SOLUCIÓN PARA EL MEJORAMIENTO DEL SERVICIO DE CORREO ELECTRÓNICO EN LA RED DE DATOS DE LA UNIVERSIDAD DEL CAUCA.

Trabajo de desarrollo

DAVID FERNANDO ANDRADE SOLANO
VICTOR ANDRES CASTRO DUEÑAS

ANEXO E
CONFIGURACIÓN DE HERRAMIENTAS ANTIVIRUS Y ANTISPAM

Director: Guefry Agredo Méndez M.Sc.

UNIVERSIDAD DEL CAUCA
FACULTAD DE INGENIERÍA ELECTRÓNICA Y TELECOMUNICACIONES
DEPARTAMENTO DE TELECOMUNICACIONES
Grupo I+D Nuevas Tecnologías en Telecomunicaciones - GNTT
POPAYÁN, 2012

TABLA DE CONTENIDO

LISTA DE FIGURAS

ANEXO E

CONFIGURACIÓN DE HERRAMIENTAS ANTIVIRUS Y ANTISPAM

Introducción

Para la verificación de la procedencia y contenido de los mensajes de correo electrónico, se necesitan de diversas herramientas que cumplan con dichas funciones, estas herramientas son los antivirus y los filtros *antispam*. Este anexo contiene las configuraciones realizadas a las diferentes herramientas que han sido escogidas con anterioridad.

1. Configuración de filtros AntisPam y AntiVirus en Postfix.

Los filtros de antivirus y *antispam* se encuentran separados en dos categorías, filtros de reputación y filtros de contenidos. La función de los filtros de reputación es evitar que lleguen mensajes *spam* y virus al servidor de correo, verificando la procedencia del mensaje y los filtros de contenidos consisten en analizar los mensajes que llegan al servidor para determinar si son mensajes *spam* o si tienen virus.

1.1 Configuración de filtros de reputación en Postfix.

Para trabajar con algunas restricciones es bueno activar en **Postfix** el análisis del formato y protocolo del mensaje, estos comandos se activan antes de trabajar con las restricciones.

```
smtpd_helo_required=yes
smtpd_rfc821_envelopes=yes
```

Los filtros de reputación de **Postfix** se dividen en cinco listas o niveles en que se ejecutan dichos filtros [1], estas listas se presentan a continuación:

- **smtpd_client_restrictions:** este parámetro determina qué clientes se deben aceptar.
- **smtpd_helo_restrictions:** restringe aquellos host que puedan enviar el comando HELO/EHLO.
- **smtpd_sender_restrictions:** restringe aquellas direcciones que el sistema acepta cuando se acepta el comando MAIL FROM.
- **smtpd_recipient_restrictions:** restringe a los clientes cuando se acepta el comando RCPT TO.
- **smtpd_data_restrictions:** restringe la conexión cuando se acepta el comando DATA.

Estas listas se ejecutan en forma secuencial y pueden trabajar con varias restricciones, el orden en que se ejecutan las restricciones también son secuenciales.

Los filtros de reputación se configuran en el servidor de correo **Postfix** en la ubicación `/etc/postfix/main.cf` [2], el primer filtro de reputación es el filtro de listas negras RBL, estas listas indican que servidores son los que envía *spam,* por lo tanto si el servidor se encuentra en estas listas la conexión será rechazada. La configuración para que **Postfix** trabaje con listas negras se hace de la siguiente forma:

Primero que todo, en `/etc/postfix/main.cf` de **Postfix** se pone la directiva `smtpd_recipient_restrictions=,` después de esta se coloca en que listas negras se va a hacer la consulta con el comando `reject_rbl_client dirección del servidor de listas negras,` esta verificará en la lista negra la resolución inversa de la dirección IP en de quien esta solicitando conexión y `reject_rhsbl_client reject_rhsbl_client,` verificará el nombre del servidor que pide conexión. La configuración quedará de la siguiente forma:

```
smtpd_recipient_restrictions=
                      reject_rbl_client zen.spamhaus.org
                      reject_rhsbl_client zen.spamhaus.org
                      reject_rbl_client bl.spamcop.net
                      reject_rhsbl_client bl.spamcop.net
```

Las direcciones `zen.spamhaus.org` y `bl.spamcop.net`, son las listas negras que tienen mejor reputación en cuanto a fidelidad y actualización, aunque no son las únicas que existen.

El segundo filtro de reputación que se configura en **Postfix**, es el filtro de resolución inversa en el DNS, esto quiere decir que se toma la dirección IP del remitente de correo y se verifica su autenticidad, como por ejemplo, si se recibe una solicitud de conexión de un dominio "mail.dominio.org" que tiene la dirección Ip "134.45.56.8", entonces el sistema de correo le pregunta al DNS por dicha dirección 8.56.45.134 in-addr.arpa, si el DNS retorna "mail.dominio.org" se puede decir que es una petición valida. La activación de este filtro se realiza con el siguiente comando:

```
smtpd_recipient_restrictions= reject_unknown_sender_domain
```

También existe otro tipo de restricciones que es importante tener en cuenta para que el sistema de correo pueda evitar el *spam* de manera más efectiva, estas se enuncian a continuación [1]:

- **reject_invalid_hostname**: rechaza la petición cuando HELO/EHLO es enviado con una mala sintaxis de host o un nombre de dominio completamente calificado *(FQDN, Fully Qualified Domain Name*) inválido.

- **reject_non_fqdn_helo_hostname**: rechaza la conexión cuando el hostname en el comando HELO/EHLO no esta de la forma especificada en el RFC821 o es un FQDN que no existe.

- **reject_non_fqdn_sender**: rechaza la solicitud cuando la dirección en MAIL FROM no cumple con FQDN

- **reject_non_fqdn_recipient**: rechaza la solicitud cuando la dirección en RCPT TO no cumple con FQDN.

- **reject_unknown_recipient_domain**: rechaza la solicitud cuando RCPT TO tiene en su dirección un dominio no existente.

- **reject_unauth_destination**: rechaza la conexión si no concuerda con la siguiente premisa, la dirección resuelta esta en `$relay_domains` o un subdominio, y la dirección no contiene ruteo específico de usuario (usuario@maquina.dominio).

Si se envía comandos SMTP antes de saber si **Postfix** soporta *command pipelining*, esto puede ser programas de correo masivo que usan estos comandos para acelerar las entregas, para esto existe una restricción que rechaza la conexión y se configura de la siguiente forma:

```
smtpd_data_restrictions=
             reject_unauth_pipelining,
                permit
```

Otro de los filtros de reputación a configurar en el servicio de correo es el de los registros SPF, este se encarga de verificar qué dominios están autorizados por el sistema para enviar su correo electrónico. El registro SPF se configura de la siguiente forma [3]:

Inicialmente se configura el servidor DNS para que tenga registros SPF de los servidores de correo pertenecientes al dominio. Se añade esta línea en la zona DNS del dominio.

```
@        IN     TXT     "v=spf1   a   mx    ip4:192.168.120.68
~all"
```

Esta línea indica que el servidor con dirección IP 192.168.120.68 tiene autorización para enviar correo electrónico del dominio unicauca.edu.co, el significado de cada componente se explica a continuación:

- **@:** hace referencia al dominio que se esta utilizando, se puede poner el dominio en vez de la @.

- **TXT:** indica que es información textual, quiere decir que los dominios pueden identificarse de forma arbitraria.

- **`V:`** indica la versión de SPF a trabajar, para este caso es la versión 1.

- **`a mx:`** se autoriza a las maquinas que tiene registros de IP de los registros A y con la IP de los registros MX.

- **`~all:`** desautoriza a todas las máquinas que no encajen en la configuración mostrada. Hay casos en que se utilizan servidores intermedios que no se encuentren en el registro SPF, para este caso con esta sintaxis se informa que si el servidor no se encuentra en el registro SPF, se puede dejar pasar, pero para ser analizado con otros métodos, en este caso un filtro antispam, si se da el caso que no se utilicen servidores intermedios, se pueden utilizar la opción "-all" que indica que se rechacen todos los servidores que no cumplan con la autorización SPF.

Existen dos paquetes de *SPF* para **Postfix**, un paquete está escrito en Perl y otro en Python, la versión en Python esta creada para servidores con altas exigencias de tráfico, mientras que la de Perl es para servidores con bajas exigencias. La versión en Perl se llama *postfix-policyd-spf-perl* y la de Python *postfix-policyd-spf-python*, para este sistema de correo se va a trabajar con la versión de Python.

El paquete *postfix-policyd-spf-python* viene acompañado con otros paquetes para que su funcionamiento sea correcto, esto son *python-dns* y *python-spf* y generalmente se instalan automáticamente en conjunto con *postfix-policyd-spf-python*.

```
apt-get install postfix-policyd-spf-python
```

Al terminar la instalación se pasa a modificar los archivos de **Postfix** que se encuentran en `/etc/postfix`, el primer archivo a cambiar es el `master.cf` el cual indica como correr cada uno de los procesos individuales de **Postfix**, a este archivo se le adicionan las siguientes líneas:

```
policyd-spf  unix  -    n       n       -       0       spawn
  user=nobody       argv=/usr/bin/python       /usr/bin/policyd-spf
/etc/postfix-policyd-spf-python/policyd-spf.conf
```

En el archivo `main.cf` se coloca la siguiente configuración:

```
policyd-spf_time_limit=3600
```

Esta línea es el *timeout* de respuesta de SPF. En la lista de restricciones `smtpd_recipient_restrictions` se coloca el siguiente término para que el sistema de correo haga un chequeo SPF de las direcciones que soliciten conexión:

```
check_policy_service unix:private/policyd-spf
```

Esta línea tiene que colocarse después de la restricción `reject_unauth_destination`

para evitar que el sistema quede en relé abierto.

Los otros filtros de reputación quedan configurados en el archivo de **Postfix** `main.cf` de la siguiente forma:

```
smtpd_helo_required=yes
smtpd_rfc821_envelopes=yes
policyd-spf_time_limit=3600

smtpd_recipient_restrictions=
                        permit_sasl_authenticated,
                        permit_mynetworks,
                        reject_invalid_hostname,
                        reject_non_fqdn_hostname,
                        reject_non_fqdn_sender,
                        reject_non_fqdn_recipient,
                        reject_unknown_sender_domain,
                        reject_unknown_recipient_domain
                        reject_rbl_client zen.spamhaus.org,
                        reject_rhsbl_client zen.spamhaus.org,
                        reject_rbl_client bl.spamcop.net,
                        reject_rhsbl_client bl.spamcop.net,
                        reject_unauth_destination,
                        reject_unlisted_recipient,
                        check_policy_service unix:private/policyd-spf,
                       check_policy_service inet:127.0.0.1:10031,
                        permit

smtp_data_restrictions=
                        reject_unauth_pipelining,
                        permit
```

1.2 Configuración de filtros de contenido en Postfix.

A continuación se observa la configuración de los programas encargados de analizar los mensajes que llegan al servidor de correo, la finalidad de estos programas es el detectar si un mensaje tiene virus o son mensajes *spam*.

