1 O servidor de correo Postix

Postfix é un servidor de correo de software libre / código aberto, un programa informático para o enrutamiento e envío de correo electrónico, creado coa intención de que sexa unha alternativa máis rápida, doada de administrar e segura ao amplamente utilizado *sendmail*.

Actualmente, Postfix é o axente de transporte por omisión en diversas distribucións de Linux e nas las últimas versións do Mac OS X.

1.1 Sumario

1.2 Instalación

O primeiro paso é instalar o paquete **postfix** e a libraría **ssl-cert**

```
apt install postfix ssl-cert
```

A continuación preguntaranos varias cuestións, e escollemos "Sen configuración" xa que o faremos despois.

Sempre que queiramos consultar os logs, visualizaremos o final do arquivo */var/log/mail.log*

1.3 Configuración

O ficheiro principal de configuración é **/etc/postfix/main.cf**. Hai que lembrarse de reiniciar o servidor postfix cada vez que se fan cambios.

```
service postfix restart
```

Sempre que queiramos consultar os logs, visualizaremos o final do arquivo */var/log/mail.log*

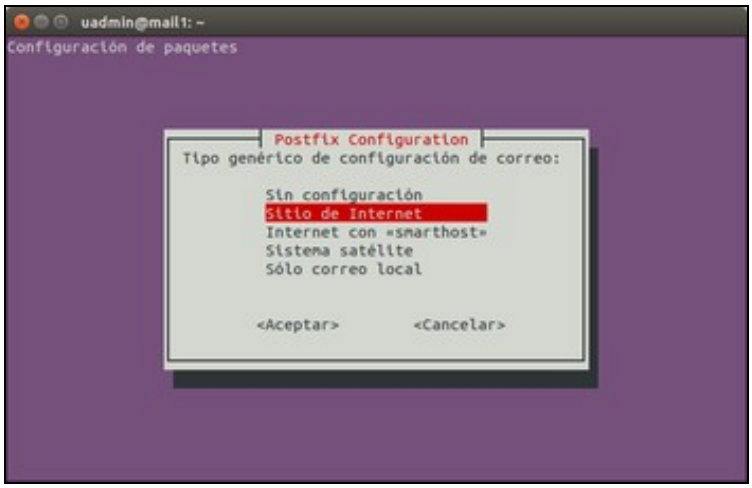
Tomaremos como nome do equipo *correo*, coma nome de dominio DNS *sitio.lan* e o enderezo IP será 192.168.0.110/24

Antes de nada editamos o ficheiro **/etc/mailname** e escribimos o nome de dominio do correo, como por exemplo **sitio.lan**

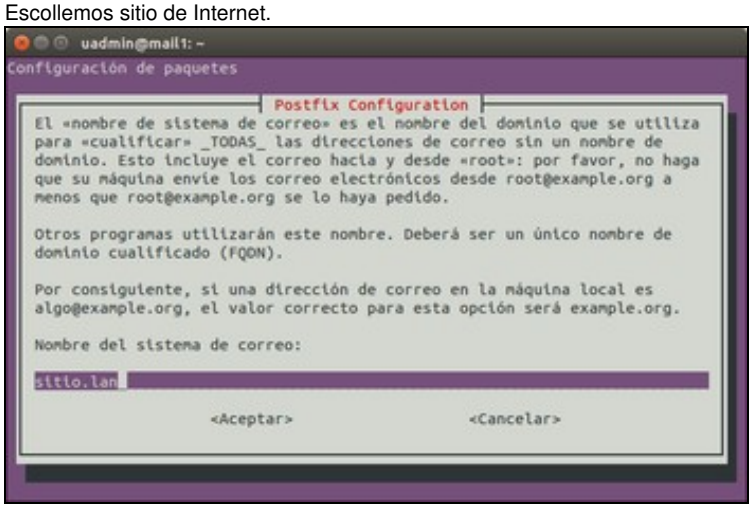
Unha vez editado ese ficheiro e establecido o nome do dominio do noso correo, reconfiguramos o *postfix*. Podemos facelo editando directamente o ficheiro */etc/postfix/main.cf* ou executando directamente (o resultado é o mesmo) o comando:

```
sudo dpkg-reconfigure postfix
```

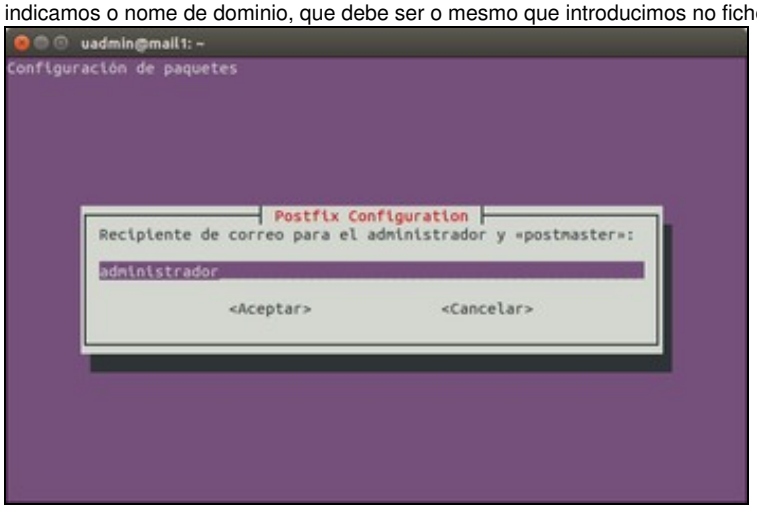
```
e respondendo as preguntas do seguinte xeito:
```

