convertido y procesado con el comando *"newaliases"*, que creara la base de datos de alias que si podrá utilizar. El formato del archivo es el siguiente:

alias: recipient, [recipient,...]

- ***Alias:*** es el nombre de la persona a la que se le envía el correo, debe ser un usuario local en la máquina.
- ***Recipient:*** es el nombre al cual se manda el correo. Puede ser un nombre de usuario, un alias o una dirección de correo electrónico completa, conteniendo tanto el nombre de usuario como el nombre de la máquina y su dominio.
- ***[recipient,...]:*** Se pueden añadir varios destinatarios para un mismo alias. Un correo mandado a ese alias será recibido por todos los receptores, una particularidad es que aunque se incluya el nombre de origen a la lista de correo, éste no recibirá el correo a menos que la opción *MeToo* este activada.

Ejemplo del archivo */etc/aliases*:

#nombres especiales para los administradores del sistema

postmaster: Juan

root: felix

#Reenvío del correo a una dirección remota

eliza: eliza@xyz.com

#lista de correo

clientes: pepe, juan@xyz.com, rox@hotmail.com

Si se manda un correo a "clientes", cada uno de los miembros de la lista recibirá una copia del correo enviado.

A nivel de usuario, sendmail también permite a cualquier usuario normal crearse si propia tabla de alias. Ésta viene definida en el archivo *.forward* del directorio *$HOME* de cada usuario, tiene precedencia sobre el archivo */etc/aliases* creado por el root.

Una función especial de sendmail para las listas de correo es la siguiente: *owner -<lista>: <nombre>*, donde *<lista>* es el nombre de una lista de correo.

La persona especificada en *<nombre>* es la responsable de la lista de correo identificada en *<lista>*. Esta función es útil si, por ejemplo, sendmail tiene problemas mandando correo a alguno de los destinatarios de la lista, gracias a esta identificación del responsable de la lista, se le mandará un mensaje de error indicándole que ha ocurrido un error en el envío del correo.

Ejemplo:

clientes: pepe, juan@xyz, al00100@yahoo.com

owner-clientes: marco

Se le enviara un mensaje a Marco indicando que ha ocurrido un error.
Una forma más fácil de mantener listas de correo sería la siguiente línea en */etc/aliases* ó en el *.forward* si no somos súper usuarios:

pepe: "| sendmail'cat<lista>"

Donde *<lista>* contiene todos los suscritores actuales de la lista de correo, con la tubería se consigue que al recibir un mail el usuario "pepe" se ejecute

la línea de comandos que sigue a la tubería. Las prioridades del programa que se ejecute, serán las mismas que las del usuario que recibe el correo.

4.2.5 Sendmail en modo traza

El parámetro *"-dx.1"* establece sendmail en modo traza. La *x* indica la parte de sendmail de la que queremos hacer la traza. El *1* indica el nivel de detalle de la traza, al ejecutar este parámetro y al mandar correo a una dirección va explicando cada uno de los pasos que se va realizando para conseguir mandar el mensaje.

Ejemplo:

sendmail –d21.20 juan@yahoo.com

Los tipos de traza más útiles son los siguientes:

- 21 Reescritura de direcciones
  - – d21.4 Muestra las direcciones rescritas.
  - –d21.20 Muestra la parte izquierda y derecha de las reglas de reescritura
- 37.1 Muestra las opciones utilizadas de sendmail.
- 35.9 Muestra todas las definiciones de macros.
- 32.1 Muestra las líneas de cabecera utilizadas.
- 0.15 Muestra las reglas de reescritura de direcciones.
- 8.8 Muestra la información *DNS* accedida.

4.2.6 Parámetros más comunes de la línea de comandos en Sendmail

-B<tipo>	Establece el tipo del cuerpo del mensaje a <tipo>, 7BIT u 8BITMIME.
-ba	Modo Arpanet. Es el mismo modo que el estándar (-bm), excepto que todas las líneas de entrada deben terminar con CR-LF, todas las líneas de salida terminaran con CR-LF y que extrae el remitente leyendo el campo de la cabecera "From".
-bd	Ejecuta el demonio de sendmail.
-bm	Envía el correo introducido por la entrada estándar hasta la dirección de destino, puede ir seguido por cualquier otro parámetro o modo en la línea de comandos, es el modo de trabajo estándar.
-bp	Visualiza el contenido de la cola de correo.
-bi	Actualiza la base de datos de alias.
-bt	Modo de prueba (*test*) de direcciones, se usa para probar el correcto funcionamiento de las reglas de reescritura de direcciones.
-bv	Modo de verificación, simplemente verifica la existencia de los nombres, no recoge ni envía los mensajes, se usa normalmente para validar usuarios o listas de correo.
-C<archivo>	Usa <archivo> como archivo de configuración de sendmail alternativo.
-dx.1	Pone a sendmail en modo traza.
-f [address]	Establece la dirección fuente de quien envía el correo, será valido solo si la dirección [*address*] se corresponde con el usuario que está ejecutando sendmail, o si éste es un "usuario de confianza" definido en el archivo de configuración.

-h [cont]	Establece el contador de "saltos" a [cont], el contador de "saltos" se incrementa cada vez que un mismo correo es procesado. Cuando se supera el límite, se abandona el envió de ese correo y se le devuelve al emisor con un mensaje de error "demasiados saltos". Por defecto, sendmail determina la cuenta de saltos de un mensaje contando las líneas "*Received:*" del mismo.
-F nombre	Establece el nombre completo del usuario a "nombre".
-Mx valor	Establece la macro "x" al valor especificado.
-n	No utiliza los alias.
-ox valor -O option=val	Establece la opción "x" al valor especificado.
-R <tipo>	Establece el tipo de devolución que hará sendmail si el mail no puede ser mandado a su destino. El parámetro "full" indicará que se devuelva todo el correo y "hdrs" indicará que se devuelva solo la cabecera.
-t	Extrae el o los destinatarios del mismo mensaje. Leerá las direcciones de "To:", "Cc:" y "Bcc:". Cualquier dirección de la línea de comandos será suprimida o sea no recibirá el correo.
-q [intervalo]	Tiempo de procesamiento automático de la cola de correo.

4.3 Instalación y configuración

4.3.1 Instalando Sendmail

El agente de transporte de correo, sendmail, se incluye preempaquetado en la mayoría de las distribuciones de GNU/Linux, así que simplemente se puede instalar cuando se haga la instalación de Linux en la máquina o también se puede instalar por medio de paquetes binarios ya compilados

para la versión de Linux que se este usando, descargando del Internet el archivo con extensión .rpm, simplemente se ejecuta el siguiente comando como root:

 # rpm -Uvh sendmail-x.y.z.rpm

Sin embargo, existen algunas razones para instalar sendmail desde los archivos fuente, especialmente por cuestiones de seguridad, pueden descargarse desde el FTP anónimo en el servidor *ftp.sendmail.org*, una vez obtenido el archivo se coloca en */usr/local/src/*, se procede a descomprimir y a compilar las fuentes:

 # cd /usr/local/src
 # tar xvzf sendmail.x.y.z.tar.gz
 # cd sendmail.x.y.z
 # ./Build

Para completar la instalación de los binarios resultantes, se necesitara ejecutar los siguientes comandos como root:

 # cd obj.Linux.x.y.z.i586
 # make install

Con el proceso anterior se han instalado los binarios de sendmail en el directorio */usr/sbin*. Muchos enlaces simbólicos al ejecutable sendmail también se instalaran en el directorio */usr/bin/*. Con los pasos anteriores el servidor de correo electrónico queda instalado en la maquina local.

4.3.2 Archivos de configuración sendmail.cf y sendmail.mc

En las versiones anteriores sendmail era configurado a través de un archivo de configuración (*sendmail.cf*) que está escrito en un lenguaje difícil de comprender, el día de hoy, sendmail crea todas las opciones de configuración a través de macros, con una sintaxis fácil de entender. El sistema de macros genera configuraciones que cubren muchas de las instalaciones, pero siempre se tiene la opción de configurar manualmente el archivo *sendmail.cf*, para trabajar en un entorno más complejo.

El programa procesador de macros m4 genera el archivo *sendmail.cf* cuando se procesa el archivo de configuración proporcionado por el administrador.

El proceso de configuración es básicamente una forma de crear el archivo *sendmail.mc* apropiado que incluya macros que describan la configuración deseada. Las macros son expresiones que el procesador de macros m4 entiende y expande en la sintaxis compleja de *sendmail.cf*. Las expresiones macro se componen del nombre de la macro, que se asemejan a una función en un lenguaje de programación y de algunos parámetros que se utilizan en la expansión.

