Instalación y administración de Servicio de Correo Electrónico en Debian

Sumario

Introducción al Correo Electrónico.

El correo electrónico, se basa en procedimientos operativos mucho más complicados que los de los servidores web. Para la mayoría de los usuarios, el funcionamiento es transparente, lo cual significa que no es necesario entender cómo funciona el correo electrónico para poder utilizarlo.

Sin embargo, a continuación se ofrece una breve introducción para ayudar a los usuarios a entender sus principios básicos, darles una idea de cómo configurar mejor los clientes de correo electrónico e informarles sobre los mecanismos subyacentes del spam.

¿Cómo funciona el correo electrónico?

El correo electrónico gira alrededor del uso de los buzones de correo electrónico. Cuando se envía un correo electrónico, el mensaje se enruta de servidor a servidor hasta llegar al servidor de correo electrónico del receptor. Más precisamente, el mensaje se envía al servidor del correo electrónico (llamado **MTA**, del inglés **Mail Transport Agent** - Agente de Transporte de Correo) que tiene la tarea de transportarlos hacia el MTA del destinatario. En Internet, los MTA se comunican entre sí usando el protocolo **SMTP**, y por lo tanto se los llama **servidores SMTP** (o a veces servidores de correo saliente).

Luego el MTA del destinatario entrega el correo electrónico al servidor del correo entrante (llamado **MDA**, del inglés **Mail Delivery Agent** -Agente de Entrega de Correo), el cual almacena el correo electrónico, mientras espera que el usuario lo acepte. Existen dos protocolos principales utilizados para recuperar un correo electrónico de un MDA:

- **POP3** (Post Office Protocol - Protocolo de Oficina de Correo), el más antiguo de los dos, que se usa para recuperar el correo electrónico y,

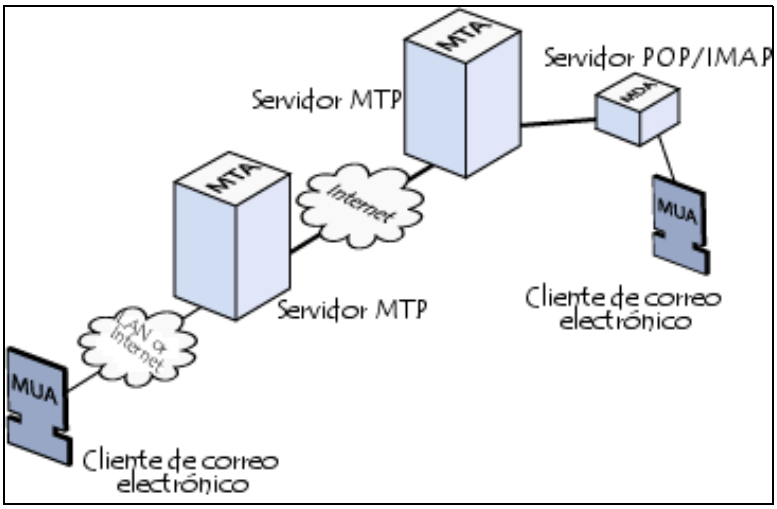
en algunos casos, dejar una copia en el servidor.

- **IMAP** (Internet Message Access Protocol -Protocolo de Acceso a Mensajes de Internet), el cual se usa para coordinar el estado de los correos electrónicos (leído, eliminado, movido) a través de múltiples clientes de correo electrónico. Con IMAP, se guarda una copia de cada mensaje en el servidor, de manera que esta tarea de sincronización se pueda completar.

Por esta razón, los servidores de correo entrante se llaman servidores POP o servidores IMAP, según el protocolo usado.

Diagrama de funcionamiento de los MTA MDA MUA

Usando una analogía del mundo real, los **MTA** actúan como la oficina de correo (el área de clasificación y de transmisión, que se encarga del transporte del mensaje), mientras que los **MDA** actúan como casillas de correo, que almacenan mensajes (tanto como les permita su volumen), hasta que los destinatarios accedan a su casilla. Ésto significa que no es necesario que los destinatarios estén conectados para poder enviarles un correo electrónico.



Para evitar que cualquiera, lea los correos electrónicos de otros usuarios, el MDA está protegido por un nombre de usuario llamado registro y una contraseña.

La recuperación del correo se logra a través de un programa de software llamado **MUA** (Mail User Agent - Agente Usuario de Correo).

Cuando el **MUA** es un programa instalado en el sistema del usuario, se llama cliente de correo electrónico (tales como Mozilla Thunderbird, Microsoft Outlook, Eudora Mail, Incredimail o Lotus Notes).

Cuando se usa una interfaz de web para interactuar con el servidor de correo entrante, se suele llamar webmail.

Relé abierto

Por defecto, y por razones históricas, antes no era necesario autenticar la propia identidad, para enviar un correo electrónico, lo cual significa que era muy fácil falsificar direcciones cuando se enviaba un correo.

Por esta razón, hoy en día, casi todos los proveedores de servicio de Internet bloquean sus servidores SMTP para que sólo sus suscriptores puedan usarlos, o más precisamente sólo las máquinas cuyas direcciones IP pertenezcan al dominio del ISP. Ésto explica la razón por la cual los usuarios que viajan, deben modificar la configuración del servidor de salida de sus clientes de correo electrónico, para que les permita enviar correos desde ISP distintos al suyo propio.

Cuando el servidor de correo electrónico de una organización está mal configurado y permite que terceros, en cualquier red, envíen correos electrónicos, ésto se denomina **relé abierto**.

Generalmente los relés abiertos son usados por los spammers, ya que al hacerlo, esconden el verdadero origen de sus mensajes. Como resultado, muchos ISP mantienen una lista negra actualizada de relés abiertos, para evitar que los suscriptores reciban mensajes de tales servidores.

Configuración de registros DNS para el servidor de correo.