1.2.1 Amavisd-new

El programa que se va a instalar es **Amavisd-new**, este programa puede contener los módulos para buscar virus y *spam*. Se utiliza **Amavisd-new** que permite buscar virus y *spam* haciendo un solo llamado a los procesos de búsqueda, quiere decir que cuando llega un mensaje a **Postfix**, este hace el llamado a **Amavisd-new** para realizar tareas de búsqueda de virus y *spam*. Si no se utilizara este programa, **Postfix** tendría que hacer el

llamado primero para la búsqueda de virus y después para la búsqueda de mensajes *spam*, pero con **Amavisd-new** el sistema se vuelve más eficiente, para esto se realiza la integración de **Spamassassin** y **ClamAV** dentro de **Amavisd-new**. La instalación del paquete se hace de la siguiente manera [4]:

```
apt-get install amavisd-new
```

El modulo antispam a instalar es **Spamassassin**.

```
Apt-get install spamassassin
```

El módulo antivirus es el **Clamav**. En este modulo se instala otras dependencias como *clamav-freshclam* que se emplea para la configuración del sistema antivirus. La configuración de **Clamav** se baso en la documentación que trae por defecto este programa.

```
Apt-get install clamav
```

Cuando finalice la instalación se inicia la configuración del antivirus, el Clamav-freshclam es el encargado de mantener el antivirus actualizado, para esto se escribe lo siguiente:

```
dpkg-reconfigure  clamav-freshclam
```

Con este comando se inicia el asistente de configuración mostrando la siguiente ventana, Figura 1:

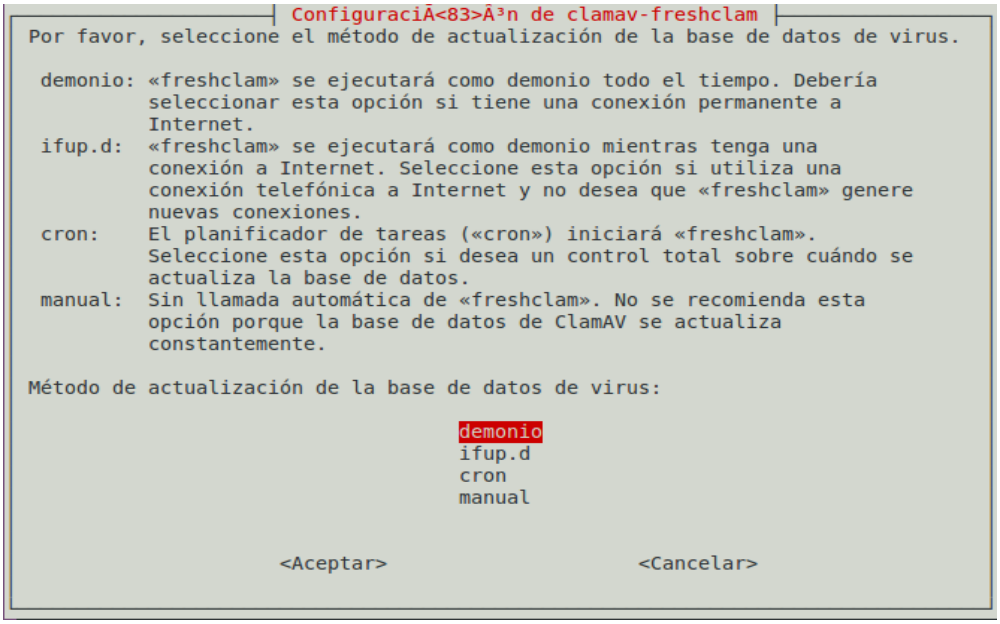

Figura 1. Ventana configuración **Clamav**.

En esta ventana se escoge la opción de demonio ya que se va a mantener una conexión

permanente a internet.

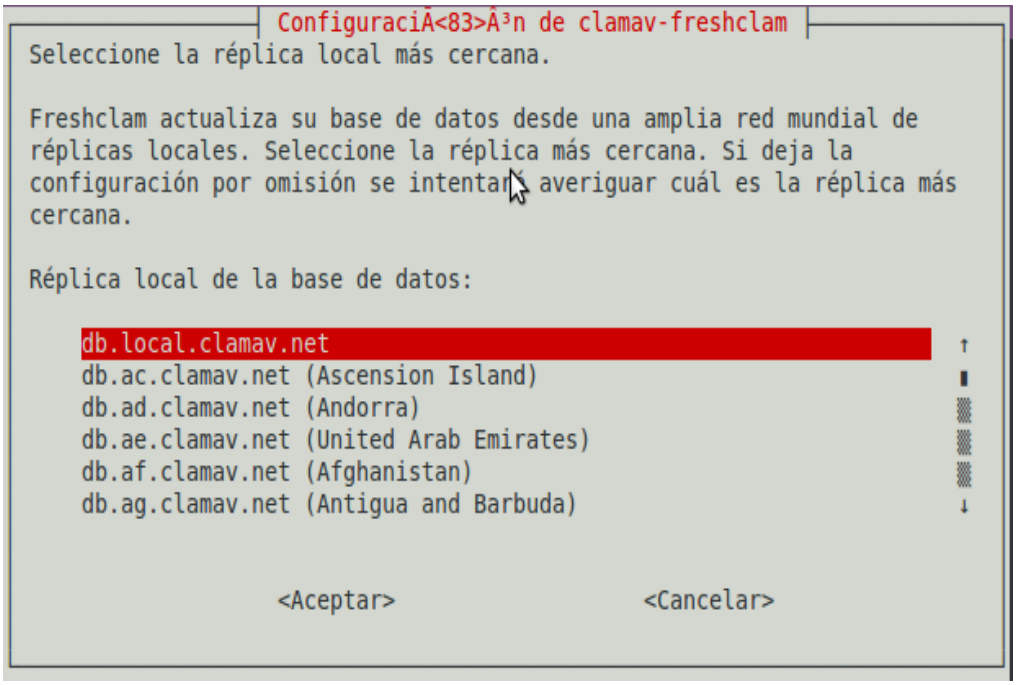

Figura 2. Ventana base de datos de actualización **Clamav**.

En la Figura 2 se escoge la opción por defecto db.local.clamav.net.

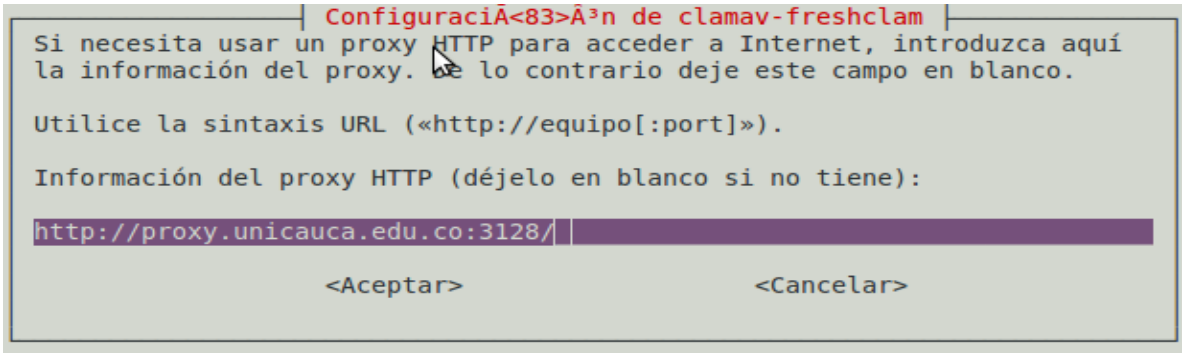

Figura 3. Ventana definición del proxy de dominio.

La Figura 3 muestra la selección del proxy en caso que el dominio tenga configurado uno, esto es necesario para que **Clamav** pueda salir a internet y poder actualizar su base de datos, en este caso se trabaja con el proxy de la universidad del cauca “proxy.unicauca.edu.co” en el puerto “3128”.

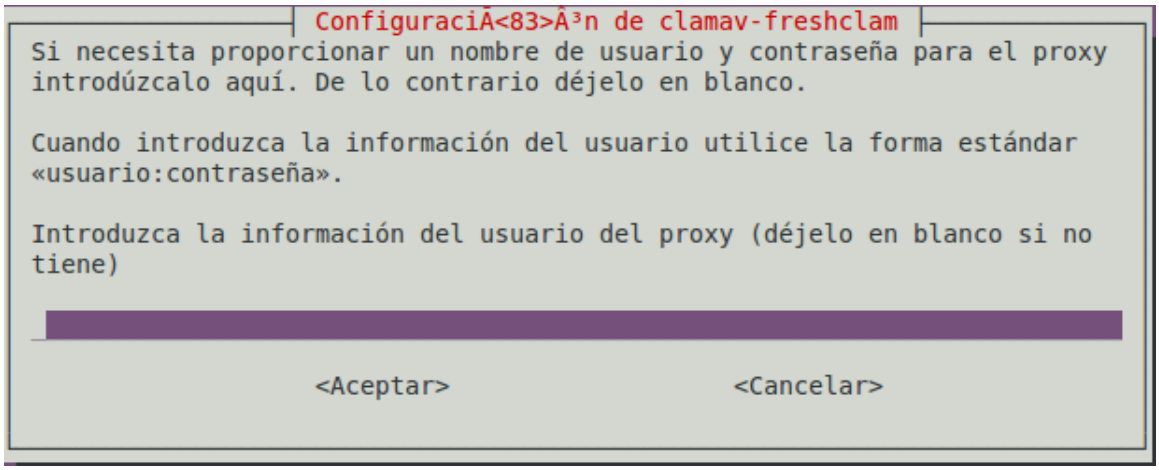

Figura 4. Contraseña del proxy.

En la Figura 4, se deja el espacio en blanco ya que el proxy de la universidad no utiliza contraseña.

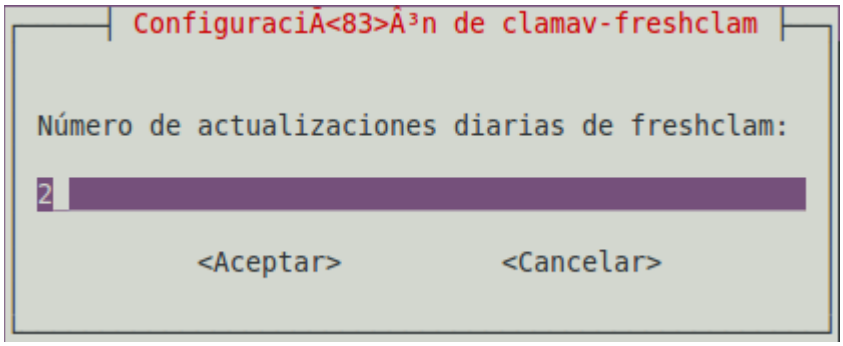

Figura 5. Actualización de **Clamav**

La Figura 5 muestra la ventana que permite escoger el número de actualizaciones diarias que puede tener **Clamav**, por defecto tiene que se realice 24 actualizaciones pero con dos será suficiente.

Después de la configuración de Clamav-freshclam se pasa a instalar el paquete para integrar el antivirus con servidores de correo, este paquete se llama Clamav-daemon también conocido como clamd.

```
apt-get install clamav-daemon
```

Ahora se integra el antivirus **Clamav** al grupo de antivirus de **Amavisd** y viceversa, de este paso se hace lo siguiente:

```
useradd clamav -g amavis
useradd amavis -g clamav
```

Aunque **Amavisd-new** trae por defecto configurado otros antivirus muy buenos tambien analiza los mensajes que llegan con el antivirus **Clamav**. Los antivirus que trae **Amavisd-new** son los que se nombran a continuación:

- Sophie.
- Sophos.
- Trophie.
- F-prot fpscand
- Avg Anti-virus
- DrWebD
- KasperskyLab AVP-aveclient
- KasperskyLab Antiviral Toolkit Pro (AVP)
- KasperskyLab AVPDaemonClient
- CentralCommand Vexira (new) vascan
- Avira Antivir
- Command Antivirus for linux, “csav”
- Symantec CarrierScan.
- Symantec Antivirus Scan Engine
- F-Secure antivirus
- CAI eTrust Antivirus
- MKS_vir for linux version beta
- ESET NOD32 for Linux File servers
- Norman Virus Control v5 / Linux, “nvcc”
- Panda CommandLineSecure 9 for Linux
- NAI McAfee AntiVirus (uvscan)
- VirusBuster
- CyberSoft Vfind
- avast! Antivirus
- BitDefender
- ArcaVir for Linux

Cuando **Amavisd-new** detecta un mensaje con virus lo puede dejar en cuarentena y los guarda en la dirección `/var/lib/amavis/virusmails`.

