INSTALACIÓN Y CONFIGURACIÓN DE UN SERVIDOR DE CORREO MEDIANTE POSTFIX

INSTALACIÓN DEL SERVIDOR POP3 E IMAP

Habiendo considerado diferentes opciones, finalmente se realizarán las instalaciones basadas en Dovecot como servidor pop/imap. Dicho servidor está disponible en los repositorios tanto en Ubuntu como en Debian y también en otras distribuciones, y ha sido el escogido debido a su sencilla configuración.

CREACIÓN DEL FICHERO /etc/dovecot/c-client.cf

El fichero */etc/dovecot/c-client.cf* no existe, pero tiene que ser creado escribiendo las siguientes frases en él:

```
I accept the risk
set disable-plaintext nil
```

EDICIÓN DEL FICHERO DE CONFIGURACIÓN /etc/dovecot/dovecot.conf

En el fichero */etc/dovecot/dovecot.conf* se editan primero las líneas correspondientes a la configuración básica de los protocolos de los que se encarga:

```
protocols = imap pop3 pop3s imaps
imap_listen=*
pop3_listen=*
imaps_listen=*
pop3s_listen=*
disable_plaintext_auth=no
```

Tras realizar los cambios básicos en el archivo de configuración del Dovecot, se reinicia el servicio */etc/init.d/dovecot restart*

Hay que asegurarse de que no haya corriendo ningún otro servidor POP3 o IMAP editando inetd.conf. También hay que indicar que al instalar Dovecot desinstalará otros servidores POP3 e IMAP (tales como WU).

El mejor modo de comprobar que Dovecot funciona correctamente es realizando un telnet al puerto correspondiente, al puerto 110:

```
telnet 127.0.0.1 110
```

INSTALAR EL SERVIDOR SMTP

Como bien se ha indicado antes Postfix será el servidor SMTP que se utilizará para las instalaciones debido a los amplios motivos a su favor, tales como la sencillez a la hora de su configuración, su cada día mayor extensión…

Antes de nada hay que instalar el paquete Postfix de los repositorios. Una vez instalado Postfix en el equipo, son dos los archivos de configuración que habrá que tener en cuenta a la hora de configurar Postfix del modo que mejor nos convenga. La mayor parte de la configuración se lleva a cabo en el fichero main.cf, ya que la funcionalidad primordial del master.cf es la de definir cómo un programa se conecta a un servicio y que dominio corre cuando un servicio es solicitado.

La configuración básica del archivo main.cf es muy sencilla, basta con configurar las siguientes l-ines de forma correcta:

1.- Especificar el Hostname y el Dominio de la máquina:

myhostname = mail.pfc-server.com

2.- Indicar el dominio del cual llega el correo local:

myorigin = $mydomain

3.- Indicar en que interfaces estará Postfix escuchando en el puerto 25. Si no se le indica nada solo escuchará por defecto en localhost:

inet_interfaces = all

4.- Indicar la lista de dominios que la máquina considerará como destino final para el que aceptar el correo:

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

5.- Indicar cual es el fichero donde se almacenan los alias. Los alias son cuentas no reales, es decir, que no existen como tal en GNU/Linux, pero que pueden ser asociadas a una o varias cuentas de correo reales del sistema. Ejemplo del archivo alias:

sistemas: nagore, borja, andoni, agustin
desarrollo: alayn, cristina, ibon, mikel

Los alias en este caso son sistemas y desarrollo. Al enviar un mail cuyo destinatario sea sistemas, dicho mail en realidad se enviará a los usuarios listados dentro de dicho alias.

El fichero alias puede disponer de muchas entradas, con lo cual tiene que ser indexado en formato base de datos de Berkley, y hay que hacerlo cada vez que se modifique, mediante el comando newaliases. Este comando genera un fichero aliases.db que es el que usará Postfix.

alias_maps=hash:/eretc/aliases
alias_database=hash:/etc/aliases

6.- Para que los clientes de nuestro servidor de correo puedan enviar correos a través de nuestro servidor, debe de habilitarse el relay en la variable mynetworks:

mynetworks_style = subnet
mynetworks = 192.168.0.0./16, 127.0.0.0/8

En este punto cabe aclarar lo que es un open relay, ya que es algo muy peligroso que se ha de evitar a toda costa. Un open relay es cuando un servidor SMTP permite enviar correo a destinatiarios que no pertenecen a nuestro dominio. Un servidor solo ha de ser open relay para sus redes de confianza y redes locales, NUNCA para IP's que no son de nuestra red, es decir, para los que no estén en la variable mynetworks.