Antes de nada comprobaremos que los datos de nuestro equipo son correctos:

```
# Comprobaremos el nombre del equipo:
# Por ejemplo servidor:
nano /etc/hostname
servidor

# Comprobaremos el fichero hosts:
nano /etc/hosts

# Contendrá algo similar a:
127.0.0.1       localhost
127.0.1.1       servidor.google.local     servidor
....

# Instalaremos el servidor de DNS:
aptitude install bind9

# o también:
apt-get install bind9

# Comprobaremos la configuración de nuestro resolver:
nano /etc/resolv.conf

# Contenido de ejemplo:
domain google.local
search google.local
nameserver 10.23.2.0
```

A la hora de instalar nuestro servidor de correo, tendremos que crear los siguientes registros en el DNS:

- Un registro de tipo A que apunte a la dirección del servidor de correo.
- Un registro de tipo PTR, para que funcione la resolución inversa (necesaria por algunos sistemas para controlar el envío de SPAM por servidores SMTP.
- Un registro de tipo MX con la prioridad que deseemos, y que apunte al equipo servidor de correo de nuestro dominio.

Cómo instalar un MTA (Postfix).

- Vamos a ver cómo podemos instalar un MTA (Postfix) con soporte SASL.

```
# Para instalar todos los paquetes mencionados anteriormente lo haremos con:
apt-get install postfix  libpq5 ssl-cert

# Durante la instalación, postfix nos preguntará algunas cuestiones como el nombre del servidor,
# y el modelo de servidor que queremos para Postfix. Seleccionaremos Internet Site.

# El fichero de configuración de Postfix se encuentra en: /etc/postfix/main.cf

# Para reiniciar el servidor Postfix, lo haremos con el comando:
service postfix restart

# El fichero de logs está en:
/var/log/mail.log
```

- Configuraciones a revisar en Postfix:

```
# Editaremos el fichero de configuración en Postfix:
nano /etc/postfix/main.cf

# Comprobaremos los siguientes parámetros:
# En este caso está configurado para un dominio pruebas.local
myhostname = servidor.pruebas.local
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = servidor.pruebas.local, localhost.pruebas.local, pruebas.local, localhost
```

```
inet_interfaces = all

# Editaremos el fichero /etc/mailname, para escribir el nombre del dominio que queremos
# que aparezca por defecto en los correos:
nano /etc/mailname

# Contendrá algo como:
pruebas.local

# Reiniciamos el servicio Postfix para aplicar los cambios:
service postfix restart
```

Cómo enviar e-mail desde la línea de comandos, de forma local.

- Envío de correos usando comandos Linux desde un terminal, en el propio servidor SMTP:

```
# Ejemplo de Envío de un correo al usuario antonio desde la línea de comandos,
# Utilizaremos el comando mail.

# Para terminar la redacción del contenido del correo, se hará
# escribiendo un punto en una nueva línea, y pulsando Enter.

mail antonio
Subject: Saludos
Hola que tal te va todo?
Saludos
.
Cc:
```

- Envío de correos usando una conexión telnet directamente con el servidor SMTP:

```
# Nos conectamos al puerto 25 del servidor SMTP:
telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 servidor.pruebas.local ESMTP Postfix (Debian/GNU)

# Una vez conectados nos tenemos que identificar:
# para ello usamos el comando HELO seguido de nuestro nombre de usuario o e-mail.
# Nota: Los comandos pueden ir tanto en mayúsculas, como en minúsculas.
HELO root
250 servidor.pruebas.local

# Indicamos el remitente del correo con MAIL FROM:
MAIL FROM: root
250 2.1.0 Ok

# Indicamos el destinatario del correo con RCPT TO:
RCPT TO: antonio
250 2.1.5 Ok

# Para escribir el contenido del mensaje tecleamos DATA y pulsamos ENTER:
# Para escribir el asunto del mensaje tecleamos SUBJECT: dentro de la sección DATA
# y una vez escrito el asunto pulsamos ENTER 2 veces y ya escribimos el contenido del correo.
# Para terminar el mensaje, tendremos que teclear un . (punto) y pulsar ENTER.
DATA
354 End data with <CR><LF>.<CR><LF>
SUBJECT: Saber que tal vas...

Hola Antonio, qué tal va todo por ahí?
Saludos.
.

# El mensaje ha sido puesto en cola para envío:
250 2.0.0 Ok: queued as 9F1EB2E530

# Nos desconectamos y cerramos la sesión con QUIT.
quit
221 2.0.0 Bye
Connection closed by foreign host.
```

Si queremos comprobar los **mensajes que están en la cola de envío del servidor SMTP**:

```
# Tecleamos el siguiente comando para comprobar los mensajes en cola de envío:
mailq

# Si queremos eliminar todos los mensajes pendientes en la cola:
postsuper -d ALL
```

Cómo chequear el e-mail desde la línea de comandos, de forma local.

Para comprobar si hemos recibido e-mail de forma local, lo podremos hacer con los siguientes comandos:

```
# Teclearemos el comando mail:
mail

# Se nos mostrará todos los correos nuevos con un número identificador de mensaje:
Mail version 8.1.2 01/15/2001.  Type ? for help.
"/var/mail/usuario": 3 messages 3 new
>N  1 root@servidor.pru  Thu Jan 26 12:46   14/489   hola
 N  2 root@servidor.pru  Thu Jan 26 12:46   14/498   Saludos
 N  3 root@servidor.pru  Thu Jan 26 12:47   14/505   Tercer correo de prueba

# Tecleando el número leeremos el e-mail directamente:
& 1

Message 1:
From root@servidor.pruebas.local  Thu Jan 26 12:46:46 2012
X-Original-To: root
To: root@servidor.pruebas.local
Subject: hola
Date: Thu, 26 Jan 2012 12:46:46 +0100 (CET)
From: root@servidor.pruebas.local (root)

que tal todo

# Para borrar todos los mensajes se pone:
& d*

# Para borrar un mensaje X en particular:
& dX

# Para más ayuda sobre comandos:
& d?
```

Configuración de redes desde las que se puede usar el servidor de correo postfix.

Postfix por defecto, dejará enviar e-mail desde clientes externos (como outlook, thunderbird...) a todos los usuarios que estén conectados a las mismas redes a las que está conectado el servidor Postfix.

ATENCION: Con esta configuración, a la hora de enviar correo, no se validará el usuario y la contraseña del usuario que está enviando correo, con lo que cualquiera podría enviar correos y hacerse pasar por otras persona, y además ésto es un agujero de seguridad que podría convertir nuestro servidor en un Relé Abierto.