Los archivos de **Amavisd-new** se ubican en la ruta `/etc/amavis/conf.d`, ahi se encuentran todos los archivos necesarios para que el funcionamiento de **Amavisd-new** sea el correcto. La configuración de estos archivos se muestra a continuación [4]:

En el archivo de **Amavisd-new** `/etc/amavis/conf.d/05-node_id` se comenta la siguiente línea:

```
#chomp($myhostname = `hostname –fqdn`);
```

Se descomenta la línea `$myhostname` a la cual se le agrega el nombre del servidor y el

dominio de trabajo:

```
$myhostname = "vasaiah.unicauca.edu.co";
```

Por defecto **Amavisd-new** trae el antivirus desactivado, para ponerlo en funcionamiento hay que descomentar en el archivo `/etc/amavis/conf.d/15-content_filter_mode`, la siguiente línea:

```
@bypass_virus_checks_maps = (
   \%bypass_virus_checks,                    \@bypass_virus_checks_acl,
\$bypass_virus_checks_re);
```

En el archivo `/etc/amavis/conf.d/15-av_scanners`, se encontrara un segmento como el siguiente:

```
### http://www.clamav.net/
 ['ClamAV-clamd',
   \&ask_daemon, ["CONTSCAN {}\n", "/var/run/clamav/clamd.ctl"],
   qr/\bOK$/m, qr/\bFOUND$/m,
   qr/^.*?: (?!Infected Archive)(.*) FOUND$/m ],
```

Hay que comprobar que la ruta `/var/run/clamav/clamd.ctl` en este segmento sea la misma que se encuentra en el archivo `/etc/clamav/clamd.conf` en la línea:

```
localSocket /var/run/clamav/clamd.ctl
```

Si estas rutas no coinciden, la ruta que se encuentra en `/etc/clamav/clamd.conf` se copia en la ruta que esta en el archivo `/etc/amavis/conf.d/15-av_scanners`.

El paso siguiente es integrar **Amavisd-new** y **Postfix**, entonces en el archivo `main.cf` de **Postfix** se coloca la siguiente línea:

```
content_filter=smtp-amavis:[localhost]:10024
```

Esta línea indica que los correos de **Postfix** serán enviados al proceso **Amavisd-new** por el puerto 10024. Por defecto **Amavisd-new** recibe los correos por el puerto 10024 y los reinyecta por el 10025.

Después se configura en el `master`.cf, como se va ejecutar la llamada a **Amavisd-new**, la configuración es la siguiente:

```
smtp-amavis  unix  -       -       y       -       2      smtp
          -o smtp_data_done_timeout=1200
          -o disable_dns_lookups=yes

127.0.0.1:10025 inet n     -       y       -       -      smtpd
   -o content_filter=
```

```
-o local_recipient_maps=
-o relay_recipient_maps=
-o smtpd_restriction_classes=
-o smtpd_client_restrictions=
-o smtpd_helo_restrictions=
-o smtpd_sender_restrictions=
-o smtpd_recipient_restrictions=permit_mynetworks,reject
-o mynetworks=127.0.0.0/8
-o strict_rfc821_envelopes=yes
```

Esta configuración indica que **Postfix** acepta los mensajes de **Amavisd-new** por el puerto 10025 y que a estos mensajes no se le aplique los filtros de reputación que se le aplican a los mensajes que llegan por el puerto 25 ya que se realizaría un doble trabajo.

También en el sistema de transporte de “pickup” hay que colocar las siguientes líneas, esto se hace para que el sistema no marque los mensajes informativos de *spam* como *spam*.

```
-o content_filter=
-o receive_override_options=no_header_body_checks
```

Las Figuras 6 y 7 son imágenes del archivo `master.cf`, las líneas resaltadas en color negro son las modificaciones que se le hicieron al archivo tanto para utilizar los filtros SPF como para utilizar **Amavisd-new**.

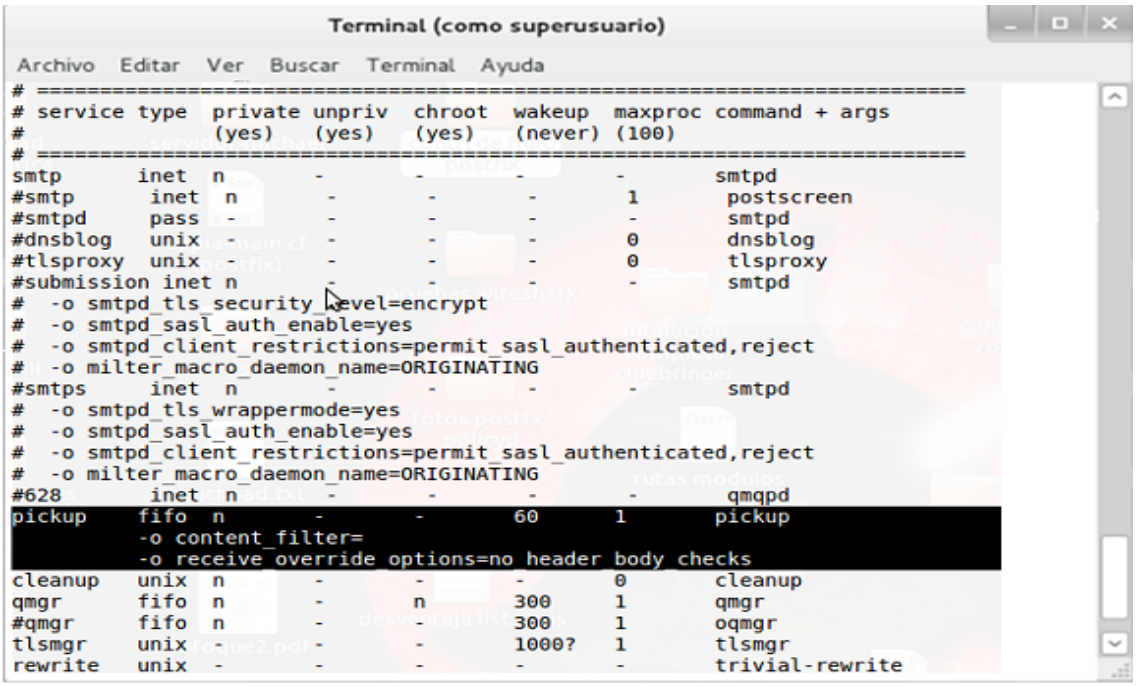

Figura 6. Configuración de mensajes spam en el master.cf

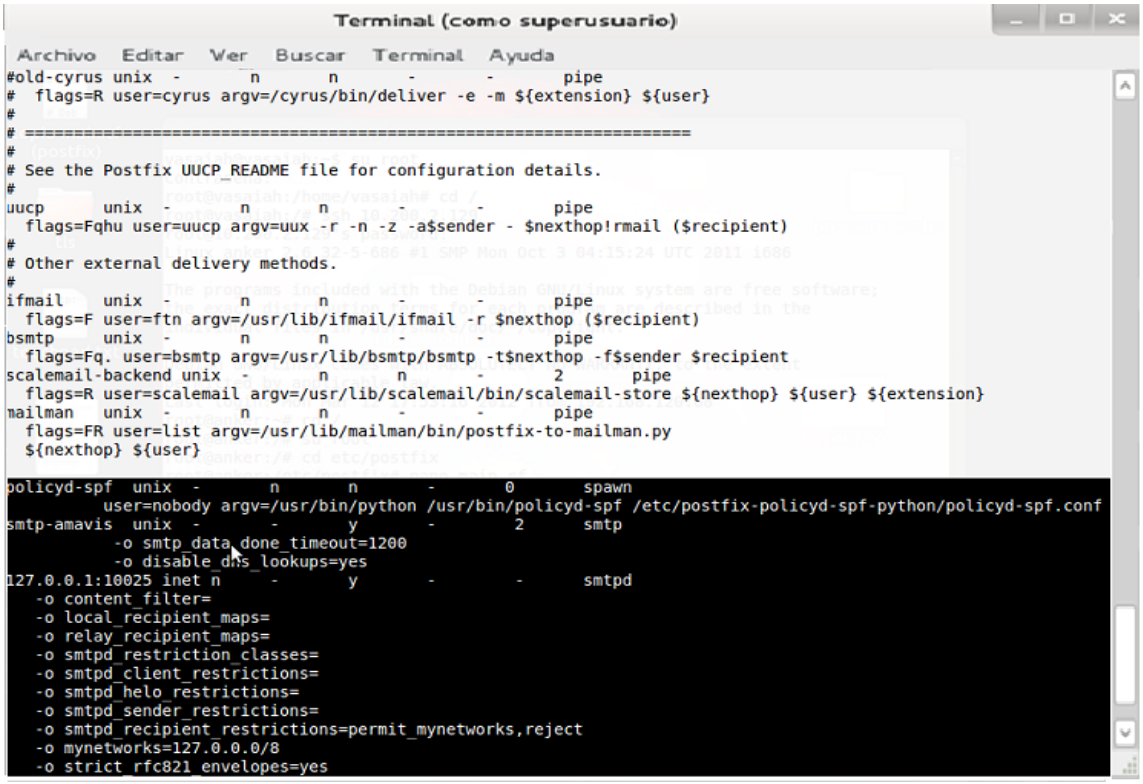

Figura 7. Configuración de Amavisd-new en el master.cf

Ahora se procede a la configuración de **Spamassasin**. Este filtro tiene varios parámetros de configuración y se puede integrar al **Amavisd-new**.

Para la integración de **Spamassasin** con **Amavisd-new** se pasa a modificar el archivo `/etc/ amavis/conf.d/15-content_filter_mode,` en este archivo se descomentan las siguientes líneas:

```
@bypass_spam_checks_maps = (

  \%bypass_spam_checks, \@bypass_spam_checks_acl,
\$bypass_spam_checks_re);
```

Ahora se pasa a la configuración de **Spamassassin** [5], primero que todo si se quiere que trabaje como demonio hay que activarlo para que trabaje como tal, para esto se modifican las siguientes líneas en el archivo `/etc/default/spamassassin`:

```
ENABLED=1
CRON=1
```

Como el **Spamassassin** se encuentra integrado al **Amavisd-new** no se hace necesario que sea ejecutado como demonio [6], su eficiencia se mantiene igual y además al ejecutarse como demonio se le dan privilegios de root y esto genera riesgos en la seguridad.

Lo siguiente es configurar el archivo `/etc/spamassassin/local.cf`. Este archivo tiene los parámetros importantes de **Spamassassin**, muchos de ellos están comentados para darle al administrador libertad de configurar las acciones que se van a tomar cuando un mensaje es detectado como *Spam*, los parámetros son los siguientes:

- **`rewrite_header Subject *****SPAM*****`**: cambia la línea del asunto y añade el texto [SPAM]. Esto permite marcar los mensajes que son *spam* y así poner un filtro en el cliente para poder colocar los mensajes en una carpeta destinada para ello.

- **`report_safe 1`**: esta opción activada indica como **Spamassassin** modifica los mensajes catalogados como *spam*, esta opción activada de esta manera "1" se añade 3 cabeceras al mensaje.

  X-Spam-Level: con * que representa la puntuación.
  X-Spam-Status: línea con la descripción del *spam* y los test coincidentes.
  Adjunto MIME: El informe del *spam.*

- **`report_safe 0`**: deja el cuerpo del mensaje sin tocar y se añade la cabecera X-Spam-Report con la descripción detallada de las reglas que coinciden.

- **`required_score [num]`**: puntuación total. Si las reglas aplicadas suman más que el valor fijado aquí, el mensaje será considerado *spam*. Valores mas altos evitan falsos positivos, pero también pueden permitir que entre mas *spam*.

- **`whitelist_from`**: se utiliza para indicar que los mensajes que lleguen de cierto dominio o cuenta de correo electrónico en particular no se consideren *spam* se pueden definir en varias líneas.

```
Whitelist_from *@dominio.net
whitelist_from 171.15.46.89
```

- **`blacklist_from`**: se utiliza para indicar que los mensajes que lleguen de cierto dominio o cuenta de correo electrónico en particular sean considerados como *spam*.

```
blacklist_from *@dominio.org
```

De acuerdo a [6], para entrenar los filtros bayesianos se utiliza el comando `sa-learn`, primero se utiliza el comando `sa-learn –spam <directorio>` para los correos *spam* y el otro es `sa-learn –ham <directorio>` para los correos *ham*[1] o validos, en el

[1] Ham: mensajes legítimos de correo electrónico

espacio que dice <directorio>, se coloca la ruta donde se encuentran los mensajes, ya sean *spam* o *ham*.