Si se quieren añadir más IP's de oficinas remotas o clientes que tienen IP's fijas pueden añadirse en el fichero access. Además en en fichero access también pueden indicarse la direcciones a las que específicamente no se les permiten conexiones con nuestro servidor:

Ejemplo del fichero access:

#/etc/postfix/access
 # Redes a las que se permite hacer relay

 # Se le permite hacer relay a la red interna
192.168 RELAY

 # Se deniegan direcciones de spamers conocidas
65.169.89 DENY no se aceptan mails de spammers
spammer@spammerland.com DENY no se aceptan mails de spammers
spammerland.com DENY no se aceptan mails de spammers

Tras añadir las líneas necesarias al fichero es necesario indexarlo del mismo modo que hacía falta para el fichero aliases, salvo que en este caso el mandato a ejecutar no es *newaliases* sino postmap: *postmap access*

Para que Postfix utilice el fichero access además de los valores indicados en mynetworks, deben añadirse las siguientes lineas al fichero ***main.cf***:

smtpd_recipient_restrictions = permit_mynetworks
check_relay_domains
smtpd_sender_restrictions = hash:/etc/postfix/access
reject_unknown_sender_domain

7.- Una medida antispam interesante o para restringir quien es quien envía correo no deseado son las directivas ***header_checks*** y ***body_checks*** del main.cf:

header_checks = regexp:/etc/postfix/header_checks
body_checks = regexp:/etc/postfix/body_checks

8.- Tras realizar esta configuración básica inicial se rearranca Postfix (*postfix restart*) para comprobar que los cambios realizados funcionan correctamente. Puede realizarse una prueba manual de que el correo funciona:

telnet 127.0.0.1 25
Trying 127.0.0.1…
Connected to 127.0.0.1.
Escape carácter is '^]'.
220 mail.pfc-server.com ESMTP Postfix
mail from: admin.@pfc-server.com
250 Ok
rcpt to: admin.@pfc-server.com
250 Ok
data
354 End data with <CR><LF><CR><LF>
Prueba de envío de correo manual
.
250 Ok: queued as A7CBC33A9C
221 Bye
Connection closed by foreign host.

Haciendo cat */var/spool/mail/admin* Podrá comprobarse si el correo ha llegado al usuario local admin.

9.- Modificación del sistema de buzones de Mailbox a Maildir.

La diferencia entre Mailbox y Maildir es que Mailbox guarda los mensajes en un único fichero y Maildir guarda los mensajes es una estructura de ficheros y directorios, con lo cual no requiere bloque de ficheros para mantener la integridad de los mensajes. Si bien ambos formatos están ampliamente extendidos y son eficaces, el que se utilizará será Maildir, debido en gran parte a la comodidad e independencia de los mensajes que aporta la estructuración en directorios.

Por defecto tanto Dovecot como Postfix están predeterminadas a utilizar Mailbox, con lo cual, para habilitar maildir habrá que modificar los ficheros */etc/postfix/main.cf* y */etc/dovecot/dovecot.conf*:

#main.cf
home_mailbox = Maildir/

#dovecot.conf
default_mail_env = maildir:/home/%u/Maildir

10.- Convertir los buzones ***de mailbox a maildir***

Pese a haber modificado las directrices que indican que ha cambiado el sistema de archivo, es primordial el cambiar el formato de buzones a los mensajes ya almacenados en el sistema, para ello se puede utilizar el programa mb2md.pl (mailbox to maildir), disponible en los repositorios: *aptitude install mb2md*

Una vez se tenga el script ***mb2md.pl***, deberá ejecutarse como cada usuario (no como root): *mb2md.pl –s /var/spool/mail/admin. /home/admin/Maildir*

11.- Definir Dominios Virtuales

En Postfix tiene que crearse un fichero con una tabla de usuarios virtuales. La directiva para indicarle a Postfix cuales son los usuarios es la siguiente:

 # Para los Dominios Virtuales
virtual_alias_maps = hash:/etc/postfix/virtual

La sintaxis del fichero es la siguiente:

#usuario@dominiovirtual	*usuario_local*
admin@zerbitzaria.com	*admin.zerbitzaria*
pfc-server.com	*whatever*

Esto significa que cuando se envíe un correo a la cuenta admin@zerbitzaria.com, el correo se entregará al usuario local admin.zerbitzaria. Este formato permite tener un usuario llamado admin. (perteneciente al dominio principal) y otro usuario llamado admin..zerbitzaria destinado al dominio virtual zerbitzaria.com.