Lo que no permite Postfix, es enviar correos a dominios externos como gmail, etc.. (nos daría un mensaje de **Relay Denied** a la hora de enviar correo), a no ser que nuestra red esté especificada en **mynetworks** como una **red confiable**.

En cambio sí que nos dejará enviar correos desde el propio servidor, ya que aparece como red confiable 127.0.0.0/8 en el parámetro **mynetworks**.

```
# En el fichero de configuracion de postfix /etc/postfix/main.cf
# Se configurarán las redes desde las que permitiremos el envío de correos:
nano /etc/postfix/main.cf

# Aquí se citan las redes confiables (aquellas desde las que se pueden enviar correos internos
# o externos sin validación previa de usuario y contraseña.
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
```

Para dar más seguridad y autenticación en el envío de correos, tendríamos que configurar SASL.

Creación de alias de usuarios.

A veces puede ser interesante redirigir el correo que recibimos a un usuario en particular. Por ejemplo todos los e-mails que vayan a postmaster, root, etc.. los podríamos redirigir al usuario antonio.

Para ello es necesario crear alias. Eso se hace modificando un archivo localizado en /**etc**/**aliases**:

```
nano /etc/aliases
```

Este archivo contiene alias, es decir, equivalencias entre una dirección local (probablemente ficticia) y una dirección real. Así, si el servidor recibe un mensaje dirigido a "postmaster@pruebas.local", y en /etc/aliases hay una línea como ésta:

```
postmaster:   root
```

(como, de hecho, hay), será root quien realmente reciba el mensaje. El archivo aliases ya contiene algunas líneas comunes. La única línea que puede interesar añadir es la que redirige el correo de root a un usuario normal (que será la que habitualmente utiliza el administrador cuando no precisa privilegios de supervisión). La línea sería, simplemente:

```
root:   antonio

# Se pueden poner más de un alias:
# Todo el correo que llegue a root, lo envías a antonio, manuel y laura.
root:   antonio  manuel   laura
```

Para más información sobre alias, aprender cómo establecer más de un destino para cada alias, enviar el mensaje a un archivo, etc.

```
man aliases

# Para que  aliases  sea tenido en cuenta por Postfix, en  /etc/postfix/main.cf
# debe haber una línea del tipo:
alias_maps = hash:/etc/aliases

# Cada vez que se modifica el archivo aliases debe ejecutarse el siguiente comando (como root):
newaliases

# Este comando genera el archivo  /etc/aliases.db , que es una versión indexada de aliases,
# para mejorar el acceso durante la ejecución de Postfix.
```

Cómo instalar un MDA (Dovecot).

Dovecot es un MDA (Mail Delivery Agent), con soporte POP3/IMAP, el cuál necesita de un MTA como Postfix, para funcionar correctamente.

Dovecot soporta:

- IMAP (including TLS/STARTTLS)
- IMAP+SSL ("imaps")
- POP3 (including TLS/STLS)
- POP3+SSL ("pops")

Autenticación:

Dovecot soporta los siguientes métodos de autenticación:

- plain
- digest-md5
- cram-md5
- apop (only for POP3)
- anonymous

```
#Para instalarlo ejecutaremos el siguiente comando en Debian:
aptitude install dovecot-imapd dovecot-pop3d dovecot-common

# El fichero de configuración de Dovecot se encuentra en:
```

```
/etc/dovecot/dovecot.conf

# Para chequear la versión de Dovecot instalada teclear:
dovecot --version

# Antes de continuar, necesitamos realizar algunos cambios en la configuración de dovecot.

#######################################################
# Modificaciones a realizar en versiones anteriores a la 2.0.15:
#######################################################

# Comprobaremos que los siguientes valores están correctamente en el fichero:
nano /etc/dovecot/dovecot.conf

# Comprobamos los protocolos que queremos activar a través de Dovecot:
# specify protocols = imap imaps pop3 pop3s
protocols = imap imaps pop3 pop3s

# Descomentamos la siguiente línea y le ponemos valor a no:
disable_plaintext_auth = no

# Debido a un bug en la versión 1.2.15 que estamos utilizando, tenemos que indicarle
# a Dovecot dónde se encuentran las carpetas de mail de los usuarios para el envío
# y para la recepción:
mail_location = mbox:~/mail:INBOX=/var/mail/%u

# Comprobamos que esta línea esté descomentada, para que clientes como Outlook 2003
# o Thunderbird funcionen ok.
pop3_uidl_format = %08Xu%08Xv

# Por último añadimos el siguiente parámetro:
mail_access_groups = mail

# De esta forma eliminiamos un bug que existe con las cuentas recién creadas:

# Para reiniciar el servicio Dovecot, lo haremos con el comando:
service dovecot restart

# IMPORTANTE: Acordaros que si a la hora de enviar correo desde un cliente os da un error,
# deberéis aceptar el certificado previamente en el cliente para que
# el envío funcione correctamente.

#######################################################
# Modificaciones a realizar en versión 2.0 o superior:
#######################################################


# Los ficheros de configuración en esta versión se encuentran en:
/etc/dovecot

# Comprobaremos que los siguientes valores están correctamente en el fichero:
nano /etc/dovecot/dovecot.conf

# Añadiremos lo siguiente al final del fichero de configuración:
pop3_uidl_format = %08Xu%08Xv
mail_location = mbox:~/mail:INBOX=/var/mail/%u
mail_access_groups = mail

# Reiniciamos dovecot:
service dovecot restart
```

A partir de este momento podemos usar un cliente de correo para comprobar si funciona o no el acceso. Simplemente introduciendo el usuario con la contraseña en la configuración, debería dejarnos acceder.

Recuerda que por ahora sólo serás capaz de enviar correos dentro de tu propia red o dominio local. Si intentas enviar correo al exterior obtendrás el mensaje "relay access denied", pero no tendrás ningún problema para recibir correos.

Para poder enviar correos a dominios externos, tendremos que configurar la autenticación vía SASL.

Tres apartados más abajo tienes indicados los pasos a seguir para dicha configuración.