Para aumentar la efectividad de **Spamassassin** se instalan las redes colaborativas para detección de *spam* RAZOR y PYZOR, estas dos redes se complementan unas a otras.

```
apt-get install razor
apt-get install pyzor
```

La configuración de estas herramientas es bastante fácil, para cargar los archivos de Razor a **Spamassassin** se escriben en consola las siguientes líneas [7]:

```
razor-admin  -home=/etc/spamassassin/.razor  -create

razor-admin  -home=/etc/spamassassin/.razor  -register

razor-admin  -home=/etc/spamassassin/.razor  -discover
```

Para la configuración de Pyzor, en la Shell de Linux se escribe lo siguiente:

```
pyzor --homedir /etc/spamassassin/.pyzor discover

pyzor discover
```

Al ingresar las líneas de código anteriores es necesario estar conectado a internet.

Para que **Spamassassin** haga uso de las redes colaborativas se incluyen los siguientes comandos al final del archivo `/etc/spamassassin/local.cf`:

```
razor_config /etc/spamassassin/.razor/razor-agent.conf
pyzor_options --homedir /etc/spamassassin/.pyzor
```

También hay que corroborar que en el archivo `/etc/spamassassin/v310.pre` se encuentren descomentadas las siguientes líneas:

```
loadplugin Mail::SpamAssassin::Plugin::Pyzor

loadplugin Mail::SpamAssassin::Plugin::Razor2
```

Para probar que **Spamassassin** se conecta con **Pyzor** se escribe el siguiente comando en consola.

```
echo "test" | spamassassin -D pyzor 2>&1 | less
```

La respuesta que da el sistema se puede ver en la Figura 8:

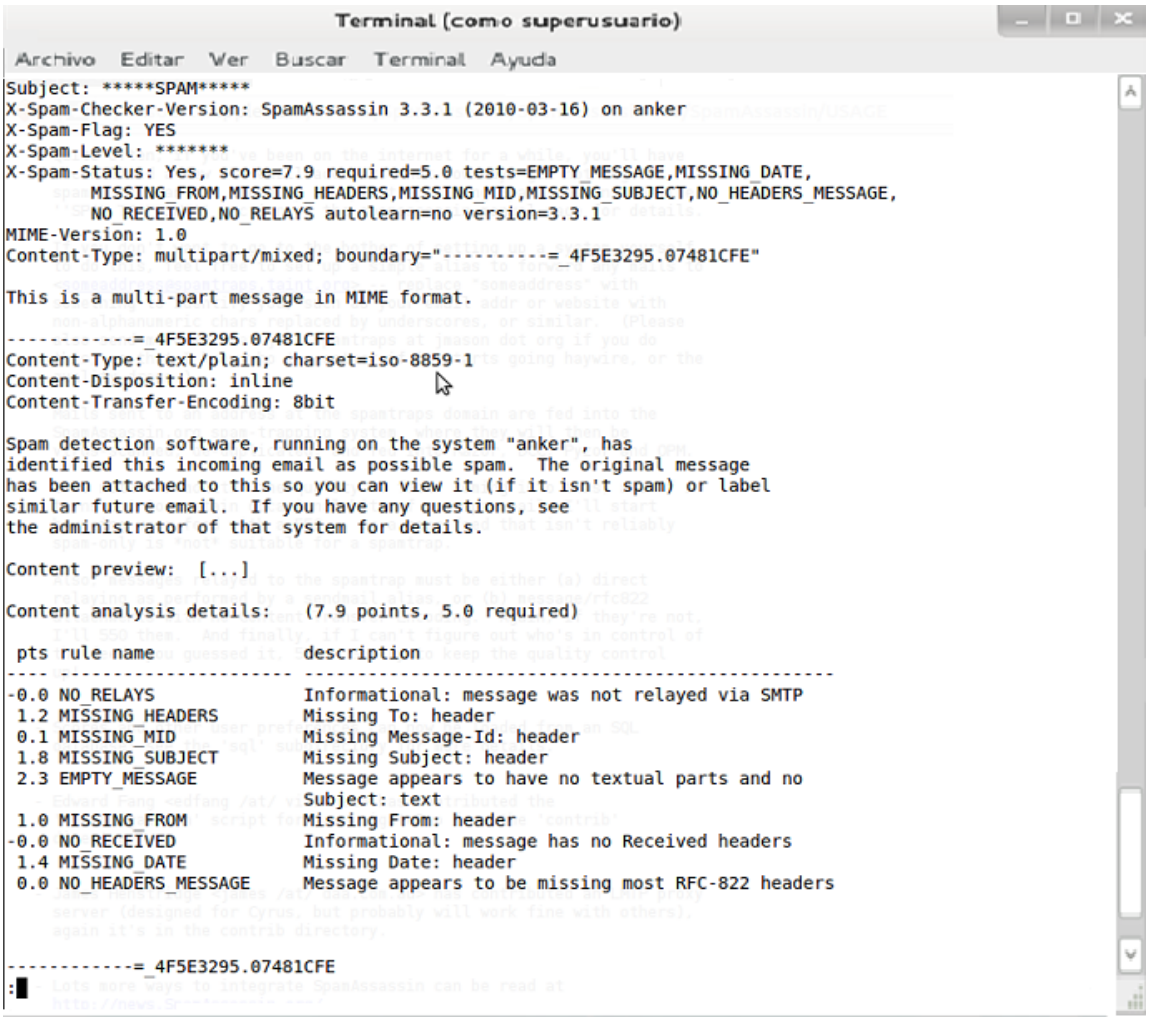

Figura 8. Conexión de **Spamassassin** con **Pyzor**.

La prueba del sistema colaborativo **Razor** para la detección de *spam*, se realiza descargando un mail de prueba de *spam* como se ve a continuación:

```
wget 'http://kb.atmail.com/attach/spam-mail.txt'
```

Por consola se ubica la ruta del archivo descargado y se escribe el siguiente comando:

```
spamassassin -t -D < spam-mail.txt
```

La respuesta debe ser que el sistema hallo al archivo como posible *spam*, esto se puede verificar en la Figura 9:

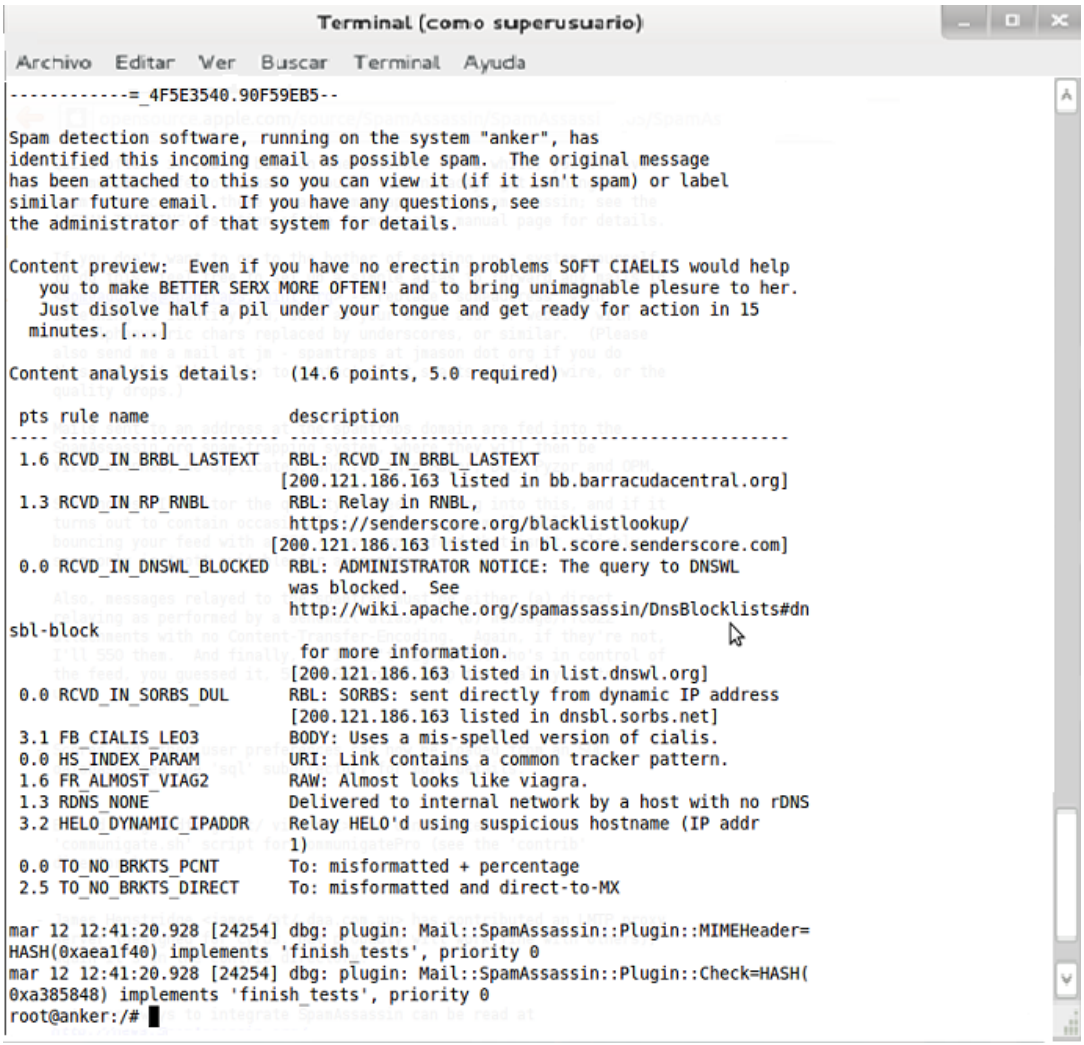

Figura 9. Conexión de **Spamassassin** con **Razor**

Ahora se configura **Spamassassin** para que utilice métodos de aprendizaje bayesiano [7], esto permite que este aprenda automáticamente sobre que mensajes son *spam* y cuales no, para activar esta función se descomenta las siguientes líneas que se encuentran en el archivo `/etc/spamassassin/local.cf`:

```
use_bayes 1: Activa el uso de bayes
bayes_auto_learn 1: Activa el autoaprendizaje
bayes_ignore_header X-Bogosity
bayes_ignore_header X-Spam-Flag
bayes_ignore_header X-Spam-Status
```

Las tres últimas líneas se colocan para establecer los encabezados que puede proporcionar señales inapropiadas para el clasificador bayesiano.

Spamassassin maneja el *spam* a través de puntuación, en el archivo de configuración se puede escoger los niveles de puntuación, pero como **Spamassassin** esta integrado con **Amavis**, en realidad importa la puntuación que tenga **Amavis**. Para estos niveles entonces la configuración que tenga **Spamassassin** acerca de esto se descarta y se toma la configuración que tenga **Amavis**d-new en `/etc/amavis/conf.d/20-debian_defaults` o también lo que se ingrese en `/etc/amavis/conf.d/50-user`. En el archivo `/etc/amavis/conf.d/20-debian_defaults`, se encuentra la siguiente configuración para el antispam:

```
$sa_spam_subject_tag = '***SPAM*** ';
$sa_tag_level_deflt  = 2.0;
$sa_tag2_level_deflt = 6.31;
$sa_kill_level_deflt = 6.31;
$sa_dsn_cutoff_level = 10;

$final_virus_destiny            = D_DISCARD;
$final_banned_destiny           = D_BOUNCE;
$final_spam_destiny             = D_BOUNCE;
$final_bad_header_destiny       = D_PASS;
```

Se explica estas líneas a continuación:

- **`$sa_spam_subject_tag = '***SPAM***'`**: se altera el asunto de los correos catalogados como Spam, el contenido del asunto se modifica y se coloca la palabra ***SPAM***.