4.3.3 Parámetros más usados en sendmail.mc

Algunos de los parámetros más usados en el archivo *sendmail.mc* para la configuración de sendmail son:

- **VERSIONID:** Esta macro es opcional, pero es útil para grabar la versión de configuración de sendmail en el archivo sendmail.cf, por ejemplo: *VERSIONID('setup for Red Hat Linux').*
- **OSTYPE:** Esta es una definición muy importante ya que provoca que se incluya un archivo de definiciones que son opciones

predeterminadas para el sistema operativo en uso. La mayoría de las definiciones en un archivo macro *OSTYPE* configuran los nombres de ruta de los archivos de configuración, transporte de correo y argumentos, así como la localización de directorios que usa sendmail para almacenar los mensajes. La función *OSTYPE* debería ser una de las primeras en aparecer en el archivo *sendmail.mc*, debido a que otras definiciones posteriores dependen de ella. Ejemplo: *OSTYPE(`linux').*

- **DOMAIN:** Esta macro es útil cuando se desea configurar un gran número de máquinas estandarizada. Si se está configurando un pequeño número de anfitriones, probablemente es mejor no preocuparse por esto. La instalación estándar contiene un directorio de plantillas de macros m4 utilizadas para dirigir el proceso de configuración (/usr/share/sendmail.cf). Para hacer uso de una macro *DOMAIN*, se debe crear su propio archivo macro conteniendo las definiciones estándar que se requieran para el sitio, y escribirlas en el subdirectorio *domain*. Ejemplo: *DOMAIN(`vbrew').*
- **FEATURE:** Esta macro permite incluir características predefinidas de sendmail en su configuración, lo cual hace a las configuraciones soportadas muy fáciles de usar. Para usar cualquiera de las características listadas se deberá incluir una línea en el archivo *sendmail.mc* muy parecida a lo siguiente:

  FEATURE(nombre)

  Donde nombre se sustituye con el nombre de la característica, algunas pueden tener un parámetro opcional y podría quedar de la siguiente manera:

  FEATURE(nombre, parámetro)

Donde parámetro es el argumento a suministrar.

- **Definiciones de macros locales:** Los archivos estándar de configuración de sendmail proporcionan una buena cantidad de maneras y variables con las que se puede personalizar la configuración. Las definiciones de macros locales son normalmente invocadas mediante el suministro del nombre de la macro con un argumento representando el valor que se quiere asignar a la variable que maneja la macro.
- **MAILER:** Si se desea que sendmail transporte correo de cualquier otra forma y no solo de manera local, se debe indicarle que medio tiene que usar. La macro *MAILER* hace esto muy fácil. Los transportes más comunes son:
    a) ***local:*** Este transporte incluye el agente de entrega local usado para enviar correo al buzón de los usuarios de la máquina.
    b) ***smtp:*** Este transporte implementa el *SMTP* (Protocolo Simple de Transporte de Correo), que es el medio más usual de transporte de correo en Internet. Cuando se incluye, se configuran cuatro transportes de correo: *smtp* (SMTP básico), *esmtp* (SMTP Extendido), *smtp8* (SMTP binario plano de 8 bits) y *relay* (específicamente diseñado para hacer de transporte a modo de pasarela entre anfitriones).
    c) ***uucp:*** Proporciona soporte para dos transportes de correo: *uucp-old*, que es el *UUCP* tradicional y *uucp-new*, que permite manipular múltiples buzones en una transferencia.
    d) ***usenet:*** Este programa de correo permite enviar mensajes directamente a redes de noticias del estilo Usenet. Cualquier mensaje local dirigido a la dirección *news.group.usenet* será introducido en la red de noticias para el grupo de *news.group*.

e) ***fax:*** Si se tiene instalado Hylafax[20], este transporte permitirá dirigir correo electrónico a él, para que así pueda construir una pasarela de correo-fax.

Hay otros como *pop*, *procmail*, *mail11*, *phquery* y *cyrus* que son útiles, pero menos comunes.

Ejemplo: *MAILER(smtp)*

4.3.4 Generando el archivo sendmail.cf

Cuando se haya terminado la edición del archivo de configuración *sendmail.mc*, se debe de procesar para crear el archivo *sendmail.cf* que es el que lee sendmail, esto se hace con el siguiente comando:

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

Esta orden invoca al procesador de macros m4 suministrándole el nombre del archivo de definición de macros a procesar.

Se puede modificar el archivo *sendmail.mc* las veces que se quiera pero para que surtan efecto en sendmail se tiene que generar de nuevo el archivo *sendmail.cf* cada vez que el archivo *sendmail.mc* sea modificado además de reiniciar el servicio de correo con la orden:

 # service sendmail restart

[20] http://www.hylafax.org/

4.3.5 Configuraciones útiles para Sendmail

Existe una infinidad de configuraciones para *sendmail*, por lo cual solo se mostraran algunas de las configuraciones que pueden ser de gran utilidad.

4.3.5.1 Usuarios de confianza

Cuando surge la necesidad de sobrescribir el campo *From*: de un mensaje de correo que va hacia fuera como cuando se tiene un programa basado en *Web* que genera correo electrónico, normalmente el mensaje aparecerá como proveniente del usuario que es dueño del proceso en el servidor *Web*, pero se requiere especificar otra dirección de remitente de tal forma que el correo parezca ser originado por otro usuario o dirección en esa maquina, sendmail permite especificar en que usuarios del sistema puede confiar para que tengan los permisos para hacer esto.

La opción *use_ct_file* habilita la posibilidad de especificar el archivo que contenga los nombres de los usuarios de confianza. Por omisión, un pequeño grupo de usuarios del sistema son de confianza de sendmail. El nombre del archivo y su ubicación es */etc/mail/trusted_user*.

Para habilitar esta opción se necesita agregar *FEATURE(use_ct_file)* en el archivo *sendmail.mc*.

4.3.6 Usando un Smart Host

En ocasiones un host encuentra correo electrónico que no puede alcanzar directamente a un sitio remoto. Es conveniente tener un único sitio en la red que tenga la función de llevar a cabo la transmisión del correo a sitios remotos que son difíciles de alcanzar, en vez de que cada sitio local intente hacer esto por si solo.

Si se tiene un host inteligente, se puede enviar correo de cualquier tipo y él se encargara de realizar el encaminamiento y la transmisión de ese correo a todos los sitios remotos deseados.

Otra buena aplicación para la configuración de hosts inteligentes es realizar la transmisión a través de un cortafuegos privado. El host inteligente que se ejecute en el cortafuego será capaz de establecer conexiones directas de red con los sitios que se encuentran tanto en la red privada como en el exterior de ella. Puede también aceptar correo de ambos anfitriones de los que están dentro de la red privada como fuera de ella, el correo se guarda en un almacenamiento local y luego se hace la transmisión de ese mensaje directamente al sitio adecuado.
Sendmail posee un método simple para configurar un host inteligente, utilizando la opción *SMART_HOST* en el archivo de configuración *sendmail.mc*.

define(`SMART_HOST', `mail.isp.net')

4.3.7 Listas negras en tiempo real

Las listas de excepción en tiempo real (RBL, *Real-time Blackhole List*), es una lista pública que ayuda a reducir el volumen de correo no solicitado. Algunas fuentes de correo electrónico están enlistadas en una base de datos que se puede consultar en la siguiente dirección de Internet http://rbls.org.

Si se habilita esta opción de sendmail, se buscará la dirección de origen de cada mensaje que llegue en la base de datos de la lista negra en tiempo real para determinar si se acepta o no el mensaje, esto puede ayudar si se tiene un gran sitio con muchos usuarios ya que se ahorra una gran cantidad de espacio en disco.

Para configurar la opción de "listas negras en tiempo real", se necesita

agregar la declaración siguiente en el archivo *sendmail.mc*:

FEATURE(rbl)

Si se quiere especificar otro servidor *RBL*, la declaración debe ser:

FEATURE(rbl, 'rbl.host.net')

4.3.8 Base de datos de acceso

La base de datos de acceso permite configurar qué hosts o usuarios serán aceptados para enviar correo y quienes pueden utilizarlo como puente.