Ejemplo de una conexión a POP3 usando Telnet

Podemos leer nuestro correo en una sesion de telnet. ¿Utilidad? piensa por ejemplo en la posibilidad de ver una lista de todos los mensajes nuevos en el servidor, antes de descargarlos, o borrarlos alli sin necesidad de bajarlos, o incluso consultar el correo desde cualquier ordenador, sin necesidad de configurar el programa de mail.

Para ello iniciaremos una sesion telnet, pero no por el puerto habitual, sino por el específico **POP3 (110)** que es donde escucha el servidor de correo. Toma nota de que esto sirve para los e-mails ordinarios, pero no para el correo-web (hot-mail y similares).

Una vez conseguida la conexión con el servidor de correo, nos indica que está listo para recibir comandos, por ejemplo:

telnet servidor pop3 o bien **telnet servidor 110**

+OK POP3 server ready

Iniciamos la sesion introduciendo nuestro nombre de usuario y password:

USER nombre_usuario

PASS password

Y obtendras contestación del servidor acerca de si has sido o no autenticado. Ten en cuenta que el servidor puede dar respuesta positiva aunque metas un nombre de usuario inexistente. Solo cuando recibas respuesta positiva a ambos, nombre y contraseña, podrás operar con el servidor.

En la especificación POP3 los comandos son una sola palabra (de tres o cuatro letras maximo), que puede ir seguida o no de argumentos. El comando y su argumento irán separados por un solo espacio.

Cada comando introducido obtendrá una respuesta del servidor, que será positiva o negativa. La respuesta puede ir o no seguida de una explicación

+OK comando introducido correctamente -ERR comando erróneo

Cuando el servidor emite a nuestro comando una respuesta con varias lineas, la ultima linea consistirá en un punto ("."). Y en el caso de que alguna de las lineas de la respuesta a su vez comenzase con un punto, se añade un punto adicional para distinguirla de la ultima linea. comandos utiles:

- **STAT** (status) solicita el estado de tu buzón de correos. El servidor responderá informando de cuantos mensajes hay a la espera, en el siguiente formato: +OK mm bb, donde mm es el numero de mensajes, y bb el numero de bytes del total.
- **LIST** te lista todos los mensajes (identificador más el tamaño). Puedes ejecutarlo solo (ofrecera el numero total de mensajes) o con un argumento (numero de mensaje) y solo obtendras como respuesta el tamaño de ese mensaje:

```
LIST
+OK 2 messages (320 octets)
1 120
2 200
.

LIST 2
+OK 2 200

LIST 3
-ERR no such message, only 2 messages in maildrop
```

- **TOP nn nl** para ver las cabeceras y primeras lineas del mensaje (nn sería el numero del mensaje que quieras ver, nl el numero de lineas de la cabecera, p ej: TOP 1 ALL)
- **RETR #** para ver un mensaje, debe especificarse su numero en la lista
- **DELE #** borra el mensaje elegido. El borrado no es al enviar el comando, sino al terminar la sesión
- **RSET** recupera los mensajes marcados para borrado
- **NOOP** (No Operation) instruye al servidor para que no ejecute ninguna acción, salvo responder con un mensaje de confirmación (+OK).

- **UIDL** (Unique Identifier List) sirve para asignar un identificador unico a todos los mensajes o a uno especifico.
- **APOP** (Authenticate Post Office Protocol) Este comando puede ser usado como sustituto del binomio USER - PASS para identificar y validar un usuario. Su utilidad es evitar que el password del usuario viaje por la red de forma no encriptada. La sintaxis es: APOP (nombre) (codigo).
- **QUIT** cierra la conexión. Si se cierra la sesión sin este comando, los mensajes marcados para borrado no se destruiran.

Instalación de un MUA web (Squirrelmail).

Antes de instalar Squirrel Web Mail, tenemos que asegurarnos de haber instalado apache2 con soporte PHP:

```
# Instalamos el servidor web apache2 y los módulos de PHP:
aptitude install apache2 libapache2-mod-php5 php5-cli php5-common php5-cgi

# Instalamos squirrelmail:
aptitude install squirrelmail

# La configuración de Squirrelmail está localizada en: /etc/squirrelmail

# Ejecutamos la utilidad de configuración de squirrelmail como root:
# Ahí podemos cambiar propiedades como el mensaje del día, preferencias de la organización, etc..
/usr/sbin/squirrelmail-configure

# Ahora configuramos para que squirrelmail funcione en apache.
# Editamos el fichero de configuración de Apache2:
nano /etc/apache2/apache2.conf

# e insertamos la siguiente línea al final del documento:
Include /etc/squirrelmail/apache.conf

# Reiniciamos el servicio usando el siguiente comando:
service apache2 restart
```

Para comprobar el correo vía webmail, lo haremos en el siguiente enlace: http://IP-dominio/squirrelmail

Probamos a acceder con alguno de los usuarios que tenemos creados en el sistema y a enviarnos correos a través de squirrelmail a ver si llegan correctamente.

Instalación de un MUA local (Mozilla Thunderbird).

Como clientes de correo (MUA), podemos usar diferentes aplicaciones. Desde Outlook Express, Windows Live Mail, Microsoft Outlook o Mozilla Thunderbird, por citar algunos ejemplos.

En la siguiente dirección os pongo un tutorial de instalación de Mozilla Thunderbird, en Windows XP.

Video Tutorial de Instalación de Mozilla Thunderbird

Autenticación SASL con TLS.

Si queremos que nuestro servidor de correo permita enviar correo a dominios externos al nuestro, tendremos que configurar la autenticación SASL.

SASL: Siglas en inglés para **Simple Authentication and Security Layer** (capa de seguridad y autenticación simple).

SASL es un framework para autenticación y autorización en protocolos de Internet. Separa los mecanismos de autenticación de los protocolos de la aplicación, permitiendo en teoría, a cualquier protocolo de aplicación que use SASL, usar cualquier mecanismo de autenticación soportado por SASL. A pesar de que mediante SASL sólo se maneja la autenticación (y se requieren otros mecanismos --como por ejemplo TLS-- para cifrar el contenido que se transfiere), SASL proporciona medios para un uso negociado del mecanismo elegido.

La configuración **SASL** + **TLS** (Simple Authentication Security Layer with Transport Layer Security), se utiliza principalmente para autenticar a los usuarios, antes de que envíen correo a un servidor externo, debido a la restricción impuesta por el relay. Es importantísimo proteger el servidor de estos usos malintencionados.