- **`$sa_tag_level_deflt`**: **Amavisd-new** califica cada correo con una calificación de *spam*. Por lo general, varía entre 0 y 10. Pero sólo mostrará la calificación en el encabezado si ésta es superior a este valor.

- **`$sa_tag2_level_deflt`**: si la calificación es superior a este valor, se añadirá un encabezado "X-Spam-Status: Yes".

- **`$sa_kill_level_deflt`**: si la calificación es superior a este valor, **Amavisd-new** se encargara de esto. La acción a emprender se define en `$final_spam_destiny`.

- **`$virus_admin = "postmaster\@$mydomain`**: es para que los avisos de virus los reciba un usuario en concreto.

Las acciones que toma **Amavisd-new** respecto a un correo calificado con virus o *spam*, se describen con las siguientes palabras:

- D_DISCARD: el mensaje es rechazado, solo queda una copia en cuarentena.

- D_PASS: el mensaje se deja pasar, pero la cabecera del mensaje tendrá la información de que el mensaje es *spam* y así podrá ser analizado por otros métodos.

- D_BOUNCE: el correo no será enviado a su destinatario pero se enviara una notificación al remitente avisando que el mensaje no ha sido entregado porque ha sido detectado con virus o *spam*.

1.2.2 Postfix-policyd

Es una herramienta de software libre creada para manejar gran cantidad de flujo de correo, cumple funciones de control de *spam* y esta diseñada especialmente para trabajar con el Agente de correo **Postfix**.

Es una herramienta que posee características de control de volumen de mensajes por dominio, control de mensajes por usuario y limite de usuarios por mensaje, también se caracteriza por manejar listas blancas, listas grises y listas negras [8].

1.2.2.1 Instalación de Postfix-policyd.

El prerrequisito para la instalación de **Postfix-policyd** es tener instalada la base de datos **Mysql**, también se requiere de las librerías *Libdbd-mysql-perl, php5-mysql y php5*. La instalación de **Postfix-policyd** se realiza a través de la shell de Linux con el siguiente comando:

```
apt-get install postfix-policyd
```

En el proceso de instalación el sistema solicita la contraseña del administrador de **Mysql** y la contraseña de la base de datos de **Postfix-policyd**, luego de esto la instalación finaliza.

Para iniciar el asistente de **Postfix-policyd** el cual configura la base de datos de este programa, se ingresa a la ruta `/usr/sbin/` y se escribe el siguiente comando:

```
dpkg-reconfigure postfix-policyd
```

Las siguientes figuras muestran todo el proceso de configuración del asistente [9]:

La Figura 10 muestra el inicio del asistente de configuración, en esta se solicita reinstalar la base de datos utilizando este asistente o hacerlo de forma manual, la opción recomendada es a través del asistente.

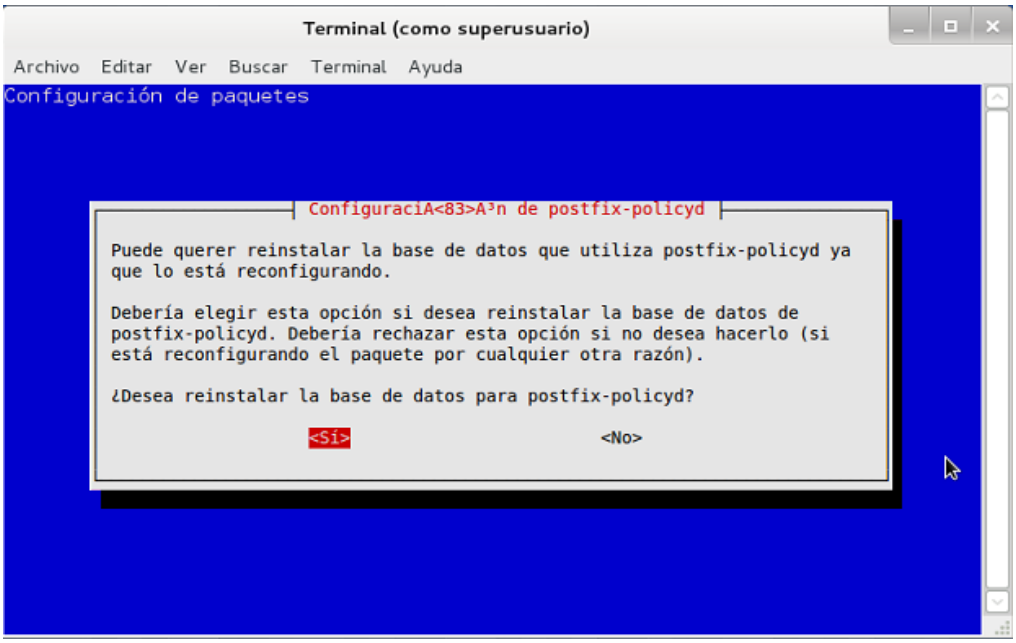

Figura 10. Inicio de asistente de configuración

La Figura 11 describe que tipo de conexión se puede utilizar para conectarse con la base de datos de **Mysql**, existen dos tipos, una conexión por socket Unix en caso de que **Mysql** se encuentre instalada en el mismo equipo y otra por TCP/IP en caso de que **Mysql** se encuentre en un host remoto. La elección para esta configuración es la TCP/IP ya que se puede utilizar para **Mysql** instalada de forma local y remota.

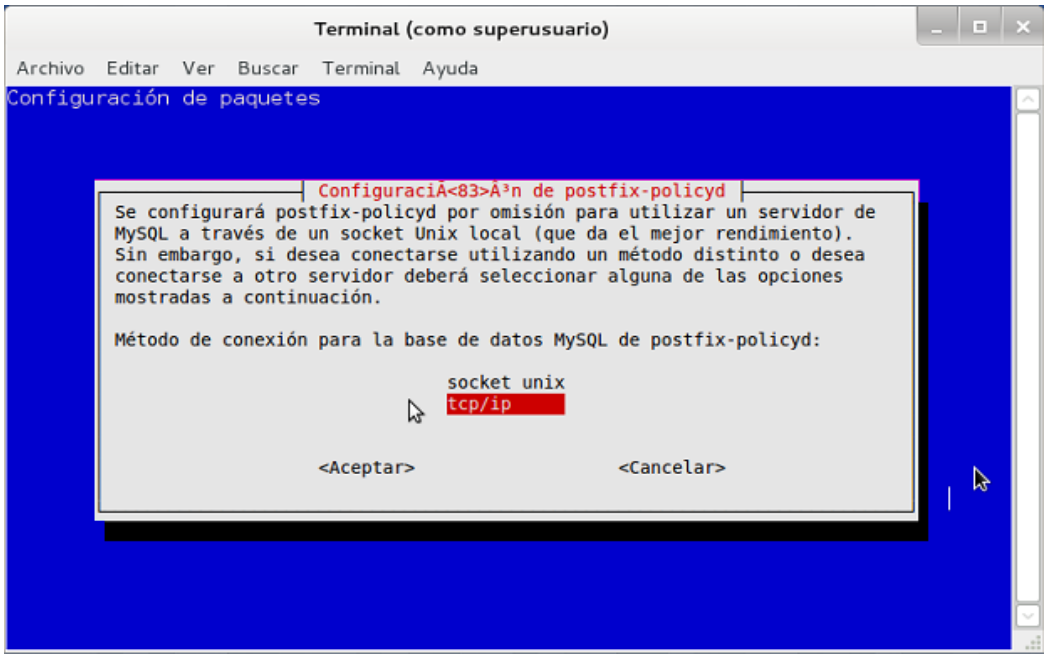

Figura 11. Configuración de la conexión a **Mysql**.

En la Figura 12, se escribe la dirección IP del servidor donde se encuentra instalada **Mysql**, como la base de datos se encuentra instalada en el mismo equipo que **Postfix-policyd** se escribe la dirección IP de localhost.

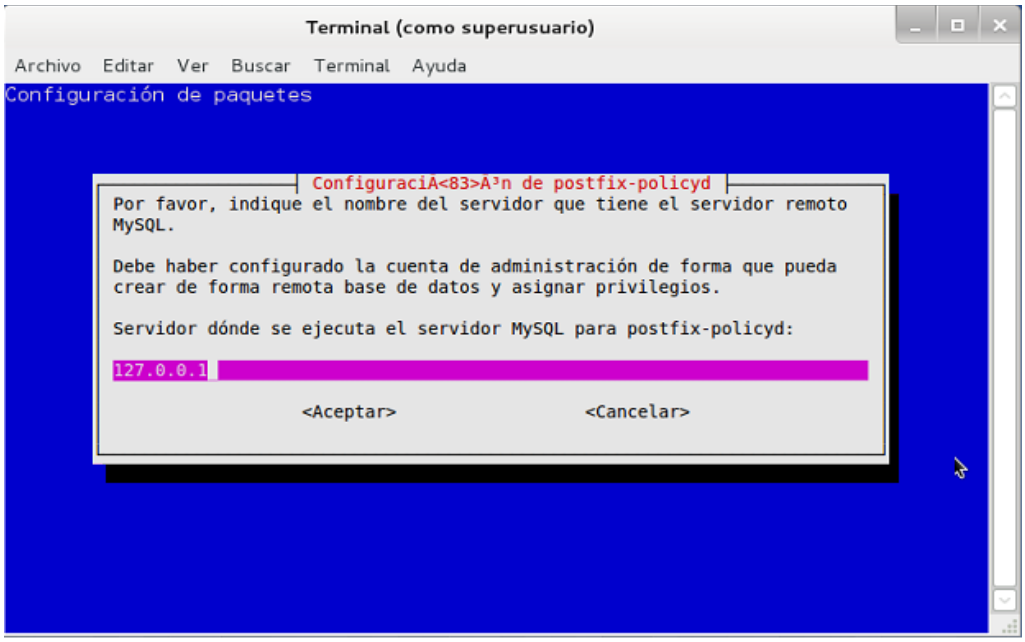

Figura 12. Dirección IP del servidor **Mysql**.

Se escribe en la Figura 13, el puerto por donde trabaja **Mysql**, por defecto la base de datos trabaja en el puerto 3306.

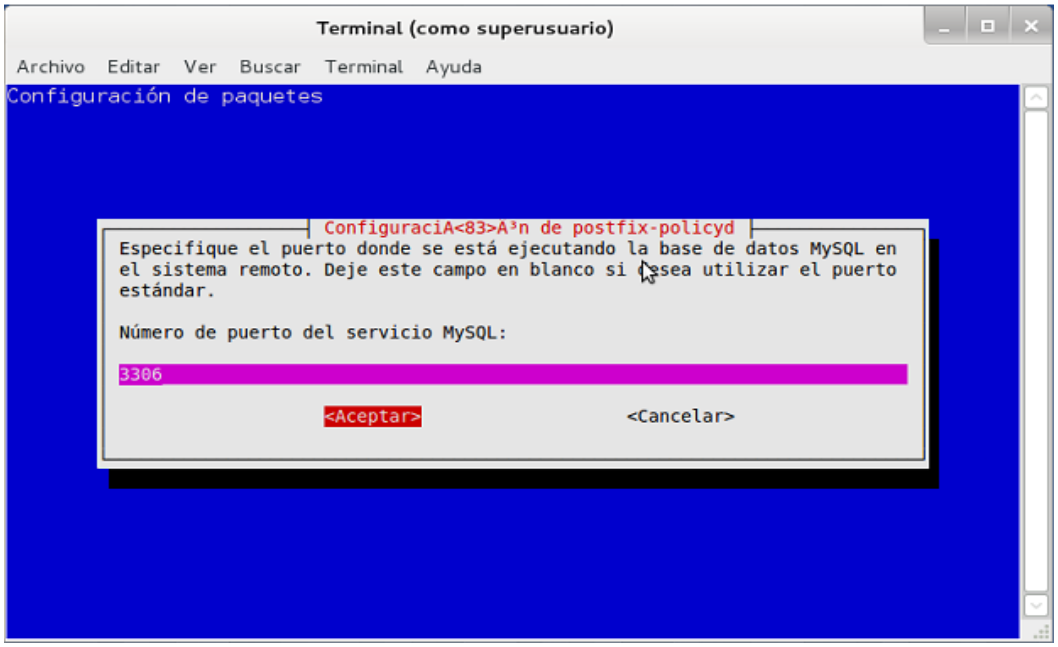

Figura 13. Puerto del servidor **Mysql**

El usuario de administración de la base de datos es el que se muestra en la Figura 14, este usuario generalmente se denomina root, en la Figura 15 se escribe la contraseña para el root de **Mysql**.