- 

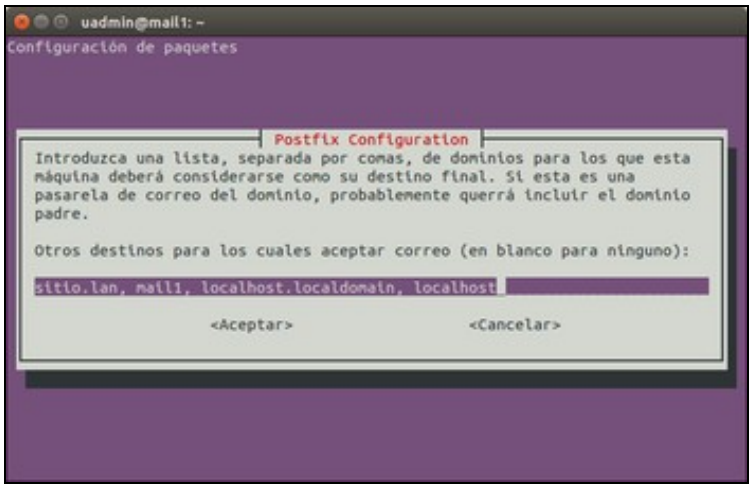
Escollemos sitio de Internet.

- 

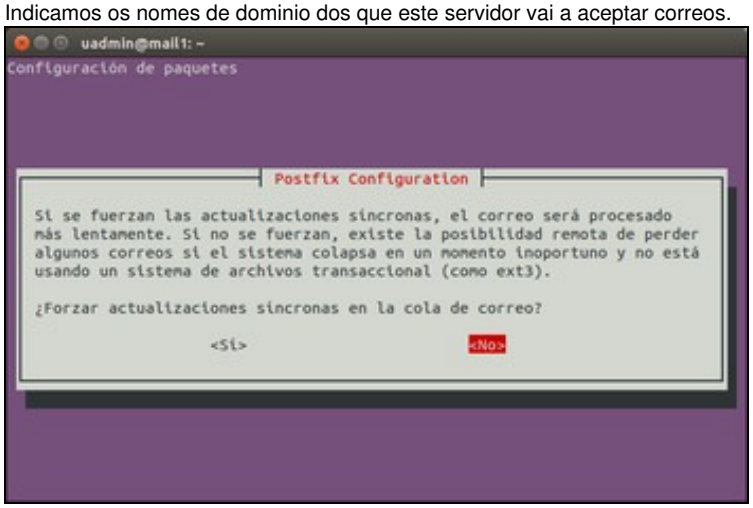
indicamos o nome de dominio, que debe ser o mesmo que introducimos no ficheiro */etc/mailname*.

- 

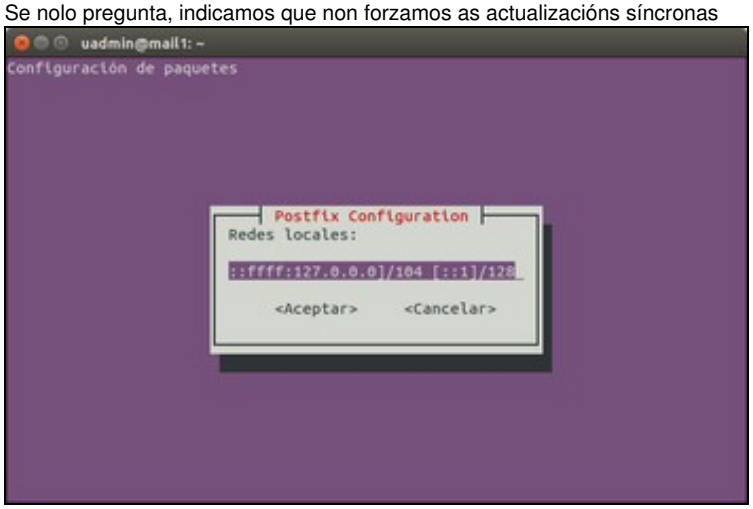
Indicamos o nome do usuario real que vai recibir o correo destinado aos usuarios *root* e *postmaster*

- 

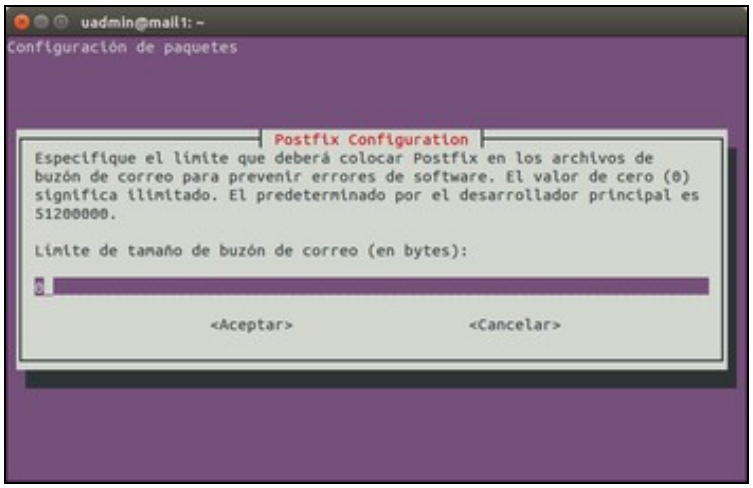
Indicamos os nomes de dominio dos que este servidor vai a aceptar correos.