ATENCION: A partir de la versión 2.1.7 de Dovecot, ya no es necesario configurar este apartado, ya que por defecto activa STARTLS para la conexión cifrada entre el cliente y el servidor y además activa la autenticación tanto para envío como recepción de correo.

```
# Si queremos conocer qué versiones de SASL soporta nuestro servidor Postfix, teclearemos:
postconf -a

# Y nos responderá algo como:
cyrus
dovecot
```

Configuración de Dovecot SASL.

ATENCION: Se recomienda seguir el siguiente enlace (que funciona desde la version 2.1.7):
http://wiki2.dovecot.org/HowTo/PostfixAndDovecotSASL

Vamos a configurar la autenticación del servidor Postfix SMTP con Dovecot SASL.

La comunicación entre el servidor SMTP Postfix y Dovecot SASL, se realiza a través de sockets Unix o también a través de sockets TCP. La versión 1 de Dovecot, sólo soporta comunicación a través de sockets Unix.

La ruta del socket y la lista de mecanismos ofrecidos a Postifx, tiene que ser especificada en Dovecot en el fichero de configuración **/etc/dovecot/dovecot.conf**.

```
# En el siguiente ejemplo se asume que la cola de postfix está en el directorio: /var/spool/postfix.
nano /etc/dovecot/dovecot.conf:

auth default {
     mechanisms = plain login

     passdb pam {
     }

     userdb passwd {
     }

     socket listen {
          client {
                    path = /var/spool/postfix/private/auth
                    mode = 0660
                    user = postfix
                    group = postfix
          }
     }
}

# En la línea mechanisms se activan los métodos plain y login para acceder al servidor SMTP Postfix.
# En la línea path se indica la trayectoria del socket SASL de Dovecot.
# En la línea user y group, se limitan los permisos de lectura y escritura para el usuario y grupo postfix solamente.

# Una vez hechos los cambios reiniciamos dovecot:
service dovecot restart
```

Habilitar autenticación SASL en Postfix SMTP.

Configuraciones previas de comunicación entre Postfix y Dovecot SASL.

Por defecto el servidor SMTP Postfix utiliza la implementación Cyrus SASL. Si queremos que utilice la implementación Dovecot SASL, tendremos que especificarlo con el siguiente parámetro:

```
nano /etc/postfix/main.cf

smtpd_sasl_type = dovecot
```

Adicionalmente podremos indicar cómo el servidor Postfix puede encontrar al servidor de autenticación Dovecot. Ésto dependerá de las configuraciones que hayamos puesto en la configuracion de dovecot.conf:

- Si hemos configurado la comunicación via **sockets Unix**:

```
nano /etc/postfix/main.cf

smtpd_sasl_path = private/auth

# Este ejemplo utiliza un directorio relativo a la cola Postfix, por lo que funcionará independientemente
# del modo de ejecución de Postfix (chrooted o no).
```

- Si hemos configurado la comunicación via **sockets TCP**:

```
# Si Dovecot se está ejecutando en una máquina diferente, tendremos que reemplazar 127.0.0.1
# por la dirección IP correspondiente.
nano /etc/postfix/main.cf:

smtpd_sasl_path = inet:127.0.0.1:12345

#Nota: Si especificas una dirección IP remota, la información será enviada en texto plano por la red.
```

Habilitando la autenticación SASL.

Independientemente del tipo de implementación SASL, el habilitar la autenticación SMTP en el servidor Postfix, siempre requiere de la opción **smtpd_sasl_auth_enable**:

```
nano /etc/postfix/main.cf:

smtpd_sasl_auth_enable = yes

# A continuación reiniciamos el servidor:
service postfix restart
```

Después del reinicio del servidor, los clientes SMTP verán una opción a mayores de autenticación AUTH, seguida de la lista de mecanismos de autenticación que soporta el servidor. Por ejemplo:

```
telnet localhost 25
    ...
    220 server.example.com ESMTP Postfix
    EHLO client.example.com
    250-server.example.com
    250-PIPELINING
    250-SIZE 10240000
    250-AUTH DIGEST-MD5 PLAIN CRAM-MD5
    ...
```

Sin embargo no todos los clientes reconocen la opción AUTH. Algunas implementaciones antiguas esperan que el servidor envíe un "=" como separador entre AUTH y la lista de mecanismos disponibles.

La opción broken_sasl_auth_clients, permite repetir la opción AUTH en una forma que estos clientes desfasados, puedan entender:

```
nano /etc/postfix/main.cf:

broken_sasl_auth_clients = yes

# Nota: Habilitaremos esta opción para clientes Outlook incluídos hasta la versión 2003
# y para clientes Outlook Express incluídos hasta la versión 6.
# Esta opción no provoca ningún tipo de incompatibilidad con otros clientes.

# Reiniciamos de nuevo Postfix:
service postfix restart
```

Después de la recarga, el servidor SMTP Postfix propagará la opción de AUTH dos veces - una para los clientes modernos y otra para los clientes que no siguen el estándar.

```
telnet localhost 25
    ...
```

```
220 server.example.com ESMTP Postfix
EHLO client.example.com
250-server.example.com
250-PIPELINING
250-SIZE 10240000
250-AUTH DIGEST-MD5 PLAIN CRAM-MD5
250-AUTH=DIGEST-MD5 PLAIN CRAM-MD5
...
```

Políticas de Postfix referentes a los mecanismos SASL.

El servidor Postifx soporta políticas que limitan los mecanismos SASL disponibles para los clientes, basados en ciertas propiedades. En esta sección se explican cómo se usan esas políticas:

Propiedades de mecanismos SASL

Propiedad	Descripción
noanonymous	No utilizar mecanismos que permitan autenticación anónima.
noplaintext	No utilizar mecanismos que permitan usuario y contraseña en texto plano (sin encriptar).
nodictionary	No utilizar mecanismos que sean vulnerables a ataques de diccionario.
forward_secrecy	Requiere forward secrecy entre sesiones (si se interrumpe una sesión no interrumpirá las sesiones anteriores).
mutual_auth	Utilizar sólo mecanismos que autentiquen el cliente contra el servidor y viceversa.