Para que Postfix pueda leer el fichero virtual, éste debe de estar en forma de base de datos de Berckley, para ello se ejecutará el comando postmap virtual.

AUTENTICACIÓN SMTP CON SASL

12.- Instalación de paquetes necesarios.

Para poder garantizar la autenticación SMTP y ponerla en funcionamiento, primero hemos de asegurarnos de que en el sistema se tienen instalados los siguientes paquetes necesarios***: Postfix-tls, libsasl2, libsasl2-modules, sasl2-bin, libgsasl7, openssl, libssl0.9.7, ssl-cert, libnet-ssleay-perl***

13.- Configuración del fichero saslauthd *(/etc/default/saslauthd*)

START = yes
MECH = pam

14.- Crear el fichero */etc/postfix/sasl/smtpd.conf* con el siguiente contenido, que permite autenticar contra usuarios del sistema:

pwcheck_method: saslauthd

15.- Reiniciar el servicio */etc/init.d/saslauthd restart*

CORREO SEGURO CON SLS Y TLS

16.- Crear la autoridad de certificación y los certificados

cd /usr/lib/ssl/misc
./CA.pl –newca
cd demoCA
mv cacert.pem newca.pem
openssl x509 –in newca.pem –days 3650 –out cacert.pem signkey private/cakey.pem

Para poder ver el contenido del certificado y comprobar que se ha realizado correctamente:

Openssl x509 –text-noout < demoCA/cacert.pem

Metemos los campos X.509 del certificado de nuestra CA y firmamos el certificado, esto crea bajo el directorio demoCA estos ficheros:

cacert.pem ➔ clave pública
cakey.pem ➔ clave privada

Ahora se crea el certificado del servidor POP3 e IMAP:

./CA.pl --newreq

Se meten los campos X.509, teniendo en cuenta que cuando se solicite el campo Common Name hemos de indicar el hostname de la máquina.

Luego ya firmamos el certificado del servidor con la clave privada de nuestra CA:

./CA.pl --sign

Esto genera los siguientes ficheros:

newcert.pem ➔ certificado del servidor
newreq.pem ➔ clave privada encriptada del servidor

Ahora se debe de desencriptar la clave privada del servidor para que no nos pregunte por la palabra secreta que hemos introducido al crear el certificado:

Openssl rsa --in newreq.pem --out server.key

Renombramos el fichero certificado del servidor con el nombre ***server.crt*** y copiamos en fichero server.key como ***imapd.pem***.

mv newcert.pem server.crt
cd server.key imapd.pem

Al fichero *imapd.pem* hay que añadirle la parte del certificado de *server.crt* que comienza con *BEGIN CERTIFICATE* y finaliza con *BEGIN RSA PRIVATE KEY.*

Creamos un enlace simbólico entre imapd.pem e ipop3.pem

ln –s imapd.pem ipop3.pem
chown root.root imapd.pem
chmod 600 imapd.pem

Copiamos los dos ficheros y el cacert.pem al directorio certs:

cp imapd.pem ipop3.pem ../certs
cp demoCA/cacert.pem ../certs

17.- Modificar */etc/dovecot/dovecot.conf*

protocols = imap imaps pop3 pop3s
ssl_disable = no
ssl_cert_file = /etc/ssl/certs/imapd.pem
ssl_key_file = /etc/ssl/private/imapd.pem

Previamente habremos copiado el certificado en ese path.

NOTA: Para que nuestro cliente de correo (Thunderbird, Evolution, Outlook …) reconozca el certificado del servidor, debemos importar el certificado de la autoridad certificadora cacert.pem. Si se trata de un cliente Windows, hay que convertir el formato del fichero:

openssl base64 –d –in cacert.pem –out cacert.bin

El formato binario .bin será reconocido por Windows sin problemas.