Gestionar a quiénes se les permitirá reenviar el correo es muy importante ya que el reenvío es una técnica de uso común para mandar correo basura a los hosts que tienen sistemas como el RBL. En vez de enviar el correo directamente, los spammers utilizaran el reenvió a través de un hosts que lo permita, para garantizar que el host no sea utilizado de esa forma, sólo se debe reenviar el correo de los sitios autorizados.

Cuando se recibe una conexión de entrada por *SMTP*, sendmail toma la información del encabezado de entrada y luego consulta la base de datos de acceso para ver si aceptara el contenido del mensaje.

La base de datos de acceso es una colección de reglas que describen que acciones se deben tomar para los mensajes recibidos de los hosts nombrados. El archivo de control de acceso se llama */etc/mail/access*. Cada línea de la tabla contiene una regla de acceso. El lado izquierdo de cada regla es un patrón utilizado para comparar con el remitente de un mensaje de correo de entrada. Puede ser una dirección de correo completa, un nombre de host o una dirección IP. El lado derecho es la acción que se

deberá tomar. Las acciones que se pueden tomar son:

- ***OK:*** Aceptar el mensaje.
- ***RELAY:*** Aceptar los mensajes para este hosts o usuario aún si no provienen del hosts local, es decir, aceptar que los mensajes sean reenviados hacia otros hosts desde este.
- ***REJECT:*** Rechazar el correo con un mensaje genérico.
- ***DISCARD:*** Descartar el mensaje utilizando la propiedad $#discard del sistema de correo.
- ***### cualquier texto:*** Contestar con un mensaje de error utilizando ### como código de error y cualquier texto será el mensaje.

Un ejemplo de un archivo */etc/mail/access*:

amigo@yahoo.com	*REJECT*
aol.com	*REJECT*
postmaster@aol.com	*OK*
linux.org.mx	*RELAY*

Se rechazaran todos los correos de *amigo@yahoo.com* y de cualquier usuario con el dominio *aol.com* pero se aceptara el de *postmaster@aol.com*, así como se permite la entrega de todos los correos del dominio *linux.org.mx*.

Para habilitar la opción de la base de datos de acceso, se debe de utilizar la siguiente declaración en el archivo *sendmail.mc*:

FEATURE(access_db)

4.4 Instalación de Clam Antivirus

ClamAV es el antivirus que estará interactuando con sendmail para evitar tanto el envío como la recepción de virus en el buzón de entrada del correo electrónico de los usuarios del sistema.

Existen diferentes maneras de instalar software en Linux, pero una de las más fáciles es por medio de los paquetes rpm, a continuación se describen los paquetes necesarios para instalar ClamAV en el sistema:

- *clamav-x.y.z.i386.rpm*: Este es el paquete principal, contiene el antivirus.
- *clamav-milter-x.y.z.i386.rpm*: Este paquete es necesario por que ClamAV no puede comunicarse directamente con sendmail y este software se encargara de hacerlo.

Los paquetes de ClamAV pueden descargarse de su sitio oficial (*http://www.clamav.net*). Una vez obtenidos los paquetes se procede con la instalación ejecutando los siguientes comandos como root:

 # rpm –Uvh clamav-0.86.1-1.i386.rpm

 # rpm –Uvh clamav-milter-0.86.1-1.i386.rpm

La otra forma de instalar el software es por medio de los archivos fuente que pueden ser encontradas en el sitio *Web* de ClamAV[21] y los pasos a seguir se estudiaran en los siguientes puntos.

[21] http://www.clamav.net

4.4.1 Requerimientos

Cuando se instala Clam Antivirus desde las fuentes se deben satisfacer los siguientes requerimientos para poder compilarlos y crear los binarios para proceder con su instalación final.

- Paquetes zlib y zlib-devel[22]

Si no se tienen instaladas las librerías zlib pueden ser descargadas para su instalación desde su sitio *Web* en *http://www.zlib.net/.*

4.4.2 Compilación e instalación de ClamAV

Para instalar ClamAV es necesario crear un nuevo usuario y grupo llamado clamav. Este usuario se requiere para poder ejecutar el servicio después de que el antivirus quede instalado en el sistema. Para crear el usuario y grupo se ejecutan los siguientes comandos:

 # groupadd clamav

 # useradd -g clamav -s /bin/false -c "Clam Antivirus" clamav

Una vez creado el usuario y el grupo clamav se procede a descomprimir el código fuente de ClamAV:

$ tar xvzf clamav-x.y.z.tar.gz

$ cd clamav-x.y.z

Asumiendo que se quieren instalar los archivos de configuración en el directorio */etc* se procede a ejecutar:

---

[22] http://www.zlib.net/

$ ./configure --sysconfdir=/etc --enable-milter

$ make

$ su -c "make install"

Ahora ya se tiene instalado el antivirus en el sistema, para probar que el antivirus funciona simplemente se ejecuta el comando:

$ clamscan nombre_archivo

Donde *nombre_archivo* es el nombre de cualquier archivo existente en el sistema, por ejemplo se escanea un archivo de prueba llamado *somefool.dat* el cual tiene una firma de un virus conocido:

$ clamscan somefool.dat

El resultado del escaneo será el siguiente:

somefool.dat: Worm.SomeFool.Gen-1 FOUND

---------- SCAN SUMMARY ----------

Known viruses: 37150

Engine version: 0.86.1

Scanned directories: 0

Scanned files: 1

Infected files: 1

Data scanned: 0.02 MB

Time: 5.237 sec (0 m 5 s)

Para ver más opciones de escaneo de clamscan se consulta el manual de la siguiente manera: *$ man clamscan*

4.4.3 Integración de ClamAV con Sendmail

Para que ClamAV y *sendmail* puedan comunicarse simplemente se tiene que ingresar las siguientes líneas en el archivo de configuración *sendmail.mc*:

define(`MILTER', 1)dnl

INPUT_MAIL_FILTER(`clamav', `S=local:/var/run/clamav/clamd.sock, F=, T=S:4m;R:4m')dnl

define(`confINPUT_MAIL_FILTERS', `clamav')

Y se regenera el archivo *sendmail.cf*.

También se tiene que verificar que la siguiente línea este habilitada en el archivo de configuración de ClamAV */etc/clamd.conf*.

LocalSocket /var/run/clamav/clamd.sock

Ahora solo se reinicia el servicio de sendmail para que los cambios surjan efecto.

4.4.4 Instalación y configuración de Spamassassin

La instalación de spamassassin se puede hacer cuando se instala la distribución de Linux que se vaya a usar ya que al ser uno de los mejores filtros antispam es incluido en todas las distribuciones actuales de Linux.

Para la instalación de software por medio de Internet se utiliza el depósito oficial de la distribución de *Fedora Core 3* en la siguiente dirección http://download.fedora.redhat.com/pub/fedora/linux/core/3/SRPMS/.

La configuración de spamassassin puede ser llevada acabo de manera global afectando todos los buzones/usuarios del sistema o bien, de manera individual donde cada usuario define reglas de filtrado más estrictas o flexibles.

Los parámetros globales de spamassassin son definidos en un archivo llamado */etc/mail/spamassassin/local.cf*, dicho archivo contiene las reglas que serán aplicadas a cualquier buzón que utilice spamassassin.

Para aquellos casos en los que un usuario desee definir reglas de filtrado especificas, éstas pueden ser definidas en un subdirectorio llamado *.spamassassin* que se encuentra en el */home/* del usuario y dentro de este existe un archivo denominado *user_prefs*, cabe mencionar que estas reglas son aplicadas una vez que han sido empleadas todas aquellas definidas a nivel global.

Cada regla en spamassassin posee un puntaje, valor que en caso de violarse dicha regla, es asignado al puntaje total del mensaje en la evaluación de ser *spam*, el valor promedio para que un correo electrónico sea considerado basura también es configurable, para efectos prácticos, spamassassin posee puntajes predefinidos "*default*" para todas sus reglas, mismas que pueden ser modificadas.

Aunque la nomenclatura utilizada para definir reglas es intuitiva, spamassassin posee un gran número de variantes, por lo que las siguientes normas sólo representan las más básicas para un filtrado elemental, por ejemplo se pueden agregar las siguientes reglas en el archivo:

 # Si el correo contiene 75%-100% letras mayúsculas asignar 0 puntos

score UPPERCASE_75_100 0

 # Reescribir titulo del mensaje con "SPAM" en caso de cumplir puntaje

rewrite_subject 1

subject_tag SPAM

 # 10 puntos requeridos para que correo sea considerado SPAM y reescribir # titulo

required_hits 10.0

La primera declaración -- *score UPPER_75_100* indica una reasignación de puntaje a cero sobre aquellos correos que contengan entre el 75% y 100% de su cuerpo en letras mayúsculas, esto evita que al ser inspeccionados mensajes de este tipo su puntaje se eleve considerablemente.