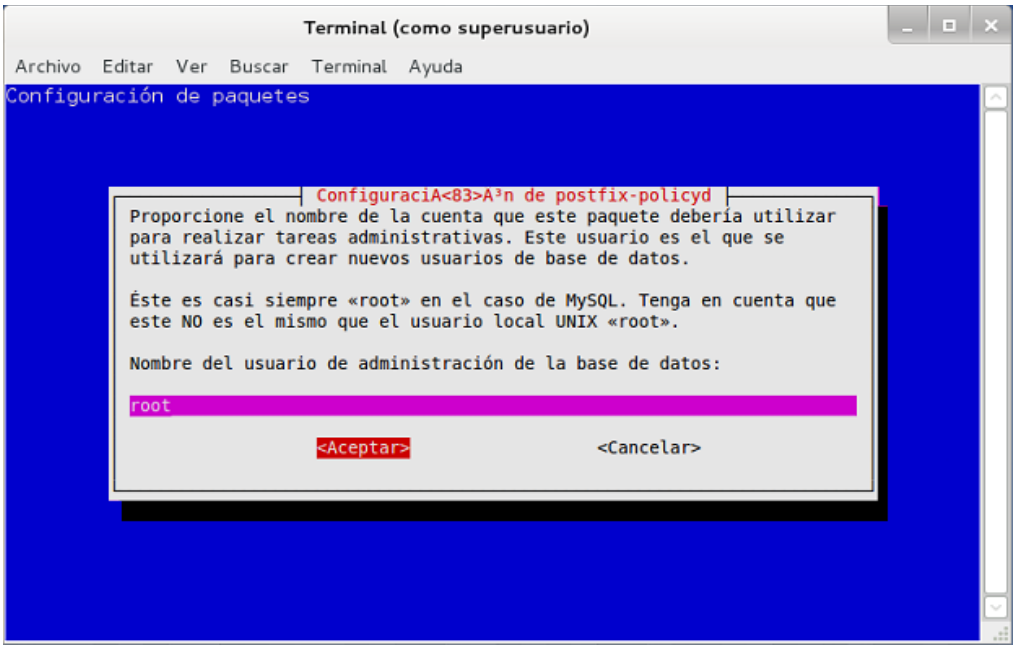

Figura 14. Administrador de **Mysql**

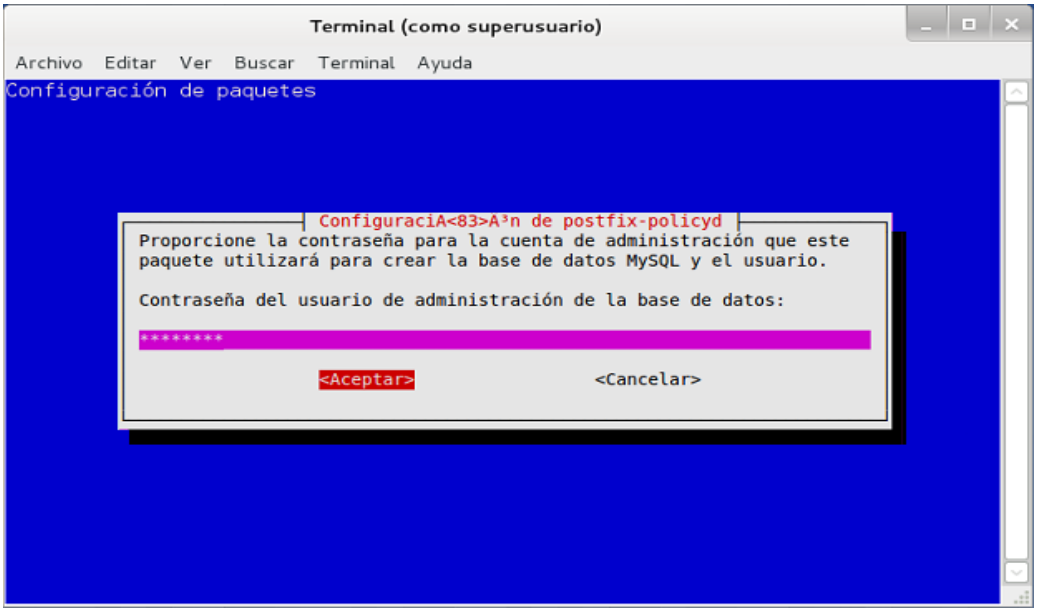

Figura 15. Contraseña administrador de **Mysql.**

En la Figura 16, se define el usuario llamado **Postfix-policyd**, este se utiliza para que el *antispam* **Postfix-policyd** se registre ante la base de datos sin tener los privilegios que maneja el usuario root.

En Figura 17, se define el nombre que tendrá la base de datos del antispam.

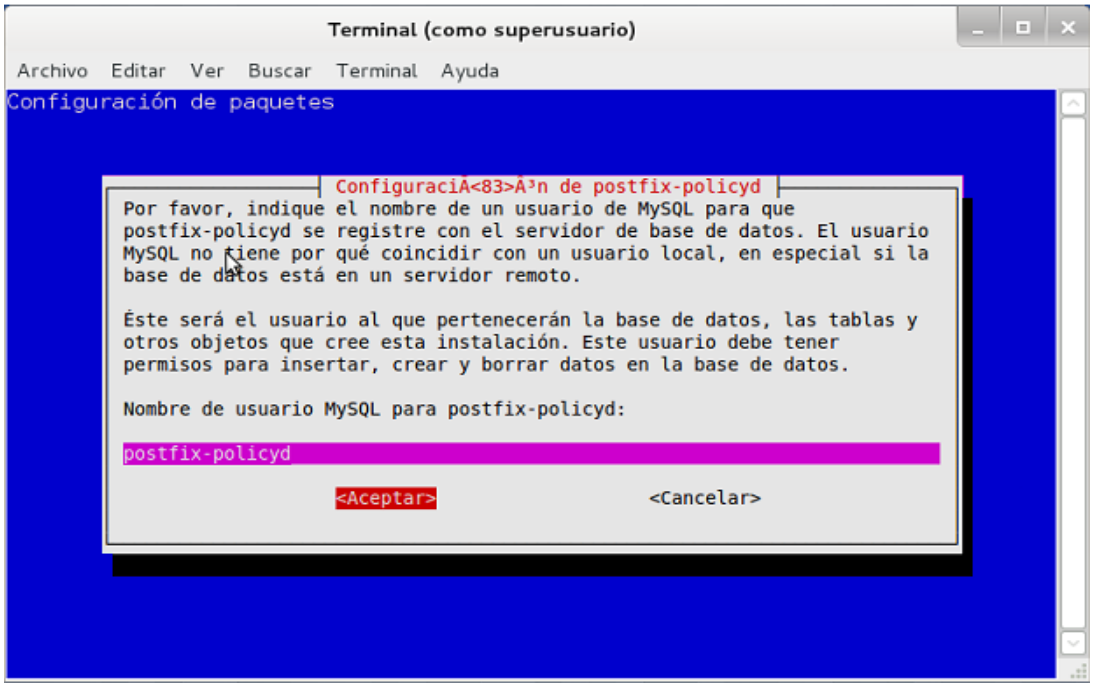

Figura 16. Usuario **Postfix-policyd** en **Mysql**.

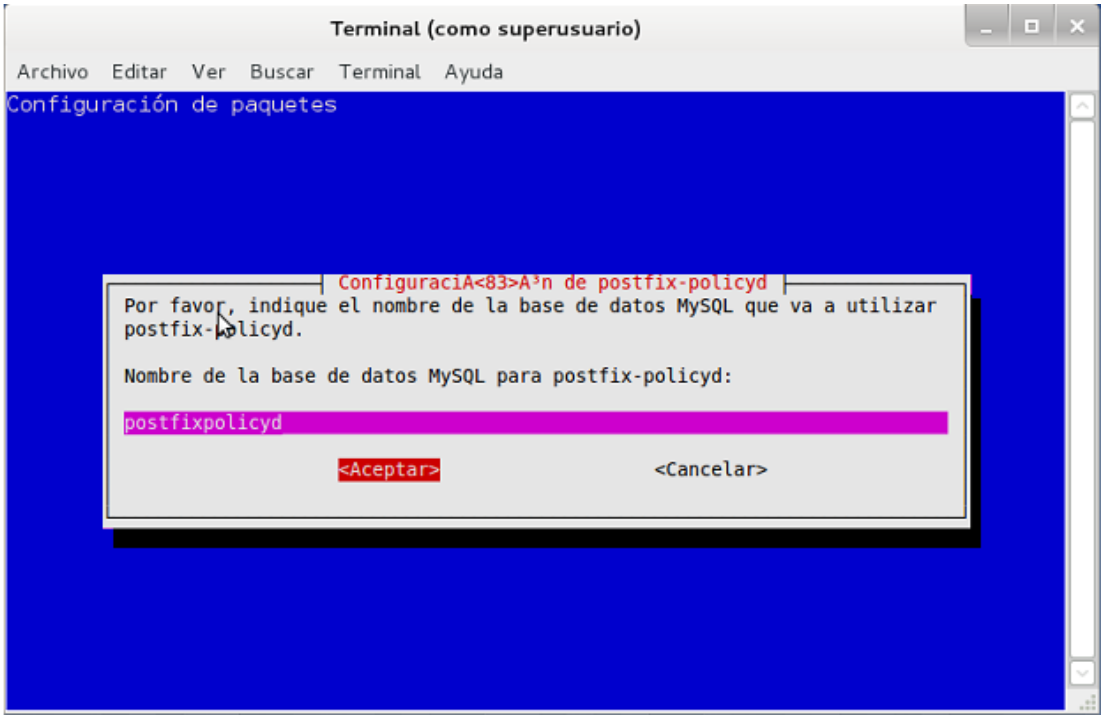

Figura 17. Nombre de la base de datos de **Postfix-policyd**

El asistente crea un usuario llamado **Postfix-policyd** el cual va a administrar la base de datos para el programa, pero es posible que el sistema no trabaje con este usuario y no lo

deje conectar a la base de datos, por lo tanto se trabaja con el usuario root de **Mysql**.

1.2.2.2 Configuración del archivo Postfix-policyd.conf.

La configuración se realiza teniendo como referencia la documentación que este programa trae por defecto .El archivo de configuración de **Postfix-policyd** se encuentra en la ruta `/etc/postfix-policyd.conf`, este archivo por defecto viene con todas sus características desactivadas para darle al administrador libertad en la configuración del antispam. Este archivo esta separado en varias secciones para facilitar su configuración, estas son:

- `DATABASE CONFIG:` maneja todo lo relacionado a la conexión con la base de datos **Mysql**.
- `DAEMON CONFIG:` parámetros generales para la configuración del programa.
- `SECURITY:` maneja lo que respecta al acceso al programa.
- `WHITELISTING:` se configura las listas blancas en **Postfix-policyd**.
- `BLACKLISTING:` aspecto generales para el manejo de las listas negras en el programa.
- `BLACKLISTING HELO:` se configura el programa para buscar en las listas negras, incluso si el servidor remitente utiliza el comando HELO/EHLO.
- `BLACKLIST SENDER:` se utiliza para bloquear dominios o direcciones IP.
- `HELO_CHECK:` revisa la información que llega con el comando HELO.
- `SPAMTRAP:` es un modulo que permite capturar hosts que envían correo hacia una trampa de *Spam*, esta trampa es una dirección de correo creada para realizar esta operación.
- `GREYLISTING:` se configura todo lo relacionado a las listas grises.
- `SENDER THROTTLE:` quotas de envió que el sistema puede manejar.
- `RECIPIENT THROTTLE:` se manejan las quotas de recepción del sistema.