POSTFIX CON SOPORTE TLS

18.- Habilitar el soporte TLS en Postfix en main.cf

smtpd_use_tls = yes
smtpd_tls_cert_file = /etc/postfix/postfix.pem

Indicamos donde está el fichero con el certificado del servidor Postfix

smtpd_tls_key_file = $smtpd_tls_cert_file

El fichero de la clave privada es el mismo que el del certificado para que Postfix no nos pida password

smtpd_tls_Cafile = /usr/lib/ssl/misc/demoCAs/cacert.pem

Este el el fichero con el certificado de nuestra autoridad de certificación

smtpd_tls_CApath = /usr/lib/ssl/certs

Este es el path al directorio con la lista de certificados de las diferentes CA's (autoridades de certificación).

Antes de reiniciar Postfix hay que copiar el fichero ipo3p.pem o imapd.pem como ***postfix.pem*** en el directorio /etc/postfix. Para comprobar que Postfix acepta TLS puede verse conectándose via telnet al puerto 25 y ejecutando el comando ***EHLO***:

telnet 127.0.0.1 25
Trying 127.0.0.1 …
Connected to 127.0.0.1.
Escape carácter is '^]'.
220 mail.pfc-server.com ESMTP Postfix
ehlo localhost
250-pfc-server.com
250-PIPELINING
250-SIZE 1200000
250-VRFY
250-ETRN
250-STARTLS
250-AUTH GSSAPI NTLM LOGIN PLAIN DIGEST-MD5 CRAM-MD5
250-8BITMIME

CONFIGURAR POSTFIX DE MODO AUN MÁS SEGURO

19.- Editar main.cf

```
smtpd_recipient_restrictions = permit_mynetworks
        check_client_access hash:/etc/postfix/access
        permit_sasl_authenticated
        reject_non_fqdn_recipient
        reject_unknown_sender_domain
        reject_unknown_recipient_domain
        reject_non_fqdn_hostname
        reject_unauth_destination

smtpd_sender_restrictions = hash:/etc/postfix/access
        reject_unknown_sender_domain warn_:if_reject

smtpd_client_restrictions = permit_mynetworks,
        reject_rbl_client relays.ordb.org
        reject_rbl_client sbl-xbl.spamhaus.org
        reject_rbl_client opm.blitzed.org
        reject_rbl_client dnsbl.njabl.org
        reject_rbl_client list.dsbl.org
        permit
```

NOTA IMPORTANTE: *Las restricciones incluidas tanto en smtpd_recipient_restrictions, smtpd_sender_restrictions o smtpd_client_restrictions se ejecutan por orden de aparición y es la primera que se cumple la que se ejecuta, sin hacer caso de las siguientes.*

smtpd_recipient_restrictions *se aplican al destinatario, es decir, al comando* ***RCPT TO****. Por defecto Postfix acepta los clientes cuyas direcciones coinciden con la variable $mynetworks o acepta el correo de destinos remotos que coinciden con la variable $relay_domains o correo a destinos locales que coinciden con $inet_interfaces o $proxy_interfaces, $mydestination, $virtual_alias_domains o $virtual_mailbox_domains.*

smptd_sender_restrictions *son restricciones opcionales (ninguna por defecto) que se aplican al comando* ***MAIL FROM****. El defecto es permitir cualquier cosa. En esta configuración base aplicable a cualquier caso general se asegura que el dominio del sender exista en los DNS de Internet y se verifica cualquier otra restricción indicada en el fichero access.*

smtpd_client_restrictions *son restricciones opcionales que se aplican a la solicitud de conexión de un cliente SMTP. El defecto es permitir cualquier conexión, pero en este caso, se permite la conexión de mynetworks y del resto pero tras consultar con las listas negras.*

20.- Usando la directiva *smtpd_sender_restrictions* se pueden limitar los mensajes cuyas direcciones sean las de spammers conocidos. Para ello basta con bajarse el fichero ***sa-blacklist.current.reject*** del proyecto http://www.sa-blacklist.stearns.org/sa-blacklist/, que pese a no ser totalmente eficiente ya que a diferencia de *Spamhaus* éste no filtra por IPs, siempre ayuda a evitar a los temidos spammers.