La segunda sección indica que el titulo original *"Subject"* del mensaje sea modificado agregando el vocablo *"SPAM"* lo cual facilita su clasificación una vez que el correo sea descargado a una utilería en PC (*Outlook*, *Eudora*, *Mozilla*).

Finalmente, la definición *required_hits 10.0* indica que aquellos mensajes con un puntaje mayor a 10 les sea agregada la leyenda antes mencionada a su titulo "*Subject*".

Aunque en las definiciones anteriores sólo se declaro una regla de spamassassin, el puntaje de todo mensaje será evaluado en base a los valores predefinidos, esto lo obligará a llevar acabo ajustes constantemente sobre el proceso de filtrado, ya sea modificando el puntaje (*required_hits*) o cambiando los puntajes de reglas individualmente.

Para que spamassassin pueda interactuar con sendmail es necesario instalar *Procmail* que es un agente de entrega de transporte, este programa se encarga de procesar el correo electrónico que ingresa al sistema, es ampliamente usado en los sistemas Unix, típicamente es invocado por un agente de transporte de correo como es sendmail, esto hace que el proceso

de envió y recepción de correo se maneje por eventos.

Las operaciones comunes realizadas por *Procmail* incluyen la filtración y clasificación de correo electrónico en diversas carpetas según las palabras claves, "Para", "De", "Tema", y también invocara a spamassassin para el filtrado de *spam*.

Para instalar *Procmail* se ejecuta el comando: *# yum install procmail,* si se tiene una conexión a Internet, también se puede obtener desde su sitio *Web* en *http://www.procmail.org/.*

Finalmente se crea el archivo de configuración /etc/procmail para que invoque a spamassassin y realizar el filtrado de correo, las siguientes líneas deben ser agregadas a dicho archivo:

```
DROPPRIVS=yes
:0fw
| /usr/bin/spamassassin
:0
* ^X-Spam-Status: Yes
$HOME/spam
```

Con la línea * ^X-Spam-Status: Yes se asegura de que todo el correo entrante sea revisado por los algoritmos de spamassassin, y con la *$HOME/spam* simplemente es la ubicación del directorio en donde se almacenara el correo basura recibido, *$HOME* es el directorio personal de cada usuario del sistema.

Finalmente para que todas las modificaciones surjan efecto se tiene que reiniciar el servicio de sendmail.

4.4.5 Actualización de ClamAV y Spamassassin

ClamAV cuenta con un programa de actualización llamado *freshclam*. Este programa lo que hace es conectarse a los servidores oficiales de *ClamAV* para comprobar si existe una actualización de la base de datos de los virus, en caso afirmativo la descarga, comprueba su firma y la instala, los modos de ejecución son los siguientes:

- **Modo interactivo:** ejecutar el comando *freshclam* en una consola con privilegios de root.
- **Como demonio:** Para ejecutar *freshclam* como demonio lo único que se debe añadir es el parámetro *-d* seguido de la opción *-c X*, siendo *X* el número de comprobaciones al día que se quiera que revise si hay actualizaciones de la base de datos.

Para un mejor funcionamiento de las actualizaciones automaticas se puede poner un *script* de inicio en */etc/cron.daily/*, un ejemplo del *script* de actualización de *freshclam* seria:

```
LOG_FILE="/var/log/clamav/freshclam.log"
if [ ! -f "$LOG_FILE" ]; then
    touch "$LOG_FILE"
    chmod 644 "$LOG_FILE"
    chown clamav.clamav "$LOG_FILE"
fi

/usr/bin/freshclam \
    --quiet \
    --datadir="/var/clamav" \
    --log="$LOG_FILE" \
    --log-verbose \
    --daemon-notify="/etc/clamd.conf"
```

Para mantener actualizadas las reglas de spamassassin se puede instalar un *bash script* llamado RulesDuJour el cual se encargara de mantener las reglas al día.

La instalación del script es la siguiente:

- Obtener el *script* principal llamado *rules_du_jour* del sitio oficial *http://www.exit0.us/.*
- Copiar el script en algun directorio por ejemplo */usr/local/sbin* y modificar sus atributos con el comando: *chmod +x /usr/local/sbin/rules_du_jour.*
- Crear un archivo de configuración llamado *config* en el directorio */etc/rulesdujour* y se ponen las líneas siguientes:

  TRUSTED_RULESETS="TRIPWIRE SARE_ADULT SARE_OBFU0 SARE_OBFU1 SARE_URI0 SARE_URI1"

  SA_DIR="/etc/mail/spamassassin"

  MAIL_ADDRESS="root@localhost"

  SA_RESTART="killall -HUP spamd"

Con esto se puede correr el *script rules_du_jour* periódicamente para mantener al día las reglas de spamassassin, ejemplo: *# rules_du_jour.*

Existe una forma de hacer la actualización de las reglas automáticamente por medio del *cron* para esto se edita el archivo *crontab* situado en el directorio */etc/* y se agregan las siguientes líneas para hacer una actualización todos los dias a las 4:28 am:

```
# Obtiene las últimas reglas de spamAssassin, se ejecuta todos los dias a
# las 4:28 AM.
28 4 * * * /usr/sbin/rules_du_jour
```

Con lo anterior tanto el antivirus como el antispam estara actualizado para

evitar el filtrado de virus o spam en el servidor de correo electrónico.

4.5 Pruebas del sistema

Las siguientes pruebas son para comprobar que el sistema es confiable y el administrador puede estar seguro que el servidor de correo electrónico estará libre de virus y *spam*.

Las pruebas constan de las siguientes partes:

a) **Recibo de correo electrónico externo:** Se enviará correo con *spam* y con virus para comprobar que sendmail, ClamAV y spamassassin trabajan correctamente.
b) **Envió de correo desde sendmail:** Se enviarán correos con archivos infectados de virus para probar el nivel de seguridad que ofrece el sistema.

Para realizar las pruebas se implemento una pequeña red local con dos computadoras, una con *Linux Fedora Core 3* como sistema operativo con sendmail, ClamAV y spamassassin funcionando como *SMTP* principal, la otra tiene Windows XP como sistema operativo y solo tiene la función de cliente para hacer uso de las cuentas de correo existentes en sendmail usando como programa cliente *Outlook Express*, están conectadas a un ruteador que es el que da la salida hacia el Internet, en la figura 5 se muestra la distribución de la red local.

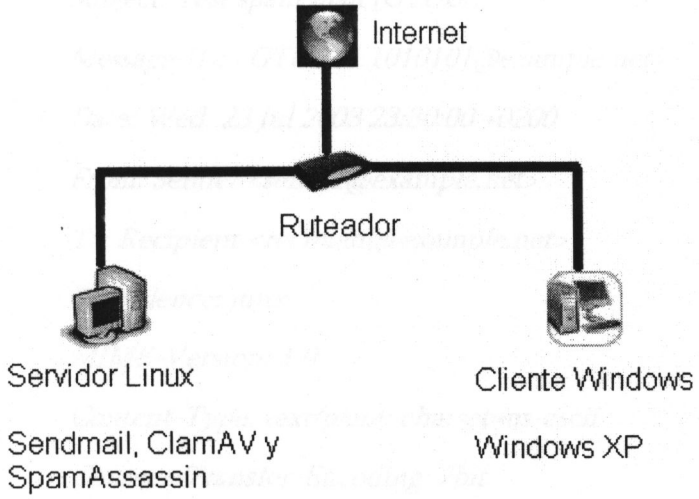

Figura 5 Red local

4.5.1 Correo de entrada

Para verificar que el sistema bloquea el *spam*, se realizaran las siguientes pruebas:

a) **Máquina local:** Se realiza una prueba desde la máquina local usando correos que son considerados como *spam* y como no *spam*, la prueba se realiza con el comando:

 # spamassassin –t correo > resultado

Donde correo es el archivo que contiene el correo tanto sus cabeceras como su contenido y resultado es el archivo donde se guardaran los resultados.

- **Prueba con spam:** Se le envía un documento que contiene un correo electrónico que es considerado como *spam*.