- 

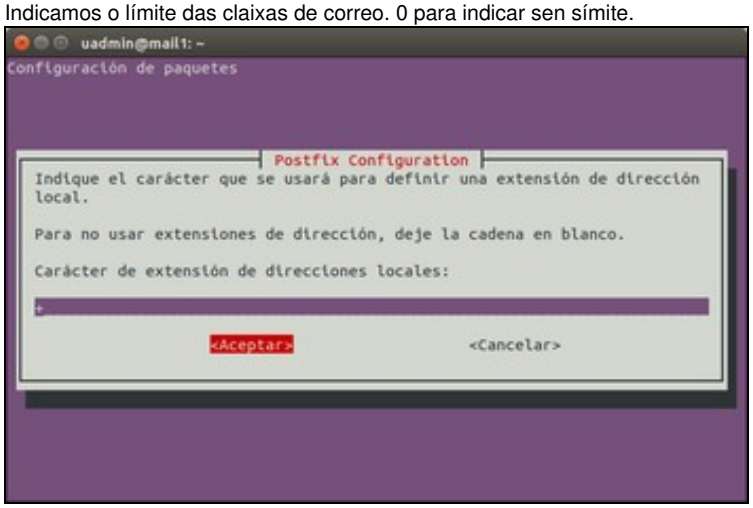
Se nolo pregunta, indicamos que non forzamos as actualizacións síncronas

- 

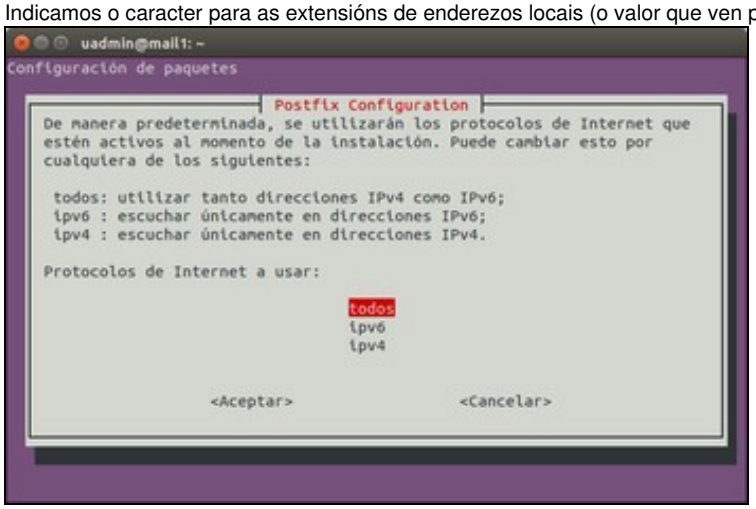
Indicamos as redes nas que se confía para enviar correos. Por defecto, so o bucle local.

- 

Indicamos o límite das claixas de correo. 0 para indicar sen símite.

- 

Indicamos o caracter para as extensións de enderezos locais (o valor que ven por defecto).

- 

Indicamos todos os protocolos (ipv4 e ipv6).

A outra alternativa é configurar os parámetros do ficheiro */etc/postfix/main.cf* editandoo directamente

Unha vez rematado, comprobamos que o contido do ficheiro de configuración é similar a este (óllo, hai algún parámetro que debemos introducir a man, xa que o asistente non o pregunta):

```
# See /usr/share/postfix/main.cf.dist for a commented, more complete version


# Debian specific:  Specifying a file name will cause the first
# line of that file to be used as the name.  The Debian default
# is /etc/mailname.
#myorigin = /etc/mailname
```

```
smtpd_banner = $myhostname ESMTP $mail_name (Ubuntu)
biff = no

# appending .domain is the MUA's job.
append_dot_mydomain = no

# Uncomment the next line to generate "delayed mail" warnings
#delay_warning_time = 4h

readme_directory = no

# TLS parameters
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_use_tls=yes
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache

# See /usr/share/doc/postfix/TLS_README.gz in the postfix-doc package for
# information on enabling SSL in the smtp client.

smtpd_relay_restrictions = permit_mynetworks permit_sasl_authenticated defer_unauth_destination
myhostname = correo.sitio.lan
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = sitio.lan, mail1, localhost.localdomain, localhost
relayhost =
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = all
```

Tamén podemos introducir as liñas do ficheiro */etc/postfix/main.cf* empregando o comando **postconf** coa avantaxe de que non imos introducir nunca directivas repetidas.

```
postconf -e 'myhostname = smtp.sitio.lan'
```