Para la descripción de algunos de los parámetros se tiene en cuenta el término "*triplet*", este término tiene incluido tres aspectos para el envío de correo, estos son [10]:

1. La dirección IP del host que intenta el envío.
2. La dirección del remitente.
3. La dirección del destinatario.

En la sección de DATABASE CONFIG del archivo postfix-policyd.conf se puede cambiar el usuario de **Mysql** con el que se va a trabajar, por defecto el usuario es **Postfix-policyd** pero en este caso se va a trabajar con el usuario root de **Mysql**, se modifica la siguiente línea de esta manera:

```
MYSQLUSER="root"
```

En el archivo postfix-policyd.conf en la sección DAEMON CONFIG se modifica la línea:

```
SYSLOG_FACILITY="LOG_MAIL | LOG_INFO"
```

Es necesario modificarla, en la parte "LOG_MAIL | LOG_INFO", se eliminan los espacios que existen entre ellas, quedando de la siguiente manera:

```
SYSLOG_FACILITY="LOG_MAIL|LOG_INFO"
```

Para cargar la configuración y reiniciar **Postfix-policyd**, se escriben los siguientes comandos:

```
/etc/init.d/postfix-policyd force-reload
/etc/init.d/postfix-policyd restart
```

Se puede limitar el envío de mensajes de correo para evitar que las cuentas sean usadas por *Spammers*, esta configuración se realiza en la sección SENDER THROTTLE, los parámetros de esta sección se explican a continuación:

- SENDERTHROTTLE: activa el límite de quotas por usuario. Este parámetro puede tener dos posibles valores “1” activado y “0” desactivado.
- SENDER_THROTTLE_SASL: limita el envío de correo a usuarios que se autentican a través de SASL. Este parámetro puede tener dos posibles valores “1” activado y “0” desactivado.
- SENDER_THROTTLE_HOST: activa el límite de quotas por dirección IP. Este parámetro puede tener dos posibles valores “1” activado y “0” desactivado.
- QUOTA_EXCEEDED_TEMP_REJECT: se especifica el tipo de rechazo en caso de que se exceda en la quota. “1” para rechazo normal y “0” rechazo severo.
- SENDER_QUOTA_REJECTION: en este parámetro se escribe el mensaje que se muestra al usuario en caso de que se exceda en la quota.
- SENDER_SIZE_REJECTION: se escribe el mensaje que se muestra al usuario en caso de que se exceda en el tamaño del mensaje.
- SENDERMSGLIMIT: cantidad de mensajes que un usuario puede enviar por periodo de tiempo.

- SENDERRCPTLIMIT: cantidad de usuarios permitidos al que se le pueden enviar un mensaje por periodo de tiempo.

- SENDERQUOTALIMIT: tamaño total de los mensajes en bytes que el usuario puede enviar en un periodo de tiempo.

- SENDERTIMELIMIT: periodo de tiempo definido para el envío de mensajes, al pasar este tiempo se reinician los contadores.

- SENDERMSGSIZE: define el tamaño permitido de un solo mensaje.

- SENDERMSGSIZE_WARN: genera una advertencia de gravedad al *syslog* cuando se supera el tamaño del mensaje a un valor en porcentaje definido en este parámetro al tamaño establecido en SENDERMSGSIZE.

- SENDERMSGSIZE_PANIC: genera una advertencia de pánico al *syslog* cuando se supera el tamaño del mensaje a un valor en porcentaje definido en este parámetro al tamaño establecido en SENDERMSGSIZE.

- SENDER_INACTIVE_EXPIRE: se guarda en una base de datos los usuarios que han sido bloqueados y se borran después de que pasen los días especificados en este parámetro.

Los parámetros de la sección WHITELISTING se describen a continuación:

- WHITELISTING: activa el uso de listas blancas.

- WHITELISTNULL: los *spammers* suelen utilizar remitentes nulos o no existentes para el envío de correo, si este parámetro se activa, el servidor devolverá la llamada para asegurarse que el remitente exista.

- WHITELISTSENDER: si la entrada Mail From del remitente de un correo coincide con alguna entrada que se encuentre en la tabla whitelist_sender de la base de datos postfixpolicyd, este remitente será eximido de cualquier validación que el antispam realice.

- WHITELISTDNSNAME: esta es la lista blanca de los dominios que tienen buenos registros de resolución en el DNS. Esta lista se encuentra en la tabla llamada whitelist_dnsname que pertenece a la base de datos postfixpolicyd, los que pertenecen a esta listas blancas son proveedores o dominios en los cuales se esta seguro que no generan *spam*

- AUTO_WHITE_LISTING: activa la integración automática a una lista blanca al momento de efectuar cierta cantidad de *triplets*.

- AUTO_WHITELIST_NUMBER: en este parámetro se específica el numero de *triplets* exitosos requeridos para que la red sea adicionada automáticamente a la lista blanca.

- AUTO_WHITELIST_NETBLOCK: en este parámetro se define si se integra una red (clase c) completa a la lista blanca o solamente el host de dicha red que cumple con los requisitos de *triplets*.

- AUTO_WHITELIST_EXPIRE: este parámetro especifica el periodo de tiempo que pertenecerán las redes que han sido integradas automáticamente a la lista blanca.

En la sección BLACKLISTING se encuentran los siguientes parámetros:

- BLACKLISTING: parámetro para activar las listas negras, puede tener dos valores posibles, “1” activado y “0” desactivado.

- BLACKLISTDNSNAME: ofrece la posibilidad de poner en la lista negra el dominio que se considere que envía correo *spam*. Puede adquirir dos valores posibles, “1” activado y “0” desactivado. Estas se incluyen en la tabla BLACKLISTING que pertenece a la base de datos postfixpolicyd.

- BLACKLIST_TEMP_REJECT: permite rechazar hosts de forma temporal o definitiva, “4xx” es rechazo temporal y “5xx” es rechazo fuerte. 1=4xx 0=5xx.

- BLACKLIST_NETBLOCK: cuando los hosts se encuentran en la lista negra, debería estar el host en la lista negra o bloquear la red o subred completa (class C). 1=class 0=host.

- BLACKLIST_REJECTION: es el mensaje de error que recibirán los hosts que han sido incluidos en la lista negra.

- AUTO_BLACKLIST_NUMBER: numero de *triplets* no exitosos antes de que la red sea puesta en la lista negra automáticamente.

- AUTO_BLACK_LISTING: permite incluir en la lista negra, redes que han excedido el AUTO_BLACKLIST_NUMBER de *triplets* no autenticados.

- AUTO_BLACKLIST_EXPIRE: especifica el periodo de tiempo en que un host estará en una lista negra. Si se coloca “0” (cero), se encontrara permanentemente en la lista negra.

En la etapa BLACKLISTING HELO se especifica lo siguiente

- BLACKLIST_HELO: poner en lista negra a quien intenta suplantar identidades. Un servidor no confiable debe estar usando la identidad helo para conectarse a al servidor. “1” activado, “0” desactivado.

- BLACKLIST_HELO_AUTO_EXPIRE: permite especificar un periodo de tiempo en el que un host se encontrara en la lista negra luego de que haya sido atrapado suplantando un HELO para identificarse al mismo. Si es “0” (cero), estará permanentemente en la lista negra.

En la sección BLACKLIST SENDER se especifica el siguiente parámetro:

- BLACKLISTSENDER: permite usar **Policyd** para bloquear dominios o direcciones de correo. “1” activado y “0” desactivado.

La etapa de SPAMTRAP contiene lo siguiente:

- SPAMTRAPPING: permite capturar hosts que envían correo a una trampa de *spam* o *spamtrap*, sin tener q recurrir a análisis de correos para identificar remitentes. En “1” se encuentra activado y en “0” desactivado.

- SPAMTRAP_REJECTION: es el mensaje que recibe el host que se conecta al *spamtrap.*

- SPAMTRAP_AUTO_EXPIRE: periodo de tiempo que el host estará en la lista negra luego de ser capturado enviando mensajes al *spamtrap*.

1.2.2.3 Configuración de las restricciones de Postfix.

La configuración en el archivo main.cf queda de la siguiente manera:

```
smtpd_helo_required=yes
smtpd_rfc821_envelopes=yes
policyd-spf_time_limit=3600
smtpd_recipient_restrictions=
                      permit_sasl_authenticated,
                      permit_mynetworks,
                      reject_invalid_hostname,
                      reject_non_fqdn_hostname,
                      reject_non_fqdn_sender,
                      reject_non_fqdn_recipient,
                      reject_unknown_sender_domain,
                      reject_unknown_recipient_domain
                      reject_rbl_client zen.spamhaus.org,
                      reject_rhsbl_client zen.spamhaus.org,
                      reject_rbl_client bl.spamcop.net,
                      reject_rhsbl_client bl.spamcop.net,
                      reject_unauth_destination,
                      reject_unlisted_recipient,
                      check_policy_service   unix:private/policyd-
```

```
spf,
                        check_policy_service inet:127.0.0.1:10031,
                        permit

smtp_data_restrictions=
                        reject_unauth_pipelining,
                        permit

smtpd_end_of_data_restrictions=

                        check_policy_service inet:127.0.0.1:10031
```

Las ultimas dos lineas `smtpd_end_of_data_restrictions=` y `check_policy_service inet:127.0.0.1:10031` son las que permiten incluir el servicio de *antispam* de **Postfix-policyd** con el MTA **Postfix**. En la sección de restricciones de recepcion `smtpd_recipient_restrictions=`, también es necesario incluir la linea `check_policy_service inet:127.0.0.1:10031.`

1.2.3 Dspam

Para manejar esta herramienta se requiere de la instalación de los siguientes paquetes:

- dspam
- dspam-doc
- libdspam7-drv-mysql
- postfix-pcre

La instalación de lo paquetes se realiza a través de los repositorios de Linux [11].

`apt-get install dspam dspam-doc  libdspam7-drv-mysql postfix-pcre`
Se muestra el asistente de configuración de la bases de datos de libdspam7-drv-mysql, se configura como el usuario lo requiera y se da aceptar.