- **Sesión SMTP Unencrypted.**

La política por defecto es permitir cualquier mecanismo en el servidor SMTP, excepto la autenticación anónima:

```
nano /etc/postfix/main.cf

# Especificamos la lista de propiedades de autenticación separadas por comas o espacios:
smtpd_sasl_security_options = noanonymous

# Importante: Tenemos que indicar siempre la opción noanonymous, ya que en otros casos
# el servidor SMTP puede dar como desconocidos a clientes autenticados correctamente.
```

- **Sesión SMTP Encrypted (TLS).**

Un parámetro adicional controla la política del mecanismo Postfix SASL durante la encriptación TLS de una sesión SMTP. Por defecto se suele copiar las opciones de una sesión unencrypted:

```
nano /etc/postfix/main.cf

# Activamos esta opción:
smtpd_sasl_tls_security_options = $smtpd_sasl_security_options

# Una política más sofisticada permite mecanismos plaintext, pero sólo sobre una conexión TLS-encriptada:
nano /etc/postfix/main.cf

# Activamos estas opciones:
smtpd_sasl_security_options = noanonymous, noplaintext
smtpd_sasl_tls_security_options = noanonymous

# Para ofrecer autenticación SASL sólo a través de una sesión TLS-encriptada tenemos que especificar:
nano /etc/postfix/main.cf

# Activar esta opción:
smtpd_tls_auth_only = yes
```

Habilitar autorización SASL en servidor Postfix

Una vez que el cliente se ha autenticado con SASL, el servidor Postfix decide qué podrá hacer ese cliente en el servidor SMTP.

Ejemplos de posibles autorizaciones SMTP son:

- Enviar un mensaje a destinatarios remotos.
- Permitir el uso de remitentes distintos en MAIL FROM.

Estos permisos NO están habilitados por defecto.

Enviar un mensaje a destinatarios remotos.

La restricción **permit_sasl_authenticated** permite a los clientes enviar e-mails a destinatarios remotos. Tendremos que añadirla a la lista de **smtpd_recipient_restrictions** de la siguiente forma:

```
nano /etc/postfix/main.cf

# Añadiremos a la opción smtpd_recipient_restrictions lo siguiente:
smtpd_recipient_restrictions =
        permit_mynetworks
        permit_sasl_authenticated
        reject_unauth_destination

# La opción de permit_sasl_authenticated, tiene que ir antes de la opción reject rule.

'''Nota:'''
# Si no habilitamos "Mi servidor requiere autenticación" en outlook u otro cliente,
# no podemos enviar correos a destinatarios distintos a nuestro dominio, y obtendremos
# el mensaje de error "relay access denied".
# No usar el login root para acceder al servidor de correo.
# No te olvides de crear un nuevo usuario, antes de probar la autenticación usando outlook.
```

Permitir el uso de remitentes distintos en MAIL FROM.

Por defecto un cliente SMTP puede especificar cualquier remitente en la sección de MAIL FROM. Ésto es debido a que el servidor SMTP Postfix, sólo conoce del cliente el hostname y la dirección IP, pero no quién es el usuario que controla el cliente SMTP.

Ésto cambia radicalmente en el momento que el cliente SMTP usa la autenticación SASL. Ahora, el servidor Postfix conoce quién es el remitente. Si le damos a Postfix una tabla de los remitentes y sus logins, el servidor Postfix puede decidir si el cliente puede usar un e-mail u otro en la sección de MAIL FROM, de esta forma estamos restringiendo el uso de e-mails incorrectos en el remitente.

```
# Tendremos que añadir lo siguiente en la configuración de postfix:
nano /etc/postfix/main.cf

# Configuración que indica la lista de remitentes permitidos asociados a su login:
smtpd_sender_login_maps = hash:/etc/postfix/controlled_envelope_senders

# Tendremos que editar también las restricciones de los destinatarios, añadiendo la opción
# reject_sender_login_mismatch

smtpd_recipient_restrictions =
       permit_mynetworks
       reject_sender_login_mismatch
       permit_sasl_authenticated
       reject_unauth_destination

# Y por supuesto la lista de remitentes con sus logins y los e-mails correctos,
# la pondremos en un fichero que crearemos en /etc/postfix/controlled_envelope_senders:
nano /etc/postfix/controlled_envelope_senders

# sender                owners (SASL login names)
john@example.com        john
mary@example.com        mary
support@example.com     john, mary

# De esta forma el usuario john puede enviar e-mails con el remitente john@example.com y
# support@example.com, mary puede enviar con el e-mail mary@example.com y support@example.com.
```

```
# Para aplicar estos cambios tenemos que ejecutar los siguientes comandos:
postmap /etc/postfix/controlled_envelope_senders

service postfix restart

#o tambien
service postfix reload
```

Puedes consultar también las siguientes restricciones adicionales:

- reject_authenticated_sender_login_mismatch
- reject_unauthenticated_sender_login_mismatch

Reenvío de correos.

¿Alguna vez te has preguntado como reenviar tus correos a una cuenta diferente? Ésto, es especialmente útil, si por ejemplo eres administrador y gestionas muchos sitios. Puede ser útil reenviar todos los e-mails que te lleguen a diferentes cuentas a tu cuenta primaria.

Es bastante fácil hacerlo: simplemente tienes que crear un fichero llamado .forward en tu carpeta home. Inserta dentro la lista de correos, separados por comas, dónde quieras que te lleguen los correos.

```
# Edita el fichero de reenvíos.
nano .forward

# Añade todos los e-mails de destino, separados por comas.
pruebas@pruebas.local, otromail@gmail.com
```

ATENCIÓN: No confundir el reenvío de correos con los alias, que vimos anteriormente.

Registros SPF para evitar Spoofing (suplantación de identidad).

SPF (Sender Policy Framework), extiende el protocolo SMTP para permitir comprobar las máquinas que están autorizadas a enviar correo para un dominio determinado. La idea es identificar las máquinas autorizadas por su dirección IP, y que esta comprobación la haga el responsable del dominio que recibirá el correo.