Subject: Test spam mail (GTUBE)

Message-ID: <GTUBE1.1010101@example.net>

Date: Wed, 23 Jul 2003 23:30:00 +0200

From: Sender <sender@example.net>

To: Recipient <recipient@example.net>

Precedence: junk

MIME-Version: 1.0

Content-Type: text/plain; charset=us-ascii

Content-Transfer-Encoding: 7bit

This is the GTUBE, the

Generic

Test for

Unsolicited

Bulk

Email

If your spam filter supports it, the GTUBE provides a test by which you can verify that the filter is installed correctly and is detecting incoming spam. You can send yourself a test mail containing the following string of characters (in upper case and with no white spaces and line breaks):

*XJS*C4JDBQADN1.NSBN3*2IDNEN*GTUBE-STANDARD-ANTI-UBE-TEST-EMAIL*C.34X*

Resultado: el correo es bloqueado agregando a las cabeceras lo siguiente:

Subject: [SPAM] Test spam mail (GTUBE)

X-Spam-Flag: YES

X-Spam-Checker-Version: SpamAssassin 3.0.0 (2004-09-13) on localhost

X-Spam-Level:

X-Spam-Status: Yes, score=997.2 required=2.0 tests=ALL_TRUSTED, DNS_FROM_AHBL_RHSBL,GTUBE autolearn=unavailable version=3.0.0

X-Spam-Report:

** -2.8 ALL_TRUSTED Did not pass through any untrusted hosts*

** 1000 GTUBE BODY: Generic Test for Unsolicited Bulk Email*

** 0.1 DNS_FROM_AHBL_RHSBL RBL: From: sender listed in dnsbl.ahbl.org*

- **Prueba sin spam:** Se envía un correo electrónico que será reconocido como no *spam*.

To: tbtf@world.std.com

From: Keith Dawson <dawson@world.std.com>

Subject: TBTF ping for 2001-04-20: Reviving

Content-Type: text/plain; charset="us-ascii"

Sender: tbtf-approval@world.std.com

Precedence: list

Reply-To: tbtf-approval@europe.std.com

-----BEGIN PGP SIGNED MESSAGE-----

TBTF ping for 2001-04-20: Reviving

Tasty Bits from the Technology Front

Timely news of the bellwethers in computer and communications technology that will affect electronic commerce -- since 1994

Your Host: Keith Dawson

Resultado: se obtienen las siguientes cabeceras, lo cual demuestra que el correo es aceptado normalmente y no es rechazado a pesar pudiera ser confundido como spam.

X-Spam-Checker-Version: SpamAssassin 3.0.0 (2004-09-13) on localhost

X-Spam-Status: No, score=0.0 required=2.0 tests=none autolearn=unavailable version=3.0.0

b) **Desde SMTP externo:** En esta prueba se usara un *SMTP* externo para el envió de correo electrónico hacia una usuario de sendmail.

- **Envió de spam:** en esta prueba el medio para enviar correos con *spam* será un *SMTP* externo, con esto se comprobara que el sistema sendmail con spamassassin es confiable.

  Para probar el bloqueo de *spam* se utilizo un banco de pruebas, un pequeño ejemplo seria el siguiente mensaje de correo:

Subject: Important Career Center Information

Message:

Campuscareercenter.com is the world's premier job and internship site!

Recruiting season has begun for Internships, Part time, and
Full Time opportunities. If you have not submitted your student profile or resume, please sign up immediately at:

http://www.campuscareercenter.com/register

Whether or not you have a resume, it is easy to create your student profile.

Although graduation may seem to be a long time away, the major recruiting process occurs NOW for all major companies and firms. Do not get left behind!

Please forward this message to any interested candidates.

www.campuscareercenter.com

If you have any questions or concerns please contact CCC at Concerns@CampusCareerCenter.com

We (CCC) are a privately held company based in ambridge, MA. While our goal is to partner with colleges and universities around the world to provide comprehensive internet career resources, we are not affiliated with MIT and encourage you to utilize your on-campus Career Service center.

Resultado: El correo fue detectado como *spam* y no es mostrado al usuario, las cabeceras del correo fueron as siguientes:

X-Spam-Prev-Subject: Important Career Center Information

X-Spam-Flag: YES

X-Spam-Checker-Version: SpamAssassin 3.0.0 (2004-09-13) on localhost

*X-Spam-Level: ***

X-Spam-Status: Yes, score=2.8 required=2.0

- **Envió de correo sin spam:** Usando un servidor de correo externo se envían mensajes al sistema local para probar que el correo normal no vaya a ser detectado como spam.

  Un ejemplo de un correo electrónico que pudiera ser considerado como correo basura seria el siguiente:

  Subject: Información sobre PLC

  Message:

  Hola que tal te envio la información sobre PLC que me pediste, ojala te sea de ayuda.

  PLC (Power Line Comunications), también denominada BPL (Broadband over Power Lines) es una tecnología basada en la transmisión de datos utilizando como infraestructura la red eléctrica.
  Esto implica la capacidad de ofrecer, mediante este medio, cualquier servicio basado en IP, como podría ser telefonía IP, Internet, videoconferencia, datos a alta velocidad, etc...

  Hay dos tipos principales de Power Line Communications:

- PLOC (Power Line Outdoors Telecoms o comunicaciones extrahogareñas utilizando la red eléctrica).
Esto es, la comunicación entre la subestación eléctrica y la red doméstica (electro-modem). El estándar es ETSI

- PLIC (Power Line Indoors Telecoms o comunicaciones intrahogareñas utilizando la red eléctrica).
Esto es, utilizando la red eléctrica interior de la casa, para establecer comunicaciones internas. Un ejemplo: PLIC es una de las vías utilizadas en domótica (otra que se suele utilizar también es la comunicación vía radio).

Resultado: A pesar de que el mensaje enviado pudiera haber sido considerado como *spam* fue entregado sin ningún problema, las cabeceras del mensaje son:

X-Spam-Checker-Version: SpamAssassin 3.0.0 (2004-09-13) on localhost
*X-Spam-Level: **
X-Spam-Status: No, score=1.4 required=2.0

c) **Recibiendo correo con archivo adjunto sin virus:** Con esta prueba se envía un correo electrónico con un archivo que no contiene virus para verificar si se recibe correctamente, para enviarlo se uso una cuenta de *gmail*.

Resultado: El correo electrónico fue recibido sin problemas en la cuenta local de sendmail, en la figura 5 se muestra el correo electrónico que se recibió con el archivo adjunto sin problemas.

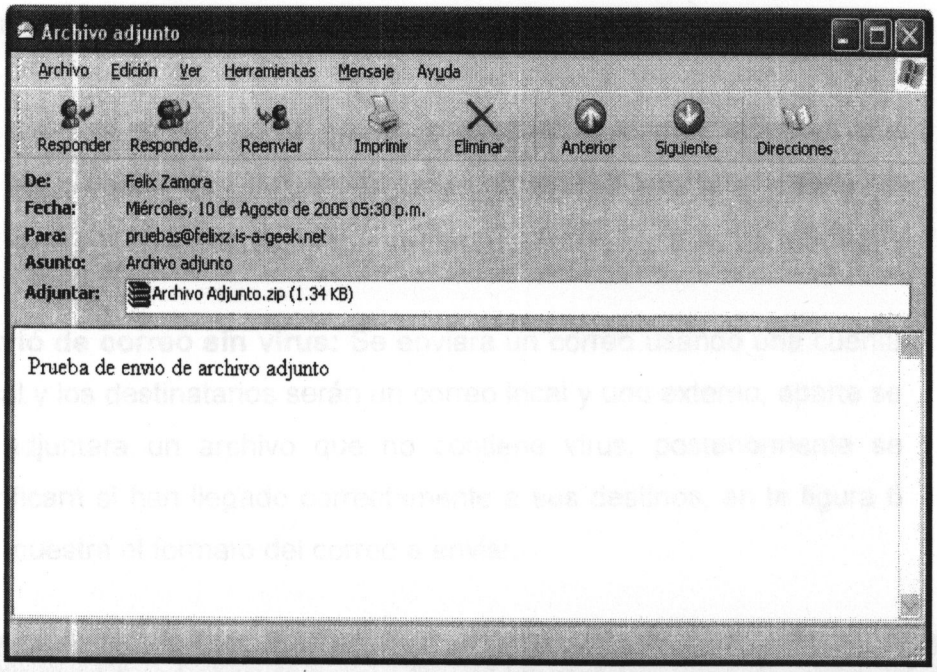

Figura 5 Correo con archivo adjunto

d) **Recibiendo correo con archivo adjunto con virus:** Sendmail esta configurado para escanear los archivos adjuntos con ClamAV en busca de virus en la siguiente prueba se envió un correo con un archivo adjunto que contiene el troyano subseven.