Por último reiniciamos o servidor

```
service postfix restart
```

1.3.1 Comprobacións

Empregando o comando mail (necesitamos ter instalado o paquete **mailutils**), podemos enviar un correo aos novos usuarios do equipo smtp.sitio.lan. Rematamos cunha liña cun único punto (".") e saímos pulsando Ctrl+D (^D)

```
uadmin@smtp:~$ mail pepe@sitio.lan
Cc:
Subject: Ola
Que tal vai todo
.
```

A continuación chequeamos que o outro usuario recibeu o correo. Accedemos con esa nova conta, e tecleamos o comando mail. Aparecerán todas as mensaxes cun identificador ao seu carón.

```
pepe@smtp:/home/uadmin$ mail
"/var/mail/pepe": 2 mensajes 1 nuevo 1 sin leer
 U   1 uadmin             lun feb  2 19:38  17/459   Ola
>N   2 uadmin             lun feb  2 19:44  14/419   Outro
?
```

Teclamos o número identifacador a apareceranos esa nova mensaxe.

```
? 1
Return-Path: <uadmin@smtp>
X-Original-To: pepe@sitio.lan
Delivered-To: pepe@sitio.lan
```

```
Received: by smtp.sitio.lan (Postfix, from userid 1000)
        id E4384A169B; Mon,  2 Feb 2015 19:38:34 +0100 (CET)
To: <pepe@sitio.lan>
Subject: Ola
X-Mailer: mail (GNU Mailutils 2.99.98)
Message-Id: <20150202183834.E4384A169B@smtp.sitio.lan>
Date: Mon,  2 Feb 2015 19:38:34 +0100 (CET)
From: uadmin@smtp (uadmin)
X-IMAPbase: 1422902334 2
Status: O
X-UID: 1

Que tal vai todo
.
?
```

Para borrar unha mensaxe indicamos *d<identificador>*. Por exemplo, para borrar a primeira *d1*. Para borralas todas *d**

1.4 Logs

Para configurar un ficheiro de log determinado indicamos:

```
postconf -e 'maillog_file=/var/log/postfix.log'
```

1.5 Almacenamento do correo

Habitualmente, os correos electrónicos que reciben os usuarios almácenanse de xeito temporal en */var/mauil/<username>* (por exemplo /var/mail/maria), e unha vez que son lidos, todo o contido gárdase nun ficheiro chamado ***mbox*** no directorio HOME de cada usuario.

Mbox é un termo xenérico para unha familia de formatos de documentos utilizados para almacenar conxuntos de correos electrónicos. Todas as mensaxes dunha caixa de correo (caixa de entrada) son concatenados nun único documento. O inicio de cada mensaxe é marcado por unha liña que comeza cos cinco caracteres "From:", e unha liña en branco para marcar o final. Por un tempo, o formato mbox era popular porque podía ser usada de forma moi sinxela por ferramentas de procesado de documentos para modificar ditos documentos.

Debido a que se almacena máis dunha mensaxe nun único documento, necesítase algún tipo de bloqueo para evitar que este poida corromperse cando dous ou máis procesos accedan simultáneamente. Isto podería acontecer se un programa de reparto de correo está a escribir unha mensaxe nova no documento, mentres que un cliente de correo electrónico está a borrar outra mensaxe ao mesmo tempo.

O formato ***maildir*** é unha alternativa posterior que soluciona os problemas de bloqueo de mbox. O correo pasa a almacenarse nun cartafol dentro do directorio HOME de cada usuario, e non presenta o problema dos bloqueos.

Para habilitar o uso de *maildir* (por exemplo, no directorio ~/**Maildir** de cada usuario) configuramos o Postfix da seguinte maneira:

```
postconf -e 'home_mailbox = Maildir/'
```

e tamén temos que garantir que non se usa *procmail* para o reparto de correo local.

```
postconf -e 'mailbox_command ='
```

Unha vez configurado o uso de *maildir* no Postfix, tamén habería que configuralo no axente MDA que se esté usando.

1.6 Creación de alias de usuarios

As veces pode ser interesante redirixir o correo que recibimos a un usuario en particular. Por exemplo todos os correos electrónicos que vaian a postmaster, root, etc.. poderiámolos redirixir ao usuario pepe.

Para iso é necesario crear un alcume (alias). Iso faise modificando un arquivo cuxo nome é /**etc**/**aliases**. Este arquivo contén alcumes, é dicir, equivalencias entre un enderezo local (probablemente ficticio) e un enderezo real. Así, se o servidor recibe unha mensaxe dirixida a "postmaster@sitio.lan", e en /etc/aliases hai unha liña como esta:

```
postmaster: root
```

(como, de feito, hai), será root quen realmente reciba a mensaxe. O arquivo aliases xa contén algunhas liñas comúns. A única liña que pode interesar engadir é a que redirixe o correo de root a un usuario normal (que será a que habitualmente utiliza o administrador cando non precisa privilexios de supervisión). A liña sería, simplemente:

```
root: uadmin
```

No arquivo de configuración de Postfix, hai unhas liñas que fan referencia aos alias

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

Cada vez que se modifica o arquivo de alias debe executarse o comando ***newaliases***, que xenera o arquivo */etc/aliases.db* para mellorar o acceso durante a execución de Portfix.

1.7 Configuración de redes desde as que se pode usar o servidor de correo Postfix

Postfix por defecto, deixará enviar correo electrónico dende clientes externos a todos os usuarios que estean conectados ás mesmas redes ás que está conectado o servidor Postfix.