El documento de configuración del **Dspam** se encuentra en la ruta `/etc/dspam/` y se denomina `dspam.conf` este documento debe quedar de la siguiente manera.

```
StorageDriver /usr/lib/dspam/libmysql_drv.so
ServerMode      auto
ServerParameters"--deliver=innocent"
ServerIdent           "localhost.localdomain"
ServerPID       /var/run/dspam/dspam.pid
ServerDomainSocketPath     "/var/spool/postfix/dspam/dspam.sock"
DeliveryHost          127.0.0.1
DeliveryPort          10026
DeliveryIdent         localhost
DeliveryProto         SMTP
ParseToHeaders        on
```

```
ChangeModeOnParse     on
ChangeUserOnParse     full
```

Ahora se activa la configuración para todos los usuarios [12], esto se hace creando el documento `group` que se encuentra en la ruta `/var/spool/dspam`, el documento se crea con el comando `nano`.

```
nano /var/spool/dspam/group
```

En este documento se incluye la siguiente línea:

```
dspam:shared:*.domain.com
```

En `domain.com` se coloca el nombre del dominio en el que se esa trabajando, en este caso unicauca.edu.co

En la ruta `/etc/default` se encuentra el documento `dspam` la siguiente línea se cambia para que quede de la siguiente manera:

```
START=yes
```

En el archivo `/etc/postfix/master.cf` la línea "`smtp inet n - n - - smtpd`" tiene que quedar de la siguiente forma:

```
smtp      inet  n       -       n       -       -        smtpd
  -o content_filter=lmtp:unix:dspam/dspam.sock
```

En el archivo `/etc/postfix/mainc.cf`, se coloca la siguiente línea para que **Postfix** trabaje con el **Dspam**, esta línea debe ir en las restricciones de rcpt:

```
smtpd_recipient_restrictions =
  [...]
  check_recipient_access pcre:/etc/postfix/dspam_filter_access
```

Después de esta restricción también se coloca la siguiente línea:

```
dspam_destination_recipient_limit=1
```

Hay que crear el archivo `dspam_filter_access` en la ruta `/etc/postfix`.

```
nano /etc/postfix/dspam_filter_access
```

Este archivo debe contener lo siguiente:

```
 /./ FILTER dspam:dspam
```

Ahora este archivo se convierte en un formato legible para **Postfix**.

```
sudo postmap /etc/postfix/dspam_filter_access
```

Al ejecutar este comando debe aparecer un archivo `dspam_filter_access.db`

Ahora al archivo `/etc/postfix/master.cf` se le incluyen también las siguientes líneas:

```
dspam                  unix    -       n       n       -       -
pipe
  flags=Ru user=dspam argv=/usr/bin/dspam
  --client
  --deliver=innocent,spam
  --user ${recipient}
  --mail-from=${sender}

127.0.0.1:10026 inet     n         -         -         -         -
smtpd
                -o content_filter=
                -o
receive_override_options=no_unknown_recipient_checks,no_header_bod
y_checks
                -o smtpd_helo_restrictions=
                -o smtpd_client_restrictions=
                -o smtpd_sender_restrictions=
                -o
smtpd_recipient_restrictions=permit_mynetworks,reject
                -o                          mynetworks=127.0.0.0/8
                -o smtpd_authorized_xforward_hosts=127.0.0.0/8
```

Se crea el archivo que contendrá el socket para la comunicación de **Postfix** y **Dspam**, este archivo se llamara `dspam` y estará en la ruta `/var/spool/postfix,` esto se hace para permitir la comunicación entre estos dos programas, de lo contrario el sistema de antispam no funcionara. El comando para crear la carpeta es el siguiente:

```
mkdir /var/spool/postfix/dspam/
```

Con el siguiente comando se cambia el usuario y grupo a la cual pertenecerá el directorio `/var/spool/postfix/dspam/`, este directorio pertenecerá al usuario y grupo `dspam:dspam`.

```
chown dspam:dspam /var/spool/postfix/dspam/
```

Ahora como **Dspam** ha sido integrado a **Postfix**, el paso a seguir es entrenar el filtro bayesiano para que el *antispam* pueda diferenciar entre un correo legítimo y uno que sea *spam*, este entrenamiento se hace con el con el siguiente comando:

```
dspam_train [dirección_carpetaSpam] [dirección_carpetaHam]
```

Los mensajes de correo que se encuentran en las capetas que contienen *spam* y las que contiene *ham*, deben estar en formato *Maildir* para que el comando `dspam_train` pueda ejecutarse correctamente.

1.3 Descripción de las pruebas realizadas a los filtros de contenido.

Para comprobar la efectividad de rechazo de mensajes *spam* por los filtros de contenido se tomo una cantidad de estos mensajes que fueron facilitados por la División de TIC de la Universidad del Cauca. Estos mensajes fueron enviados uno a uno de forma manual por medio de un cliente de correo hacia un usuario que pertenece al servicio, esto se realiza para que los filtros de contenido tomen cada mensaje para su posterior análisis y así verificar cual es detectado como *spam* y cual no.

El filtro genera un registro por cada mensaje que es analizado informando si este es *spam* o legítimo tal y como se muestran en las Figura 18 y Figura 19 respectivamente.

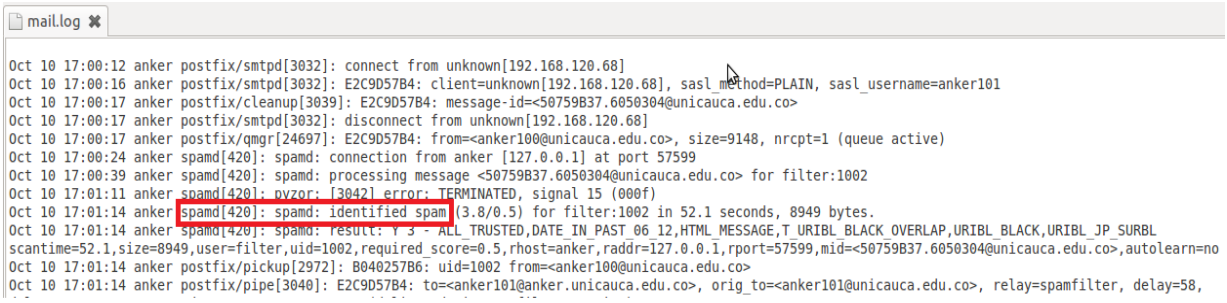

Figura 18. Mensaje de correo detectado como *spam*

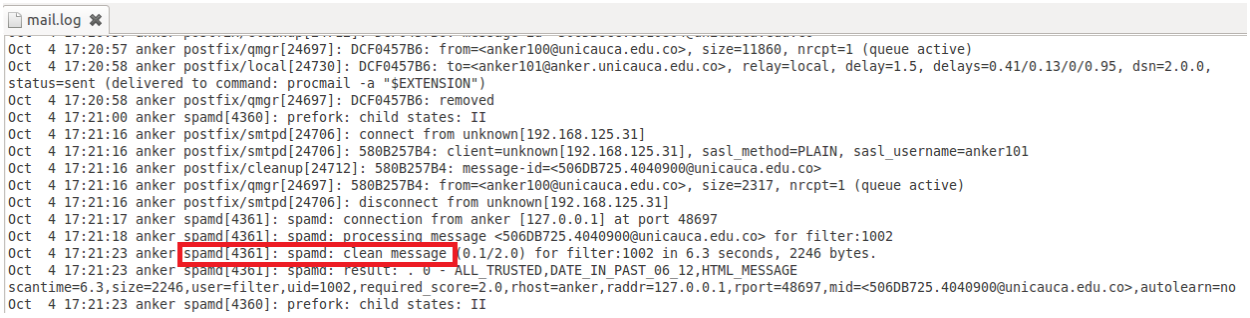

Figura 19. Mensaje de correo detectado como legítimo

Se toma este registro y por medio de la herramienta **Mailgraph** se genera una gráfica que permite observar de manera detallada la cantidad de mensajes que fueron detectados por el filtro de contenido como *spam* o legítimos. La instalación y configuración de **Mailgraph** se describe a continuación.

El primer paso es la instalación de la herramienta.

```
apt-get install mailgraph
```

Después de la instalación de la herramienta se procede a configurar el archivo que se encuentra en la ruta /etc/default/mailgraph y se modifica la línea que se muestra continuación.

```
MAIL_LOG= /var/log/mail.log
```

El comando anterior indica la ruta del archivo de registro del sistema de correo, esto servirá para que **Mailgraph** pueda generar su grafica de estadísticas de mensajes legitimos y *spam.*

Por ultimo para guardar la configuración en el sistema se reinicia el servicio de **Apache** y **Mailgraph.**

```
/etc/init.d/apache2 restart
/etc/init.d/mailgraph restart
```

Para visualizar las graficas generadas por el **Mailgraph** es necesaria la utilización de un navegador web, en este se introduce la dirección http://127.0.0.1/mailgraph que indica la IP del equipo local y el nombre de la interfaz de esta herramienta, la Figura 20 muestra el funcionamiento de **Mailgraph**.

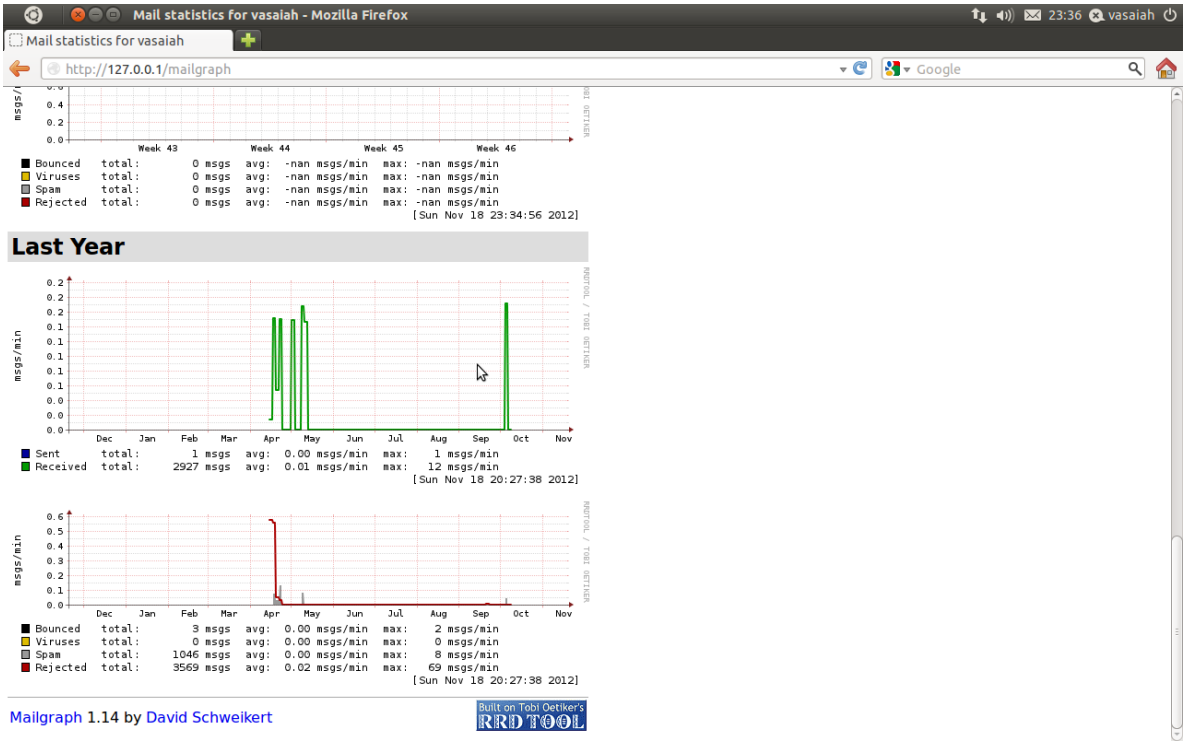

Figura 20. Graficas generadas por **Mailgraph**

REFERENCIAS BIBLIOGRÁFICAS

[1] P. Moreno, 2010. [En línea]. Available: http://pheriko.blogspot.com/2010_04_01_archive.html.

[2] M. Nieto, 2010. [En línea]. Available: http://www.slideshare.net/miguelangelnieto/curso-smtp-avanzado.

[3] 2010. [En línea]. Available: http://phenobarbital.gnu.org.ve/doku.php/weyu:postfix:spf.

[4] 2011. [En línea]. Available: http://www200.pair.com/mecham/spam/virtual3p1.html.

[5] «spamassassin.apache.org,» [En línea]. Available: http://spamassassin.apache.org/full/3.3.x/doc/Mail_SpamAssassin_Conf.html.

[6] J. Sabater. [En línea]. Available: http://www.scribd.com/doc/42149057/35/SpamAssassin.

[7] F. Timme, 2007. [En línea]. Available: http://www.howtoforge.com/amavisd_postfix_debian_ubuntu.

[8] [En línea]. Available: http://www.policyd.org/.

[9] 2007. [En línea]. Available: http://www.uno-code.com/?q=node/69.

[10] [En línea]. Available: http://tuxjm.net/docs/Gentoo-Postfix+Policyd.txt.

[11] [En línea]. Available: http://2stech.ca/index.php/linux/linuxtutotials/tutorials/147-dspam-in-postfix.

[12] 2007. [En línea]. Available: http://tuxedlinux.wordpress.com/2007/09/12/integracion-de-clamav-y-dspam-a-nuestro-postfix/.