Una aproximación a la solución podría suponer que el remitente del correo, hace los envíos desde la misma máquina que los recibe. Como se puede resolver la dirección IP a donde se enviarían correos al remitente a través del registro MX del servicio DNS (RMX, del inglés Reverse MX), si esta dirección coincide con la que genera el envío, se puede entender que es el remitente real. Pero esta suposición no siempre es cierta, especialmente en grandes proveedores de soluciones de correo como Yahoo!, Hotmail, o GMail.

Otra propuesta, la **DMP** (Protocolo de Servidores de Correo Identificados, del inglés Designated Mailer Protocol), consiste en que los proveedores de servicios de internet identifiquen las máquinas responsables del envío del correo. Esta solución es válida, pero para que sea efectiva requiere que todos los proveedores la adopten e implementen.

Como mezcla de estas dos propuestas, surge la idea de usar registros DNS para identificar las máquinas autorizadas para envío de correo (sean del proveedor de servicios de internet que sean). Ésto es lo que se propone en la solución SPF.

Cómo se configuran los registros SPF.

El registro SPF define uno o más test a llevar a cabo para verificar el servidor que está enviando el correo. Cuando uno de los test se evalúa con éxito, entonces se da por válido ese MTA. En caso de que el primer test de resultado negativo, continuaría con los siguientes, hasta llegar al final de la línea indicada en el registro SPF.

Para configurar los registros SPF, el propietario del nombre de dominio, debe añadir en la configuración de la zona DNS del dominio, las direcciones IP de las máquinas utilizadas para enviar correo. Esto se consigue utilizando **registros tipo TXT** y **registros tipo SPF**. **Ambos llevan la misma información**.

Se recomienda crear los dos tipos de registros y usar las IP y los nombres de hosts con el objetivo de acelerar el proceso de contacto entre MTA (en lugar de usar PTR).

Un ejemplo de registro SPF almacenado en registro TXT y SPF:

```
pruebas.local.  IN TXT      "v=spf1 a:10.23.5.10 a:10.25.6.10 mx:pruebas.local ip4:10.23.5.0/24 -all"
pruebas.local.  IN TXT      "v=spf1 a:10.23.5.10 a:10.25.6.10 mx:pruebas.local ip4:10.23.5.0/24 -all"

# Otro ejemplo:
pruebas.local.  IN TXT      "v=spf1 a mx ip4:10.23.5.0/24 -all"
```

Explicación de los registros SPF:

- **v=** define la versión usada de SPF (versión 1).
- **mx** Indica las máquinas autorizadas a enviar correo desde ese dominio. Para ello lo que hará es comprobar que la máquina desde la que se envía coincida con las indicadas en el registro MX del dominio pruebas.local.
- **a** Indica las máquinas autorizadas a enviar correo desde ese dominio. Para ello lo que hace es comprobar que la máquina desde la que se envía coincida con la dirección IP indicada en a (address).
- **ip4** Indica las máquinas autorizadas a enviar correo desde ese dominio. Para ello lo que hace es comprobar que la máquina desde la que se envía esté en la red indicada en ip4.
- **-all** Cualquier otro tipo de origen será denegado. Si quisiéramos permitir cualquier otro origen lo indicaríamos con +all, y si lo quisiéramos marcar como sospechoso lo pondríamos como ~all.
- **ptr** Utiliza la IP de la máquina indicada en PTR y una resolución inversa. Si la IP del equipo indicado en PTR coincide con la IP del MTA remitente y además el dominio del remitente es el mismo dominio que el del host obtenido del PTR, entonces se permite la transferencia.

Para el último ejemplo:

- Si la dirección del remitente puede ser resuelta por cualquier registro A del dominio, o el servidor de correo está especificado en un registro MX, entonces no es necesario especificar la lista de hosts. Y quedaría reducida a:

"v=spf1 a mx ip4:10.23.5.0/24 -all"

Si queremos consultar el registro SPF para el dominio terra.es, podremos hacer:

```
dig +short terra.es txt

# Y obtendremos algo como:
"v=spf1 mx ptr:mailhost.terra.es mx:mx.dominios.terra.es +a:spf.terra.es
       ip4:213.4.129.0/24 ip4:213.4.138.0/24 ip4:213.4.149.0/24 ~all"

# Si comprobamos la información de los MX de Terra, obtendremos:
.....
;; ADDITIONAL SECTION:
mx.terra.es.            26444   IN      A       213.4.149.224

# Con lo que podemos comprobar que ese servidor está en una de
# las subredes citadas en el registro SPF.
```

Direcciones interesantes para generar registros SPF:

- http://www.royhochstenbach.com/projects/spfgenerator/
- http://www.mtgsy.net/dns/spfwizard.php

Más información detallada sobre la creación de registros SPF:

- http://www.zytrax.com/books/dns/ch9/spf.html

Antispam con SPAMASSASSIN.

Hoy en día las redes están sobrecargadas con tráfico SPAM, sin embargo hay una forma de filtrar ese tráfico con software como **spamassasin**.

Por ahora tenemos instalado un servidor SMTP ejecutando Postfix.

```
# Para instalar spamassasin y su cliente spamc, tendremos que realizar lo siguiente:
aptitude install spamassassin spamc
```

El paquete spamassassin incluye un daemon que puede ser llamado por aplicaciones como procmail, y que se puede integrar fácilmente en un MTA como Postfix.

Configuración de Spamassassin

Por defecto spamassassin se ejecuta como root. Vamos a modificarlo para que se ejecute con un usuario diferente:

```
groupadd -g 5001 spamd
useradd -u 5001 -g spamd -s /sbin/nologin -d /var/lib/spamassassin spamd
mkdir /var/lib/spamassassin
chown spamd:spamd /var/lib/spamassassin

# Ahora necesitamos hacer algunos cambios en /etc/default/spamassassin para asegurarnos
# que tiene los siguientes valores:
nano /etc/default/spamassassin

# Revisar las siguientes opciones y añadir las que falten:
# Modificaremos ENABLED=1 para habilitar spamassassin:
ENABLED=1

# Indicamos que vamos a ejecutar spamd daemon con el usuario spamd y que
# su home directory es (/var/lib/spamassassin/) y sus logs estan en /var/lib/spamassassin/spamd.log
SAHOME="/var/lib/spamassassin/"
OPTIONS="--create-prefs --max-children 5 --username spamd --helper-home-dir ${SAHOME} -s ${SAHOME}spamd.log"
PIDFILE="${SAHOME}spamd.pid"

#Arrancamos el servicio:
service spamassassin start

# Ahora necesitamos indicarle a spamassassin algunas reglas de filtrado.
# Los parámetros por defecto van bastante bien, pero se pueden adaptar un poco:
nano /etc/spamassassin/local.cf

# Podemos dejar valores semejantes a los siguientes:
rewrite_header Subject ***** SPAM  *****
report_safe 0
required_score 2.0

#   Use Bayesian classifier (default: 1)
use_bayes 1
bayes_auto_learn 1

# Por último reiniciamos de nuevo spamassassin:
service spamassassin restart
```