Con esta configuración, á hora de enviar correo, non se validará o usuario e o contrasinal do usuario que está a enviar correo, co que calquera podería enviar correos e facerse pasar por outras persoa, e ademais ésto é un burato de seguridade que podería converter o noso servidor nun Relé Aberto.

O que non permite Postfix, é enviar correos a dominios externos como gmail, etc.. (daríanos unha mensaxe de *Relay Denied* á hora de enviar correo), a non ser que a nosa rede estea especificada en **mynetworks** como unha rede confiable.

En cambio si que nos deixará enviar correos dende o propio servidor, xa que aparece como rede confiable 127.0.0.0/8 no parámetro *mynetworks*.

No ficheiro de configuración de Postfix, podemos engadir as redes confiables desde as que permitimos enviar correos electrónicos

```
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128
```

1.8 Envío de correo a dominios externos

Para poder enviar correos a dominios externos ao do servidor SMTP, temos que habilitar a autenticación SASL (Simple Authentication and Security Layer) no servidor Postfix.

SASL é un framework para autenticación e autorización en protocolos de Internet. Separa os mecanismos de autenticación dos protocolos da aplicación, permitindo en teoría, a calquera protocolo de aplicación que use SASL, usar calquera mecanismo de autenticación soportado por SASL. A pesar de que mediante SASL só se manexa a autenticación (e requírense outros mecanismos --como por exemplo TLS-- para cifrar o contido que se transfire), SASL proporciona medios para un uso negociado do mecanismo elixido.

A configuración SASL + TLS (Simple Authentication Security Layer with Transport Layer Security), utilízase principalmente para autenticar os usuarios, antes de que envíen correo a un servidor externo, debido á restrición imposta polo relay. É importante protexer o servidor destes usos malintencionados.

Os pasos son os seguintes:

1. Instalamos o paquete *dovecot-common*:
   ```
   apt install dovecot-common
   ```
2. Editamos o ficheiro de configuración */etc/dovecot/conf.d/10-master.conf* da seguinte maneira:
   ```
   service auth {
     # auth_socket_path points to this userdb socket by default. It's typically
     # used by dovecot-lda, doveadm, possibly imap process, etc. Its default
     # permissions make it readable only by root, but you may need to relax these
     # permissions. Users that have access to this socket are able to get a list
     # of all usernames and get results of everyone's userdb lookups.
     unix_listener auth-userdb {
       #mode = 0600
       #user =
       #group =
     }

     # Postfix smtp-auth
   ```

```
unix_listener /var/spool/postfix/private/auth {
  mode = 0660
  user = postfix
  group = postfix
}
```

3. E tamén o ficheiro */etc/dovecot/conf.d/10-auth.conf* para permitir que os clientes *Outlook* se poidan conectar:

```
auth_mechanisms = plain login
```

4. Reiniciamos o servizo dovecot

```
service dovecot restart
```

5. Configuramos Postfix para que empregue autenticación SASL. Podemos facelo editando */etc/postfix/main.cf* ou executando os seguintes comandos:

```
postconf -e 'smtpd_sasl_type = dovecot'
postconf -e 'smtpd_sasl_path = private/auth'
postconf -e 'smtpd_sasl_local_domain ='
postconf -e 'smtpd_sasl_security_options = noanonymous'
postconf -e 'broken_sasl_auth_clients = yes'
postconf -e 'smtpd_sasl_auth_enable = yes'
postconf -e 'smtpd_tls_auth_only = yes'
```

6. Unha vez que teñamos o certificado dixital necesario e a chave privada (gardados no directorio */etc/ssl/private* e */etc/ssl/certs* volvemos a editar o ficheiro ou executar estes comandos:

```
postconf -e 'smtp_tls_security_level = may'
postconf -e 'smtpd_tls_security_level = may'
postconf -e 'smtp_tls_note_starttls_offer = yes'
postconf -e 'smtpd_tls_loglevel = 1'
postconf -e 'smtpd_tls_received_header = yes'
postconf -e 'smtpd_tls_key_file = /etc/ssl/private/server.key'
postconf -e 'smtpd_tls_cert_file = /etc/ssl/certs/server.crt'
```

7. So nos queda reiniciar o servizo postfix

```
service postfix restart
```

Podemos probar se usa a autenticación SASL. Supoñendo que o equipo chámase *mail.example.com* executamos o comando telnet:

```
telnet mail.example.com 25
```

Enviamos o comando

```
ehlo mail.example.com
```

E debería producirse unha saída similar a esta

```
250-STARTTLS
250-AUTH LOGIN PLAIN
250-AUTH=LOGIN PLAIN
250 8BITMIME
```

1.8.1 Políticas de Postfix referentes aos mecanismos SASL

O servidor Postifx soporta políticas que limitan os mecanismos SASL dispoñibles para os clientes, basados en certas propiedades. En esta sección explícanse como se usan esas políticas:

Propiedades de mecanismos SASL

Propiedade	Descripción
noanonymous	Non utilizar mecanismos que permitan autenticación anónima.
noplaintext	Non utilizar mecanismos que permitan usuario e contrasinal en texto plano (sen encriptar).
nodictionary	Non utilizar mecanismos que sexan vulnerables a ataques de diccionario.
forward_secrecy	Requiere forward secrecy entre sesións (se se interrompe unha sesión non interrompera as sesións anteriores).
mutual_auth	Utilizar so mecanismos que autentiquen o cliente contra o servidor e viceversa.