Resultado: El correo electrónico no fue entregado debido a que ClamAV detecto que el archivo que contenía estaba infectado de virus y lo regresa al servidor de origen, el log de sendmail muestra lo siguiente:

Milter add: header: X-Virus-Status: Infected with Trojan.SubSeven.21.A

Milter: data, reject=554 5.7.1 virus Trojan.SubSeven.21.A detected by ClamAV - http://www.clamav.net

4.5.2 Correo de salida

Sendmail tiene la capacidad de enviar correo electrónico, en este apartado se harán las pruebas de envió tanto para una cuenta local como para un correo de un dominio externo.

a) **Envió de correo sin virus:** Se enviara un correo usando una cuenta local y los destinatarios serán un correo local y uno externo, aparte se le adjuntara un archivo que no contiene virus, posteriormente se verificará si han llegado correctamente a sus destinos, en la figura 6 se muestra el formato del correo a enviar.

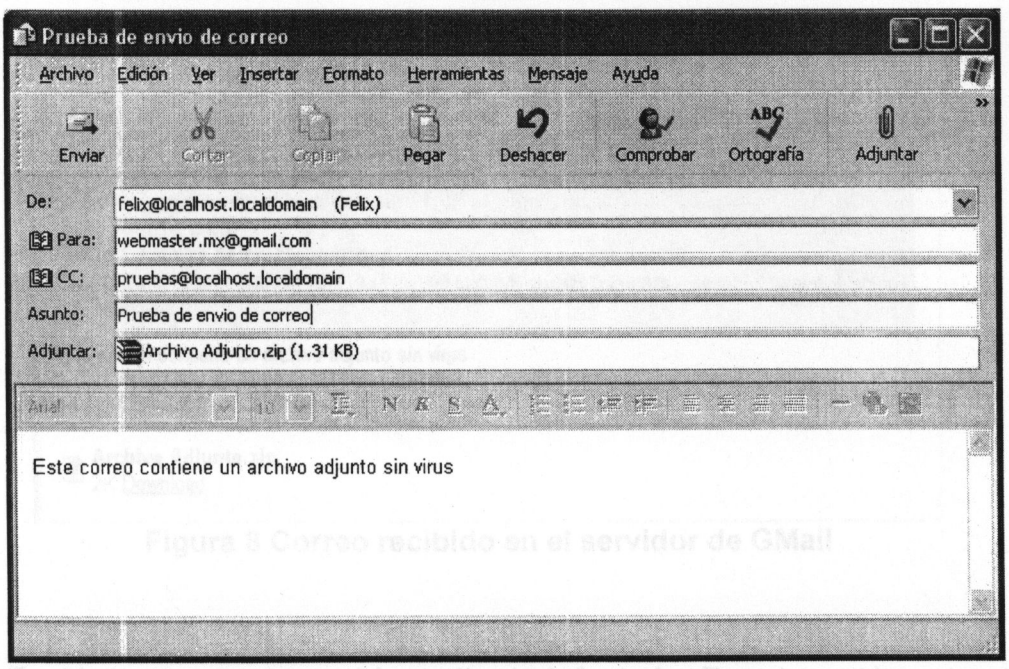

Figura 6 Envió de correo con archivo adjunto

Resultado: El correo fue entregado sin contratiempos en los respectivos buzones de entrada de cada cuenta, en la figura 7 se muestra el correo de la cuenta local y en la figura 8 se muestra el buzón de entrada de la cuenta de *gmail*.

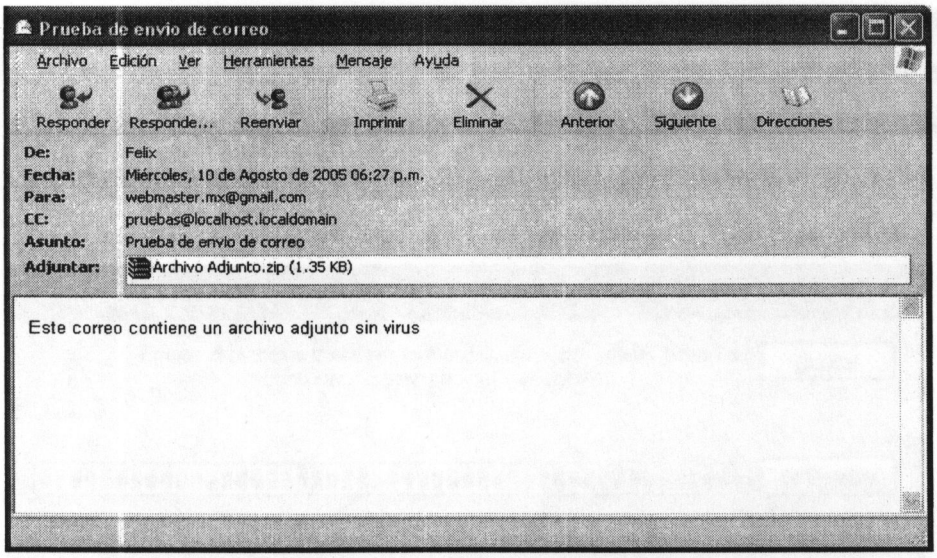

Figura 7 Correo recibido en cuenta local

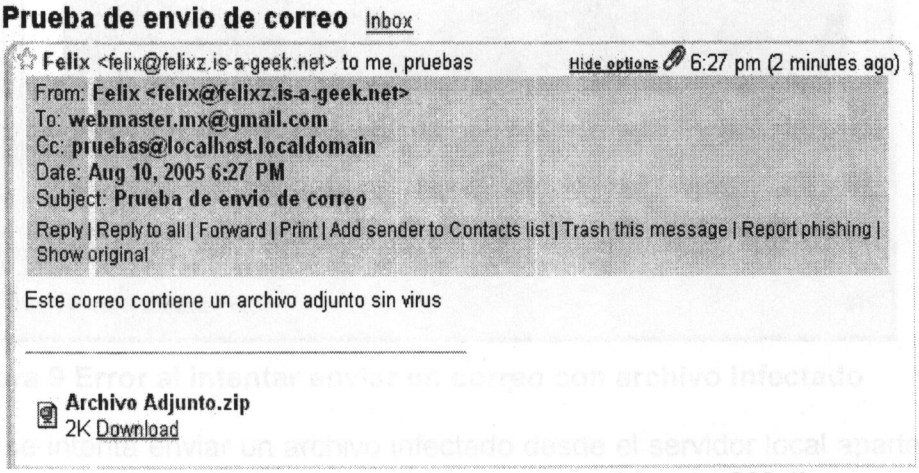

Figura 8 Correo recibido en el servidor de GMail

b) Envió de correo con archivo adjunto infectado: En esta prueba se envió un archivo adjunto infectado con un virus para verificar el comportamiento del antivirus y comprobar si detecta los archivos infectados al tratar de enviar un correo desde alguna cuenta local, se intentara enviarlo a un correo externo.

Resultado: El correo no pudo ser entregado debido a que ClamAV detecto un virus en el archivo adjunto y sendmail no permite que el

correo sea entregado al remitente, mostrando un error en software que usa el cliente informando el por que no pudo ser entregado el correo, en la figura 9 se muestra la ventana de error que muestra *Outlook Express.*