21.- Creación de un script (sa-blacklist) que se baja el fichero, crea el mapa y reinicia Postfix:

```
#/bin/sh

cd /etc/postfix
wget http://www.sa-blacklist.stearns.org/sa-blacklist/sa-blacklist.current.reject
cp –f sa-blacklist.current.reject sender_restrictions
/usr/sbin/postmap sender_restrictions
rm –f sa-balcklist.current.reject

Postfix reload
```

Poniéndolo en un *cron*, puede realizarse su actualización con la periodicidad que se quiera:

```
0    7    *    *    1,3,5  /etc/postfix/sa-blacklist > /dev/null 2>&1
```

ANTISPAM-SPAMASSASSIN

Instalación de los paquetes necesarios: ***Spamassassin y spamc***

Los ficheros de configuración a tener en cuenta son los siguientes:

- */etc/Spamassassin/local.cf* → fichero de configuración principal
- *~/.Spamassassin/user_prefs* → fichero de configuración de los usuarios
- */usr/share/Spamassassin* → directorio donde están los ficheros con las reglas (ficheros con la extensión .cf)

El fichero de configuración principal es ***local.cf*** y estas son sus opciones básicas de configuración:

- ***required_score [num]*** → Score o puntuación total. Si las reglas aplicadas suman más que el valor fijado aquí, del mensaje será considerado spam. Valores más altos aquí evitan falsos positivos, pero también pueden permitir que entre más spam sin marcar.

- ***report_safe 1*** → indica como Spamassassin modifica los mensajes catalogados como spam. Si se activa esta opción, Spamassassin añade tres cabeceras:
  - X-Spam-Level → con * que representan la puntuación
  - X-Spam-Status → línea con la descripción del spam y los test coincidentes
  - Adjunto MIME → el informe del spam

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

- ***Rewrite_header Subject [SPAM]*** → Rescribe la línea de asunto y le añade el texto [SPAM]. Esto permite marcar los mensajes como spam y así poder poner un filtro en el cliente de correo para mover los mensajes de spam a una carpeta destinada a ello.

SPAMASSASSIN EN MODO DEMONIO SPAMD/SPAMC

Spamassassin es un script en perl. Si se tienen bastantes mensajes, el proceso de arranque del intérprete de perl cada vez puede sobrecargar mucho al servidor. Como alternativa existe la versión demonizada Spamd. Correo como demonio y usa módulos de perl que se cargan en el proceso de arranque del demonio. El cliente Spamc se invoca con cada mensaje filtrado y se los pasa al deminio spamd.

Antes de ejecutar spamd ha de crearse un usuario filter: ***adduser filter***

Editar el fichero */etc/default/Spamassassin*

 # Change to 1 to enable spamd

ENABLED=1
OPTIONS="—create-prefs –max-children 5 –helper-home-dir –u filter"
PIDFILE="/var/run/spamd.pid"

Para arrancar spamd: */etc/init.d/Spamassassin start*

INTEGRAR SPAMASSASSIN CON POSTFIX

El modo más sencillo de integrar Spamassassin con Postfix es hacerlo a modo de filtro externo. Así, el filtro antispam se ejecuta para todos los usuarios y no hace falta crear ficheros .procmail para cada usuario. Para ello hay que añadir lo siguiente al master.cf:

smtp inet n - n - - smtpd content_filter=spamfilter
……..
……..
Spamfilter unix - n n - - pipe user=filter
argv=/usr/bin/spamc –f –e /usr/sbin/sendmail –oi –f ${sender} ${recipient}

AJUSTES EN SPAMASSASSIN

REGLAS DE SPAMASSASSIN

Para realizar el chequeo del spam, Spamassassin realiza unos tests basándose en los grupos de reglas que hay en los ficheros del directorio */usr/share/Spamassassin*. Así mismo, los scores de los tests son almacenados en un único fichero.

La estructura de los fichero consiste de un nombre de test, una descripción y la acción que se puede aplicar en la cabecera, el cuerpo del mensaje y el score.