A política por defecto é permitir calquera mecanismo no servidor SMTP excepto a autenticación anónima:

```
postconf -e 'smtpd_sasl_security_options = noanonymous'
```

Para permitir unha política máis sofisticada que permite mecanismos *plaintext*, pero so sobre unha conexión TLS-encriptada:

```
postconf -e 'smtpd_sasl_security_options = noanonymous, noplaintext'
postconf -e 'smtpd_sasl_tls_security_options = noanonymous'
```

Neste caso, so podemos enviar correos con contrasinal sen cifrar empregando os portos 587 ou 465. Será necesario que estén habilitados.

Para permitir enviar correos a destinos de fora do noso dominio, necesitamos activar a seguinte opción:

```
postconf -e 'smtpd_relay_restrictions = permit_sasl_authenticated,permit_mynetworks,reject_unauth_destination'
```

e poñer a opción *permit_sasl_authenticated* diante de *reject_unauth_destination*

1.8.2 Permitir o uso de remitentes distintos en EMAIL FROM

Por defecto un cliente SMTP pode especificar calquera remitente na sección de EMAIL FROM. Ésto é debido a que o servidor SMTP Postfix, só coñece do cliente o hostname e a dirección IP, pero non quen é o usuario que controla o cliente SMTP.

Ésto cambia radicalmente no momento que o cliente SMTP usa a autenticación SASL. Agora, o servidor Postfix coñece quen é o remitente. Se lle damos a Postfix unha táboa dos remitentes e os seus logins, o servidor Postfix pode decidir se o cliente pode usar un correo electrónico ou outro na sección de EMAIL FROM, desta forma estamos a restrinxir o uso de correos electrónicos incorrectos no remitente.

Indicamos cal e a lista de remitentes aos que se lle permite indicar outro correo remitente, e que politica aplicamos na directiva **smtpd_recipient_restrictions**

```
postconf -e 'smtpd_sender_login_maps = hash:/etc/postfix/controlled_envelope_senders'
postconf -e 'smtpd_sender_restrictions = permit_sasl_authenticated,reject_sender_login_mismatch'
```

Tamén indicamos cales son os usuarios aos que se lle permite cambiar o remitente, e que remitentes se lle permiten. Editamos o ficheiro */etc/postfix/controlled_envelope_senders*:

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

Desta forma o usuario *john* pode enviar e-mails co remitente *john@example.com* e *support@example.com*, *mary* pode enviar con *mary@example.com* e *support@example.com*.

No caso de que a versión de postfix, sexa anterior a 2.11, haberá indicar absolutamente todos os usuarios do sistema, xa que de non facelo, os que non aparezan nesa lista, non poderán enviar correos electrónicos.

Para aplicar estos cambios temos que executar os seguintes comandos:

```
postmap /etc/postfix/controlled_envelope_senders
service postfix restart
```

ou se so cambiamos a lista de usuarios:

```
postmap /etc/postfix/controlled_envelope_senders
service postfix reload
```

1.8.3 Execución en contedores Docker

É necesario copiar varios ficheiros ao entorno chroot que executa o postfix

```
cp -f /etc/services /var/spool/postfix/etc/services
cp -f /etc/resolv.conf /var/spool/postfix/etc/resolv.conf
```

1.9 Uso de encriptación SSL/TLS e protocolo SMTPS

Por defecto o protocolo SMTPS está desactivado. Anque admite conexións encriptadas mediante TLS, non permite contrasinais encriptadas (anque se está habilitado TLS non é problema enviar as contrasinais en texto plano), nin atende peticións nos portos 465/587, so no 25. Se queremos habilitalo

hai que descomentar as seguintes liñas do ficheiro **/etc/postfix/master.cf** Para habilitar contrasinais encriptados:

```
submission inet n       -       -       -       -       smtpd
  -o syslog_name=postfix/submission
  -o smtpd_tls_security_level=encrypt
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_relay_restrictions=permit_sasl_authenticated,reject
  -o milter_macro_daemon_name=ORIGINATING
```

e para habilitar SSL

```
smtps     inet  n       -       -       -       -       smtpd
  -o syslog_name=postfix/smtps
  -o smtpd_tls_wrappermode=yes
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_relay_restrictions=permit_sasl_authenticated,reject
  -o milter_macro_daemon_name=ORIGINATING
```

~~Tamén é importante deshabilitar o uso dos protocolos inseguros coma SSLv3. Modificamos o ficheiro */etc/postfix/main.cf* executando o comando~~
Postfix xa por defecto deshabilita os protocolos inseguros SSLv2 e SSLv3, polo que a seguinte liña xa non é necesaria.

```
postconf -e 'smtpd_tls_mandatory_protocols =!SSLv2, !SSLv3'
```

Unha vez rematado, reiniciamos o servidor postfix

1.10 Rexistros SPF para evitar Spoofing (suplantación de identidade)

Un rexistro SPF é un tipo de rexistro DNS que identifica que servidores de correo teñen permitido enviar correos en nome do dominio. É tan sinxelo como engadir rexistros tipo A ou MX.