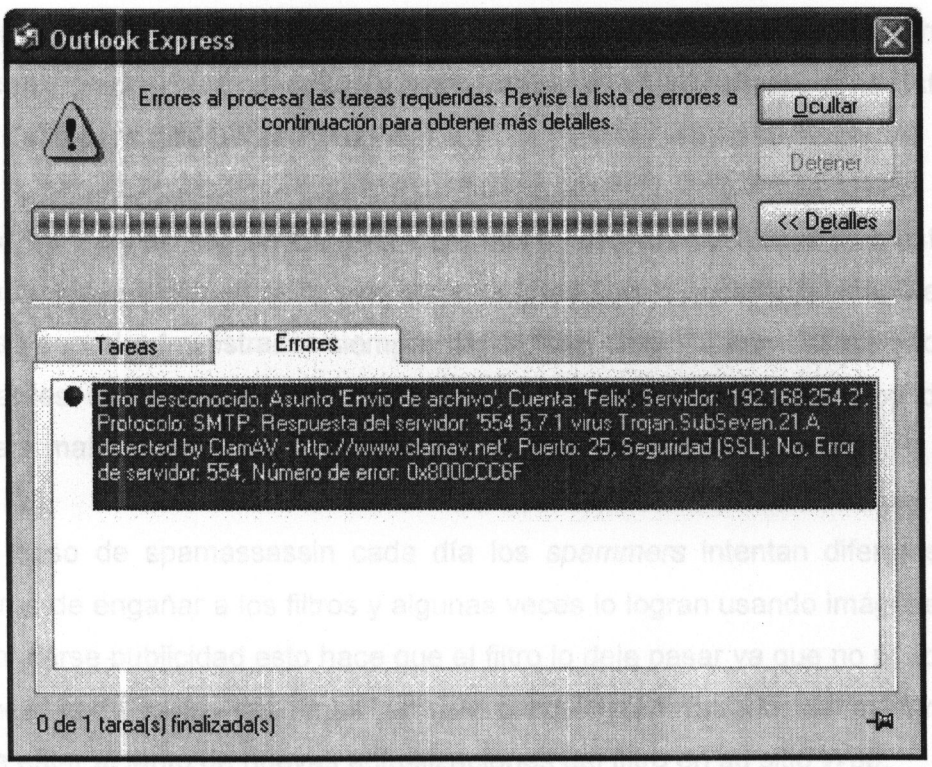

Figura 9 Error al intentar enviar un correo con archivo infectado

Cuando se intenta enviar un archivo infectado desde el servidor local aparte de bloquearlo y no dejarlo salir, envía un correo electrónico al administrador del sistema informándole que se intento enviar virus desde el servidor. El contenido del correo de aviso es el siguiente:

Subject: Virus intercepted

A message you sent to
<webmaster.mx@gmail.com>
contained Trojan.SubSeven.21.A and has not been delivered.

La dirección electrónica que se muestra es a donde se iba enviar el correo electrónico.

4.5.3 Posibles problemas

Ningún sistema es confiable al cien por ciento, a pesar de que sendmail ha sido configurado con uno de los mejores antivirus y filtro antispam no esta descartado para fallar alguna vez.

Una posible causa de que ClamAV deje entrar un virus puede ser que este sea muy nuevo y aun no se cuente con una firma para detectarlo a tiempo es por eso que el administrador siempre debe estar atento a los avisos de los fabricantes de antivirus para estar al día en noticias sobre este tipo de software mal intencionado.

En el caso de spamassassin cada día los *spammers* intentan diferentes maneras de engañar a los filtros y algunas veces lo logran usando imágenes para hacerse publicidad esto hace que el filtro lo deje pasar ya que no puede escanear texto en las imágenes, en este caso el administrador del sistema puede estar al tanto de nuevas actualizaciones del filtro en su sitio *Web*.

Aun con las ventajas que ofrece el servidor de correo con antivirus y antispam nunca esta de mas que el usuario tenga instalado en su máquina un antivirus local y si así lo desea un filtro antispam adicional para el software que utilice para leer su correo electrónico.

4.5.4 Logs del Sendmail

Los logs de sendmail sirven para llevar una bitácora del comportamiento del sistema como por ejemplo quien se ha logeado, el envió y recepción de correo, también informa sobre los escaneos que hace ClamAV a los correos

tanto de entrada como de salida y de spamassassin grabando también cuando se ejecuta para verificar si hay o no *spam* en los mensajes.

Esto puede ser útil sobre todo si el servidor se cae, el administrador puede leer a que hora sucedió y las posibles causas que lo generaron encontrando a veces algunos fallos del propio software que se pueden solucionar aplicando los parches de actualización.

En *Fedora Core 3* los logs de sendmail los contiene un archivo llamado *maillog* y se encuentra en:

 # /var/logs/maillog

Al ser un archivo de texto plano se puede abrir con cualquier editor de textos para darle lectura.

CONCLUSIONES Y TRABAJO A FUTURO

El Internet cada día es mas usado en el mundo y con el los servicios que se ofrecen de distinta forma ya sea pagando o gratis como el correo electrónico que es sin duda uno de los servicios mas usados en toda la red.

Al ser el correo electrónico muy solicitado se presta a que muchas empresas hagan publicidad por este medio creando así el molesto *spam* o correo basura, es por esta sencilla razón que en este trabajo de investigación se propone a los administradores de correo que usen sendmail como servidor *SMTP* por el buen rendimiento que ofrece.

El filtrado de correo es para evitar que los usuarios del servicio de correo electrónico tengan que estar eliminando el correo indeseado causando la perdida de tiempo y recursos, se uso como filtro spamassassin que es el encargado de desechar el correo basura de una forma transparente para el usuario final.

Existe el riesgo de recibir virus que se propagan por medio del correo electrónico, para esto se integro Clam Antivirus para frenar estas amenazas de la red.

Aunque en Linux no son muy comunes los virus ya que en su mayoría están programados para la plataforma de Windows en cualquiera de sus versiones no esta por demás tener instalado un antivirus para frenar los ataques que pudieran ocasionar cualquier daño en el sistema, además como en este caso en el cual se monto un servidor de correo el cual no va a ser accesado solamente por usuarios Linux sino que desde cualquier plataforma se puede configurar los programas clientes de correo electrónico para enviar y recibir correo fue la razón por la cual se decidió implementar el antivirus junto con el servidor sendmail, así no se pone en peligro las maquinas de los usuarios y se tiene la seguridad de que se bloqueara cualquier código malicioso.

Una recomendación para un administrador de un servidor de correo es que siempre debe estar al día sobre posibles fallos de seguridad y mantener la base de datos de los virus actualizada por que día a día se estrenan nuevos virus en la red.

En un futuro este trabajo puede ser complementado desarrollando una interfaz web para acceder al servidor de correo electrónico tomando en cuenta las siguientes sugerencias:

1. Una interfaz *Web* para que el correo electrónico pueda ser leido por medio de un navegador y no solo por programas como *Outlook*, con esta mejora el sistema se hará más accesible con el usuario final y la opción de administrar su propia cuenta por medio de un panel de control.
2. Creación de cuentas en linea, esta caracteristica ofrecera la posibilidad de dar de alta cuentas por medio del acceso *Web* que se trato en el punto anterior.
3. Un sistema de autorespuesta, cuando el servidor reciba un correo proveniente de una dirección de correo electrónico que se encuentre en la lista de contactos del usuario, le envíe un mensaje avisando que se recibio el correo satisfactoriamente.

Con los puntos anteriores el alcance del servicio podría llegar a ser de uso público y no solo de uso privado como fue desarrollado en este trabajo.

APENDICE A

1) Archivo de configuración de Sendmail /etc/mail/sendmail.mc

```
divert(-1)dnl
include(`/usr/share/sendmail-cf/m4/cf.m4')dnl
VERSIONID(`setup for Red Hat Linux')dnl
OSTYPE(`linux')dnl
dnl define(`SMART_HOST', `smtp.your.provider')
define(`SMART_HOST',`felixz.is-a-geek.net')
define(`confDEF_USER_ID',``8:12'')dnl
define(`confTO_CONNECT', `1m')dnl
define(`confTRY_NULL_MX_LIST',true)dnl
define(`confDONT_PROBE_INTERFACES',true)dnl
define(`PROCMAIL_MAILER_PATH',`/usr/bin/procmail')dnl
define(`ALIAS_FILE', `/etc/aliases')dnl
define(`STATUS_FILE', `/var/log/mail/statistics')dnl
define(`UUCP_MAILER_MAX', `2000000')dnl
define(`confUSERDB_SPEC', `/etc/mail/userdb.db')dnl
define(`confPRIVACY_FLAGS', `authwarnings,novrfy,noexpn,restrictqrun')dnl
define(`confAUTH_OPTIONS', `A')dnl
TRUST_AUTH_MECH(`EXTERNAL DIGEST-MD5 CRAM-MD5 LOGIN
PLAIN')dnl
define(`confTO_IDENT', `0')dnl
FEATURE(`no_default_msa',`dnl')dnl
FEATURE(`smrsh',`/usr/sbin/smrsh')dnl
FEATURE(`mailertable',`hash -o /etc/mail/mailertable.db')dnl
FEATURE(`virtusertable',`hash -o /etc/mail/virtusertable.db')dnl
FEATURE(redirect)dnl
FEATURE(always_add_domain)dnl
FEATURE(use_cw_file)dnl
FEATURE(use_ct_file)dnl
FEATURE(local_procmail,`',`procmail -t -Y -a $h -d $u')dnl
FEATURE(`access_db',`hash -T<TMPF> -o /etc/mail/access.db')dnl
```

FEATURE(`blacklist_recipients')dnl

EXPOSED_USER(`root')dnl

LOCAL_DOMAIN(`localhost.localdomain')dnl

dnl MASQUERADE_AS(`mydomain.com')dnl

MASQUERADE_AS(`felixz.is-a-geek.net')dnl

FEATURE(masquerade_envelope)dnl

FEATURE(masquerade_entire_domain)dnl

FEATURE(`dnsbl', `relays.ordb.org', `Rechazado - ver http://ordb.org/')dnl

define(`MILTER', 1)dnl

INPUT_MAIL_FILTER(`clamav', `S=local:/var/run/clamav/clamd.sock, F=, T=S:4m;R:4m')dnl

define(`confINPUT_MAIL_FILTERS', `clamav')

MAILER(smtp)dnl

MAILER(procmail)dnl

Lineas modificadas:

define(`SMART_HOST',`felixz.is-a-geek.net')

Con la linea anterior se especifica el smart host el cual sirve para resolver los DNS y sea posible el envio de correo hacia SMTP exteriores.