LISTAS BLANCAS Y LISTA NEGRAS

Se le puede indicar a Spamassassin que algunas direcciones no sean nunca marcadas como spam. Se le indica con la directiva ***whitelist_form*** en el fichero local.cf:

whitelist_form nagore@pfc-server.com
*whitelist_from pfc-server.com, *pfc-server.com*
*whitelist_form *@pfc-server.com*

Otra posibilidad muy útil es justamente la opuesta, esto es, la de las listas negras. En ella se listan los spammers conocidos mediante la directiva **blacklist_from**:

blacklist_from 0-sexshop.com 001bastconsumer.com

ENTRENAMIENTO USANDO LOS FILTROS BAYESIANOS

Una de las partes más importantes para que un sistema antispam sea efectivo es el del aprendizaje. Que Spamassassin tan solo se base en las mismas reglas hace que sea un antispam estático al que a la postre se pueda “engañar” con facilidad.

Para que el entrenador bayesiano funcione se le deberán mostrar tanto mensajes que son spam como los que no lo son. Hay dos estrategias para entrenar a Spamassassin: la de entrenar con cada mensaje o la de hacerlo con los errores cometidos. La primera es muy eficiente pero demasiado exigente ya que hay que entrenarle con todos los correos. La segunda no es tan eficiente ya que tan solo se le indica qué es spam y qué no lo es, es decir, se le corrigen los falsos positivos y los falsos negativos. Es eficiente pero no responde tan rápido a los cambios en los patrones del spam.

Mediante el script sa-learn se entrena a Spamassassin mediante los correos almacenados en la carpeta SPAM y los correos almacenados en la carpeta HAM.

sa-learn --spam --mbox /var/spool/mail/spam → le indicamos que aprenda como spam el correo de la cuenta spam en formato mbox

sa-learn --ham --mbox /var/spool/mail/ham → le indicamos que aprenda como ham el correo de la cuenta ham en formato mbox.

Ejecutar el script a mano cada vez que quisiéramos que el Spamassassin aprendiera resultaría tedioso y poco efectivo, es por eso que se crean dos scripts, uno para que aprenda del spam y otro para que aprenda del ham:

```
#!/bin/sh
# Script para que Spamassassin aprenda lo que es spam

sa-learn --spam --mbox /var/spool/mail/spam
rm –f /var/spool/mail/spam
cd /var/spool/mail
touch spam
chown spam:mail spam
/etc/init.d/Spamassassin restart
```

```
#!/bin/sh
# Script para que Spamassassin aprenda lo que no es spam
sa-learn --ham --mbox /var/spool/mail/ham
rm –f /var/spool/mail/spam
cd /var/spool/mail
touch ham
chown ham:mail ham
/etc/init.d/Spamassassin restart
```

ANTIVIRUS MAILSCANNER Y ANTIVIRUS CLAMAV

Primero se instalan los paquetes necesarios: ***clamav y mailscanner***

Para integrarlo con Postfix hay que editar el ***main.cf*** de /etc/postfix y habilitar los siguiente:

header_checks = regexp:/etc/postfix/header_checks

Hay que crear el fichero */etc/postfix/header_checks* con la siguiente línea:

/^Received:/HOLD #Hace que los mensajes vayan a la cola hold

En el fichero */etc/MailScanner/MailScanner.conf* hay que editar lo siguiente:

Run As User = Postfix
Run As Group = Postfix
Incoming Queue Dir = /var/spool/postfix/hold
Outgoing Queue Dir = /var/spool/postfix/incoming
MTA = Postfix
Virus Scanners = Clamav

Hay que hacer que el propietario del directorio sea Postfix:
Chown –R Postfix:Postfix /var/spool/MailScanner
Chown –R Postfix:Postfix /var/lock/subsys/MailScanner
Chown –R Postfix:Postfix /var/lib/MailScanner
Chown –R Postfix:Postfix /var/run/MailScanner
Postfix stop

Editamos el fichero */etc/default/mailscanner* y activar lo siguiente:

Run_mailscanner=1

Y reiniciamos el servicio: /etc/init.d/mailscanner restart

Para integrar MailScanner con Spamassassin en el fichero */etc/MailScanner/MailScanner.conf*:

Use Spamassassin = yes
Spamassassin User State Dir = /var/spool/MailScanner/Spamassassin

Mkdir /var/spool/MailScanner/Spamassassin
Chown Postfix:Postfix /var/spool/MailScanner/Spamassassin

Para integrarlo con ClamAV en el MailScanner.conf ponemos:

Spam Checks = no # porque Postfix ya ejecuta el Spamassassin
Virus Scanners = clamav
Monitors for ClamAV Updates = /var/lib/clamav/.cvd*

De este modo, aunque se detecte un virus al ser independientes el antivirus y el antispam, un mensaje entra primero en mailscanner y al salir de allí pasa de todos modos por Spamassassin.