E por que é tan importante? Porque hoxen en día soen chegar as caixas de correo dos usuarios centros de mensaxes non desexadas ou fraudulentas. Os spammers, envian correo desde os seus servidores, pero co teu dominio como remitente.

O propósito dos rexistros SPF é previr que os spammers envíen correos poñendo enderezos do noso dominio, como remitentes. Os servidores de correo receptores, poden comprobar, e de feito comproban, que unha mensaxe de correo, que ten como remitente un enderezo de correo dun dominio determinado, foi enviado por un servidor SMTP autorizado para ese dominio.

1.10.1 Configuración

O rexistro SPF define un ou máis test a levar a cabo para verificar o servidor que está a enviar o correo. Cando un dos test se avalía con éxito, entón dáse por válido ese MTA. En caso de que o primeiro test dé resultado negativo, continuaría cos seguintes, ata chegar ao final da liña indicada no rexistro SPF.

Para configurar os rexistros SPF, o propietario do nome de dominio, debe engadir na configuración da zona DNS do dominio, os enderezos IP das máquinas utilizadas para enviar correo. Isto conséguese utilizando rexistros tipo TXT e rexistros tipo SPF. Ambos os dous levan a mesma información.

Recoméndase crear os dous tipos de rexistros e usar os enderezos IP e os nomes de hosts co obxectivo de acelerar o proceso de contacto entre MTA (en lugar de usar PTR).

Exemplos:

```
sitio.lan.      IN TXT   "v=spf1 a:10.23.5.10 a:10.25.6.10 mx:sitio.lan ip4:10.23.5.0/24 -all"
sitio.lan.      IN TXT   "v=spf1 a mx ip4:10.23.5.0/24 ~all"
```

sendo:

- **v=** define a versión usada de SPF (versión 1).
- **mx** indica as máquinas autorizadas a enviar correo dende ese dominio. Para iso o que fará é comprobar que a máquina dende a que se envía coincida coas indicadas no rexistro MX do dominio sitio.lan.
- **a** indica as máquinas autorizadas a enviar correo dende ese dominio. Para iso o que fai é comprobar que a máquina dende a que se envía coincida coa dirección IP indicada.
- **ip4** indica as máquinas autorizadas a enviar correo dende ese dominio. Para iso o que fai é comprobar que a máquina dende a que se envía estea na rede indicada en ip4.

- **-all**: Calquera outro tipo de orixe será denegada. Se quixésemos permitir calquera outra orixe o indicariamos con **+all**, e se o quixésemos marcar como sospeitoso poñeriámolo como **~all**.
- **ptr** Utiliza o enderezo IP da máquina indicada en PTR e unha resolución inversa. Se o enderezo IP do equipo indicado en PTR coincide coa IP do MTA remitente e ademais o dominio do remitente é o mesmo dominio que o do host obtido do PTR, entón permítese a transferencia.

Se o enderezo IP do remitente pode ser resolto por calquera rexistro A do dominio, ou o servidor de correo está especificado nun rexistro MX, entón non é necesario especificar a lista de hosts. E quedaría reducida a:

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

Se queremos consultar o rexistro SPF para o dominio xunta.es, poderemos facer:

```
dig +short xunta.es txt
```

que producirá a saída:

```
"v=spf1 mx ip4:85.91.64.213 ip4:85.91.64.214 -all"
```

1.11 AntiSPAM con SPAMASSASSIN

Hoxe en día as redes están sobrecargadas con tráfico SPAM, sen embargo hai unha forma de filtrar ese tráfico con software como spamassasin.

SpamAssassin usa unha gran variedade de tests locais e de rede para identicar trazas de correos SPAM. Isto dificulta que as mensaxes SPAM circulen libremente polo mundo adiante. Está distribuido baixo os mesmos termos e condicións que o servidor web Apache.

Instalamos os paquetes *spamassassin* e *spamc*

```
apt install spamassassin spamc
```

O seguinte que temos que facer é engadir un grupo chamado *spamd*, e un usuario chamado *spamd* con directorio home /var/log/spamassassin (que tamén haberá que crear e cambiarlle o usuario e grupo propietario):

```
groupadd spamd
useradd -g spamd -s /bin/false -d /var/log/spamassassin spamd
mkdir /var/log/spamassassin
chown spamd:spamd /var/log/spamassassin
```

1.11.1 Configuración

Editamos o ficheiro **/etc/default/spamassassin** e introducimos o seguinte:

```
CRON=1
SAHOME="/var/log/spamassassin/"
OPTIONS="--create-prefs --max-children 2 --username spamd -H ${SAHOME} -s ${SAHOME}spamd.log"
PIDFILE="${SAHOME}spamd.pid"
```

A continuación reiniciamos o servizo *spamassassin*

```
service spamassassin restart
```

1.11.2 Integración con Postfix

Hai que indicarlle a Postfix, que pase todos os correos polo demonio de spamassassin.

Editamos o ficheiro **/etc/postfix/master.cf** e despois liña similar esta

```
smtp      inet  n       -       -       -       -       smtpd
```

engadimos

```
-o content_filter=spamassassin
```

No caso de que empreguemos submision ou smtps, tamén haberá que engadilo debaixo desas liñas de igual modo.