MASQUERADE_AS(`felixz.is-a-geek.net')dnl

Esta sirve para enmascarar el dominio de salida y el correo aparezca como usuario@felixz.is-a-geek.net y no como usuario@localhost.localdomain, aparte que si es enviado hacia un SMTP como Hotmail o Gmail puedan responder el correo sin problemas de que vaya a ser rechazado por no encontrar el dominio.

define(`MILTER', 1)dnl

INPUT_MAIL_FILTER(`clamav', `S=local:/var/run/clamav/clamd.sock, F=, T=S:4m;R:4m')dnl

define(`confINPUT_MAIL_FILTERS', `clamav')

Con las líneas anteriores se le indica a sendmail la forma en como va a interactuar con el antivirus.

2) Archivo de configuración de ClamAV /etc/clamd.conf

LogFile must be writable for the user running daemon.
LogFile /var/log/clamav/clamd.log
Maximal size of the log file.
Value of 0 disables the limit.
You may use 'M' or 'm' for megabytes (1M = 1m = 1048576 bytes)
and 'K' or 'k' for kilobytes (1K = 1k = 1024 bytes). To specify the size
in bytes just don't use modifiers.
Default: 1M
LogFileMaxSize 0
Log time with each message.
Default: disabled
LogTime
Use system logger (can work together with LogFile).
Default: disabled
LogSyslog
This option allows you to save a process identifier of the listening
daemon (main thread).
Default: disabled
PidFile /var/run/clamav/clamd.pid
Optional path to the global temporary directory.
Default: system specific (usually /tmp or /var/tmp).
TemporaryDirectory /var/tmp
Path to the database directory.

```
# Default: hardcoded (depends on installation options)
DatabaseDirectory /var/clamav
# The daemon works in a local OR a network mode. Due to security reasons
# we recommend the local mode.

# Path to a local socket file the daemon will listen on.
# Default: disabled
LocalSocket /var/run/clamav/clamd.sock
# Remove stale socket after unclean shutdown.
# Default: disabled
FixStaleSocket
# Maximum length the queue of pending connections may grow to.
# Default: 15
MaxConnectionQueueLength 30
# Maximal number of threads running at the same time.
# Default: 10
MaxThreads 20
# Waiting for data from a client socket will timeout after this time (seconds).
# Value of 0 disables the timeout.
# Default: 120
ReadTimeout 300
# Perform internal sanity check (database integrity and freshness).
# Default: 1800 (30 min)
SelfCheck 3600
# Run as a selected user (clamd must be started by root).
# Default: disabled
User clamav
# Initialize supplementary group access (clamd must be started by root).
# Default: disabled
AllowSupplementaryGroups
# PE stands for Portable Executable - it's an executable file format used
# in all 32-bit versions of Windows operating systems. This option allows
```

```
# ClamAV to perform a deeper analysis of executable files and it's also
# required for decompression of popular executable packers such as UPX,
# FSG, and Petite.
# Default: enabled
ScanPE
# This option enables scanning of Microsoft Office document macros.
# Default: enabled
ScanOLE2
# Enable internal e-mail scanner.
# Default: enabled
ScanMail
# Perform HTML normalisation and decryption of MS Script Encoder code.
# Default: enabled
ScanHTML
# ClamAV can scan within archives and compressed files.
# Default: enabled
ScanArchive
# Files in archives larger than this limit won't be scanned.
# Value of 0 disables the limit.
# Default: 10M
ArchiveMaxFileSize 10M
# Nested archives are scanned recursively, e.g. if a Zip archive contains a
# RAR file, all files within it will also be scanned. This options specifies how
# deep the process should be continued.
# Value of 0 disables the limit.
# Default: 8
ArchiveMaxRecursion 8
# Number of files to be scanned within an archive.
# Value of 0 disables the limit.
# Default: 1000
ArchiveMaxFiles 1000
ArchiveMaxCompressionRatio 300
```

Mark encrypted archives as viruses (Encrypted.Zip, Encrypted.RAR).
Default: disabled
ArchiveBlockEncrypted
ArchiveBlockMax

Lineas modificadas:

LocalSocket /var/run/clamav/clamd.sock

Se le indica el *path* del *socket* local que se usara para activar ClamAV

3) Archivo de configuración local /etc/rc.local

!/bin/sh
This script will be executed *after* all the other init scripts.
You can put your own initialization stuff in here if you don't
want to do the full Sys V style init stuff.

touch /var/lock/subsys/local
clamav-milter -blo /var/run/clamav/clamd.sock

Solo se agrego la ultima linea para crear activar el *clamav-milt*er y sendmail pueda mandarlo llamar cuando sea necesario.

3) Archivo de configuración de SpamAssassin
/etc/mail/spamassassin/local.cf

required_hits 5
report_safe 0
rewrite_header Subject [SPAM]
required_score 2
ok_languages es
En este archivo solo se modifico el score y se le puso un valor de 2.

BIBLIOGRAFÍA

Sendmail Milters A Guide for Fighting Spam.
Costales B., Flynt M. (2005).
Addison Wesley.
Sendmail 8.13 Companion.
Assmann C., Costales B., Jansen G., Shapiro G. (2004).
OReilly.
SpamAssassin.
Schwartz A. (2004).
O'Reilly.
Clam Antivirus 0.86.1 User Manual.
Clam Antivirus.

Referencias

Advanced Spam Protection
Sendmail, Inc.
http://www.sendmail.com/products/mailstream/antispam/ (Octubre 2005)
What's New in SpamAssassin 3.0
Alan Schwartz
http://www.onlamp.com/pub/a/onlamp/2004/09/09/spamassassin.html
(Octubre 2005)
Eugenio Siccardi
http://www.rompecadenas.com.ar/ (Octubre 2005)
Bombardeo spam electrónico
http://www.mipunto.com/temas/3er_trimestre03/spam.html (Octubre 2005)
Correo Basura
Gonzalo Álvarez Marañó
http://www.iec.csic.es/criptonomicon/spam/ (Octubre 2005)
Cómo configurar Sendmail y Fetchmail para intranets y redes caseras.
Autor: Joel Barrios Dueñas
http://www.linuxparatodos.net/linux/como-sendmail-fetchmail.php
(Octubre 2005)

SpamAssassin-ClamAV-Procmail-Howto
Falko Timme
http://www.falkotimme.com/howtos/spamassassin_clamav_procmail/index.php
(Octubre 2005)
ClamAV as a sendmail milter
G. Stewart
http://linux.sgms-centre.com/howto/sendmailclamav.php (Octubre 2005)
Clam AntiVirus Milter Setup and Debugging
Sial
http://sial.org/howto/clamav/clamav-milter/
Clam AntiVirus with Sendmail on Fedora Core 1
Ron Goulard
http://fedoranews.org/contributors/ron_goulard/clamav/
(Octubre 2005)
Sendmail + SpamAassassin + Clam AV en Gentoo GNU/Linux
Ivan Belmonte
http://www.assl-site.net/docs/docs/sendmail-spamd-clamd.html
(Octubre 2005)
Sendmail
Pello Xabier Altadill Izura
http://www.pello.info/guias/boletin-001.html
(Octubre 2005)
Mail Server Filtering
Michael W. Lucas
http://www.onlamp.com/pub/a/bsd/2004/04/01/Big_Scary_Daemons.html
(Octubre 2005)