INSTALACIÓN Y CONFIGURACIÓN DE SQUIRRELMAIL EN MODO SEGURO

Como en casi todas las instalaciones, bajar el paquete adecuado desde los repositorios vuelve a ser el primer paso:

aptitude install squirremail

Con la instalación de Squirrelmail se instalará también el paquete squirrelmail-locales. Si por cualquier cosa no se instalara, habrá que descargarlo de los repositorios de todos modos.

Una vez instalado el webmail, hay que proceder a su correcta configuración. Lo primero será habilitar Squirrelmail en modo seguro, para ello el modo más sencillo es crear un enlace simbólico en ***sites-enabled*** (sitios habilitados) de apache para squirelmail. Para ello se hace el link hacia /etc/squirrelmail/apache.conf desde /etc/apache2/sites-enables/squirrelmail.

ln -s /etc/squirrelmail/apache.conf /etc/apache2/sites-enabled/squirrelmail

Hay que editar /etc/apache2/sites-available/default y que el inicio sea el siguiente:

```
NameVirtualHost *:80
<VirtualHost *:80>
```

Hay que editar /etc/apache2/sites-available/https tambien, para que funcione en modo seguro. Para ello el archivo debe editarse para que inicie así:

```
NameVirtualHost *:443
<VirtualHost *:443>
    SSLEngine on
    SSLCertificateFile /etc/apache2/ssl/apache.pem
```

Editar /etc/squirrelmail/apache.conf para que se muestre de la siguiente forma:
El archivo apache.conf de squirrelmail tiene la siguiente forma:

```
Alias /squirrelmail /usr/share/squirrelmail
<Directory /usr/share/squirrelmail>
  php_flag register_globals off
  Options Indexes FollowSymLinks
  <IfModule mod_dir.c>
    DirectoryIndex index.php
  </IfModule>

  # access to configtest is limited by default to prevent information leak
  <Files configtest.php>
    order deny,allow
    deny from all
    allow from 127.0.0.1
  </Files>
</Directory>

# users will prefer a simple URL like http://webmail.example.com
#<VirtualHost 1.2.3.4>
#  DocumentRoot /usr/share/squirrelmail
#  ServerName webmail.example.com
#</VirtualHost>

# redirect to https when available (thanks omen@descolada.dartmouth.edu)
#
#  Note: There are multiple ways to do this, and which one is suitable for
#  your site's configuration depends. Consult the apache documentation if
#  you're unsure, as this example might not work everywhere.
#
#<IfModule mod_rewrite.c>
#  <IfModule mod_ssl.c>
#    <Location /squirrelmail>
#      RewriteEngine on
#      RewriteCond %{HTTPS} !^on$ [NC]
#      RewriteRule . https://%{HTTP_HOST}%{REQUEST_URI}  [L]
#    </Location>
#  </IfModule>
#</IfModule>
```

Hay que asegurarse de que la línea DirectoryIndex en /etc/apache2/apache2.conf aparezca lo siguiente:

```
DirectoryIndex index.html index.htm index.shtml index.cgi index.php index.php3 index.pl index.xhtml
```

Si no se ha hecho previamente en la configuración de apache, habrá que editar /etc/apache2/ports.conf para que escuche también o solo en el puerto 443:

```
Listen 80
Listen 443
```

Por defecto Squirrelmail estará en ingles. Si se quiere que esté en castellano, teniendo squirrelmail-locales, basta con añadir la línea correspondiente en /var/lib/locales/supported.d/local.