Agora xa filtrará todo o correo por spamassassin.

Para facer que filtre o contido despois de saír da cola de envio, engadimos ao final do ficheiro */etc/postfix/master.cf*

```
spamassassin unix -     n       n       -       -       pipe
        user=spamd argv=/usr/bin/spamc -f -e
        /usr/sbin/sendmail -oi -f ${sender} ${recipient}
```

Para aplicar os cambios reiniciamos o servizo *postfix*

1.11.3 Configuración de regras antispam

Podemos establecer algunha regra antispam no ficheiro de configuración **/etc/spamassassin/local.cf**, ben quitando comentarios as xa existentes ou establecendo unha nova.

Por exemplo, para engadir unha cabeceira aos correos spam, quitamos o comentario ou engadimos:

```
rewrite_header Subject [***** SPAM _SCORE_ *****]
```

Spamassassin dalle puntuación a cada correo despois de realizar certos tests sobre el. A seguinte liña, fará que se marque clasifique un email como spam, se a puntuación é maior.

```
required_score           3.0
```

Para usar o teorema de Bayes para comprobar emails e facer que este teña configurada a autoaprendizaxe introducimos as seguintes liñas:

```
use_bayes   1
bayes_auto_learn        1
```

Por último, reiniciamos o servizo spamassassin.

```
service spamassassin restart
```

1.11.4 Comprobación con GTUBE

Para facer unha proba de se realmente o sistema de detección de spam funciona correctamente, podemos utilizar unha cadea de texto específica no corpo da mensaxe. É o que se coñece como **GTUBE (Generic Test for Unsolicited Bulk Email)**.

A cadea en cuestión é a seguinte (debe ir nunha liña independente):

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

Veremos no log (*/var/log/mail.log*) algunha mensaxe como:

```
...
relay=spamassassin, delay=1, delays=0.02/0/0/1, dsn=2.0.0, status=sent(delivered via spamassassin service)
...
```

A mensaxe chegará ao remitente, pero coa cabeceira *[***** SPAM _SCORE_ *****]*

Máis información sobre GTUBE en:

- http://spamassassin.apache.org/gtube/
- Exemplo de correo coa cadea GTUBE

1.12 Antivirus CLAMAV para o correo

Antes de comezar, hai que ter en conta, que os antivirus consumen moita memoria. É probable que necesites engadir máis memoria á máquina virtual. (2GB)

A parte de protexernos do SPAM, podemos configurar o noso servidor PostFix para que escanee todos os correos, en busca de arcquivos que conteñan virus.

Instalamos os seguintes paquetes:

```
apt install clamav clamav-freshclam clamsmtp
```

e actualizamos a base de datos de virus con

```
freshclam
```

Engadimos ao usuario clamav ao grupo correcto

```
usermod -g clamav -G clamsmtp clamsmtp
```

Creamos o seguinte ficheiro, e establecemos o usuario propietario

```
mkdir /var/run/clamav/
touch /var/run/clamav/clamd.ctl
chown clamsmtp /var/run/clamav/
chown clamsmtp /var/run/clamav/clamd.ctl
```

No ficheiro, */etc/clamsmtpd.conf* indicase, a que porto envía clamav os emails escaneados (10025), e en que porto atende peticións (10026).

```
OutAddress: 10025
Listen: 127.0.0.1:10026
```

No ficheiro */etc/postfix/main.cf* engadimos

```
content_filter = scan:127.0.0.1:10026
receive_override_options = no_address_mappings
```

E no ficheiro */etc/postfix/master.cf* engadimos o seguinte codigo, para analizar as mensaxes enviadas, e inxectar de novo na cola smtp as mensaxes que pasan o filtro do antivirus.

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

# For injecting mail back into postfix from the filter
127.0.0.1:10025 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
```

So queda reiniciar os servizos ***postfix***, ***clamsmtp*** e ***clamav-daemon***

```
systemctl restart clamsmtp
systemctl restart clamav-daemon
systemctl restart postfix
```

1.12.1 Comprobación do funcionamento do antivirus

É necesario comprobar que o noso correo sae libre de virus. A mellor maneira é descargar arquivos de test, e adxuntalo a un correo e envialo.

Descargamos os arquivos de proba EICAR

```
wget http://www.eicar.org/download/eicar.com
```

Enviamos o arquivo como adxunto e chequeamos o ficheiro de log */var/log/mail.log* e veremos algo similar a

```
Mar  8 17:12:02 localhost clamsmtpd: 100004: from=info@domain1.com, to=info@domain1.com, status=VIRUS:Eicar-Test-Signature
Mar  8 17:12:02 localhost postfix/smtp[15634]: 4A6C852110: to=<info@domain1.com>, relay=127.0.0.1[127.0.0.1], delay=0, status=sent (
```

1.12.2 Actualización da base de datos de virus

É importante ter actualizada sempre a base de datos de virus, para garantir unha mellor detección.

No caso de que tras un reinicio, o proceso freshclam non se esté executando, podemos engadir unha tarefa no CRON para que todos os dias descargue a base de datos de virus actualizada.

Executamos:

```
crontab -e
```

e engadimos

```
00 1 * * * /usr/bin/freshclam
```