Ya casi lo tenemos todo listo, sólo nos falta configurar postfix para que pase todos los correos que recibe, previamente a spamassassin, antes de entregarlos a los buzones.

Integración Postfix - Spamassassin

Ahora, tenemos que decirle a Postfix que use spamassassin. En nuestro caso, spamassassin será llamado solamente una vez que postfix ha terminado con el correo.

```
# Para decirle a Postfix que use spamassassin, editaremos el archivo /etc/postfix/master.cf y
# modificamos la siguiente línea:
nano /etc/postfix/master.cf

smtp      inet  n       -       -       -       -       smtpd

# por la siguiente línea:
smtp      inet  n       -       n       -       -       smtpd -o content_filter=spamassassin

# Chequeamos dónde están instalados spamc y sendmail:
which spamc
/usr/bin/spamc

which sendmail
/usr/sbin/sendmail

# Y al final del fichero master.cf añadimos lo siguiente:
nano /etc/postfix/master.cf

spamassassin unix -     n       n       -       -       pipe user=spamd argv=/usr/bin/spamc -f -e
```

```
                                                    /usr/sbin/sendmail -oi -f ${sender} ${recipient}

# A partir de ahora podremos enviar correo a través del servidor y probablemente
# veremos en /var/log/mail.log algunas líneas como:
.....
relay=spamassassin, delay=1, delays=0.02/0/0/1, dsn=2.0.0, status=sent(delivered via spamassassin service)
...
```

NOTA: Se recomienda ver otras formas de integración con spamassassin a parte de ésta, que está cada vez más en desuso.

Comprobación con GTUBE

Para hacer una prueba de si realmente el sistema de detección de spam funciona correctamente, podemos utilizar una cadena de texto específica en el BODY. Es lo que se conoce como **GTUBE (Generic Test for Unsolicited Bulk Email)**.

La cadena en cuestión es la siguiente:

```
# IMPORTANTE: Esta cadena debe ir en una línea independiente:
XJS*C4JDBQADN1.NSBN3*2IDNEN*GTUBE-STANDARD-ANTI-UBE-TEST-EMAIL*C.34X
```

Más información sobre GTUBE y un e-mail de prueba en:

- http://spamassassin.apache.org/gtube/
- Ejemplo de correo con la cadena GTUBE

Antivirus CLAMAV para el correo.

A parte de protegernos del SPAM, podemos configurar nuestro servidor PostFix para que escanee nuestros correos, en busca de archivos que contengan virus.

```
aptitude update

aptitude install clamav clamav-freshclam clamsmtp

# Editamos el archivo /etc/clamsmtpd.conf:
nano /etc/clamsmtpd.conf

# Modificamos OutAddress: 10025 a OutAddress: 10026
OutAddress: 10026

# Modificamos 127.0.0.1:10026 a Listen: 127.0.0.1:10025
Listen: 127.0.0.1:10025

# Editamos los ficheros Postfix y añadimos lo siguiente:
nano /etc/postfix/main.cf

# Añadimos lo siguiente:
content_filter = scan:127.0.0.1:10025
receive_override_options = no_address_mappings

# Añadimos lo siguiente al fichero /etc/postfix/master.cf:
nano /etc/postfix/master.cf

# AV scan filter (used by content_filter)
scan      unix  -       -       n       -       16      smtp
        -o smtp_send_xforward_command=yes

# Para inyectar el correo una vez escaneado de nuevo a Postfix:
127.0.0.1:10026 inet  n -       n       -       16      smtpd
        -o content_filter=
        -o receive_override_options=no_unknown_recipient_checks,no_header_body_checks
        -o smtpd_helo_restrictions=
        -o smtpd_client_restrictions=
        -o smtpd_sender_restrictions=
        -o smtpd_recipient_restrictions=permit_mynetworks,reject
        -o mynetworks_style=host
        -o smtpd_authorized_xforward_hosts=127.0.0.0/8
```

```
# Para actualizar el antivirus se hace  con la instrucción:
freshclam

# Si queremos que se actualice el antivirus una vez al día:
crontab -e

# Añadimos la siguiente línea, para que a la 1:00 AM se actualice cada día:
00 1 * * * /usr/bin/freshclam

# Reiniciamos la máquina
reboot
```

Comprobación de detección de virus

Una vez reiniciada la máquina vamos a enviar un correo con un patrón vírico.

Te puedes descargar ficheros de pruebas, que contienen un patrón vírico desde: http://www.rexswain.com/eicar.html

```
# Podemos comprobar si el virus ha sido detectado mirando el log de mails.
more /var/log/mail.log

# Veremos algún mensaje similar a Virus Detected; Discarded Email:
# Si se detecta el virus, el e-mail será eliminado automáticamente y no será entregado.

Jan 30 01:10:59 servidor postfix/smtp[1511]: 8B0682E637: to=<manuela@pruebas.local>, relay
=127.0.0.1[127.0.0.1]:10025, delay=0.16, delays=0.02/0.04/0.06/0.04, dsn=2.0.0, status=sen
t (250 Virus Detected; Discarded Email)
Jan 30 01:10:59 servidor clamsmtpd: 100000: from=manuela@pruebas.local, to=manuela@pruebas
.local, status=VIRUS:Eicar-Test-Signature
Jan 30 01:10:59 servidor postfix/qmgr[1463]: 8B0682E637: removed
```

--Veiga (discusión) 19:16 30 ene 2013 (CET)