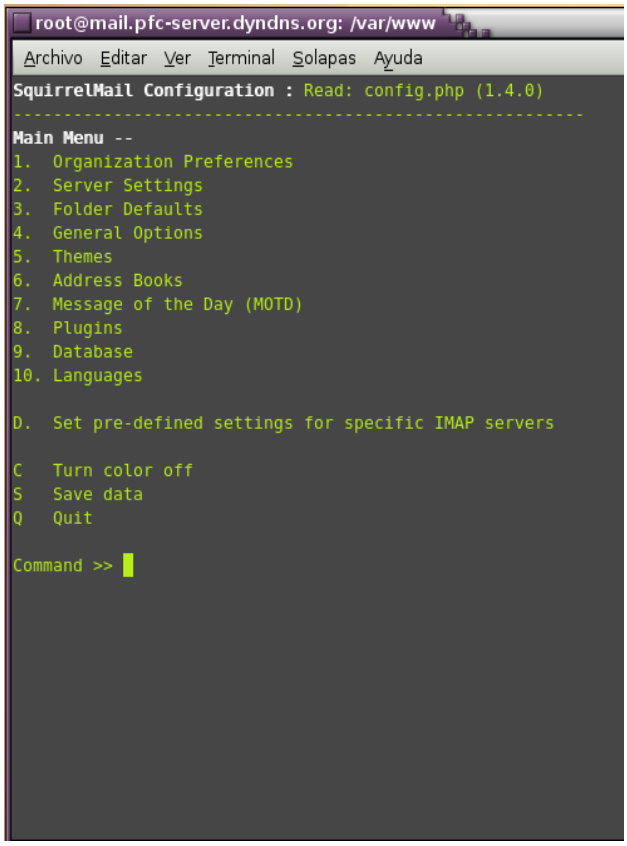
```
en_US.ISO-8859-1                ISO-8859-1
en_US.UTF-8                     UTF-8
es_ES.UTF-8                     UTF-8
es_ES.ISO-8859-1                ISO-8859-1
```

Tras este cambio ya existirá la opción de cambiarle el idioma a squirrelmail una vez reconfiguradas las locales:

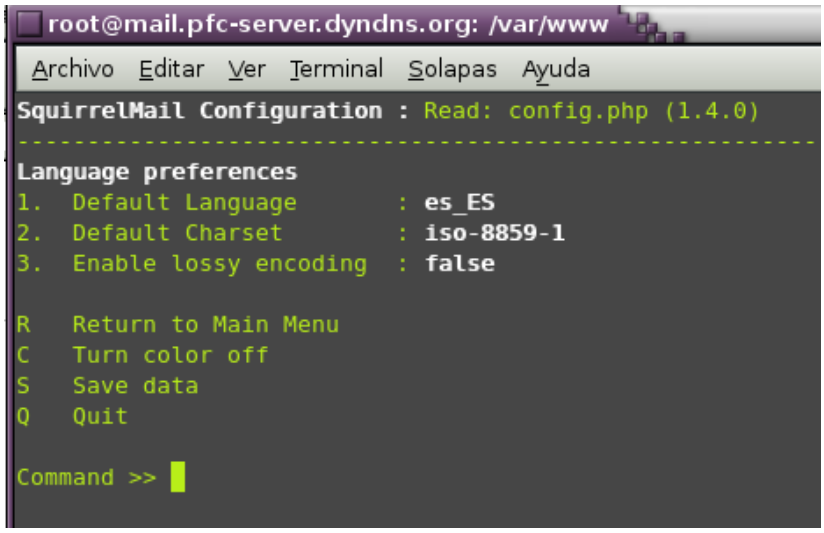
dpkg-reconfigure locales

Squirrelmail tiene múltiples opciones de configuración, y es el momento de configurarlas. Para ello tras ejecutar el siguiente comando se accede a las opciones de configuración y personalización de Squirrelmail:

squirrelmail-configure



En ***lenguages*** podrá seleccionarse el idioma con el que funcionará la aplicación:



Como se puede observar en las opciones generales, son muchas las cosas que pueden modificarse y personalizarse en Squirrelmail, pero una vez llegados a este punto, no cabe hacer aquí una explicación más extensa de cada una de ellas ya que en cada caso las opciones diferirán.

Una vez terminada la configuración bastará con acceder a la interfaz de Squirrelmail desde el navegador y acceder al mismo con usuario y contraseña del correo:

https://localhost/webmail