Cyrus imap sasl tls

Por Davidfm

Instalación y configuración del servidor imap cyrus de Carnegie Mellon y de los mecanismos de autentificación y cifrado para implementar un servicio de correo imap robusto, fácilmente administrable y fácil de replicar en backups.

1. Introducción

Cyrus Imap (*http://asg.web.cmu.edu/cyrus/imapd/*) es un potente servidor de correo y news que soporta **POP, IMAP, IMAPS, NNTP** y el lenguaje de scripts de filtrado de mensajes y respuestas **SIEVE** para crear servidores de alta disponibilidad.
Las ventajas que tiene un servidor imap frente a uno pop son considerables. Por una parte, la carpetas se crean en el servidor, no en el cliente, por lo que es fácil realizar un backup de todas las carpetas de los usuarios sin tener que tocar los clientes. Por otra parte, al estar en el servidor es sencillo configurar diferentes modos de acceso a las carpetas que permiten que con una sóla autentificación imap podamos ver todas las cuentas asociadas en caso de que un usuario tenga más de una. Además podemos establecer cuotas, para limitar el espacio en disco que ocupará el correo del usuario.

Cyrus Sasl (*http://asg.web.cmu.edu/sasl/sasl-library.html*) es un framework que provee mediante plug-ins métodos de autentificación comúnmente utilizados por programas de correo como **Cyrus Imap, Courier, Postfix, Sendmail** ,etc Permite definir nuestros propios plug-ins para hacer implementaciones propias de los métodos comúnmente usados o definir nuestros propios métodos. Los métodos que soporta actualmente son éstos:

Intercambio de secretos codificados con **MD5**

-SASL CRAM-MD5

-SASL DIGEST-MD5

Kerberos

-SASL KERBEROS_V4

-SASL GSSAPI(kerberos 5)

Texto plano

-SASL LOGIN

-SASL PLAIN

Otros

-LOGIN(sin Sasl)

-EXTERNAL(servidores de correo externos)

-APOP(Autenficación POP con cifrado)

Los métodos más comúnmente usados por clientes de correo son los de secretos en **MD5** y texto plano, siendo **Kerberos** utilizado principalmente para autentificar clientes servidores entre diferentes dominios. Si vamos a usar autenficación en texto plano(hay que tener en cuenta que las contraseñas en **MD5** son fácilmente reversibles), es necesario establecer una sesión una sesión cifrada a la hora de realizar la autentificación que nos permita intercambiar nuestras contraseñas de una forma segura, en previsión de que puedan ser interceptadas mediante algún ataque de evesdropping(o sea, un sniffer o una suplantación del servidor). Para ellos compilaremos nuestros demonios con soporte para **SSL**, habilitando así un mecanismo llamado **TLS** que, mediante certificado(s) nos permitirá cifrar y además nos posibilitará de un medio para asegurar que el servidor no es suplando (y el cliente si lo requerimos y esta provisto de un certificado cliente).
Hay varias métodos de realizar la autenficación con los plug-ins de **SASL** y también hay varias formas donde guardar las contraseñas y usuarios de nuestros dominios. La más sencilla y la que se trata aquí con prfundidad es autentificar contra la base de datos de berkeley /etc/sasl2/sasldb2. Tiene la ventaja de que no necesitamos instalar ninguna cosa más ya que la base de datos de **Berkeley** es un requisito para instalar **SASL** y además la instalación en chroot es sencilla. La desventaja es que si algún intruso se hace con el archivo obtendrá sin problemas los usuarios y contraseñas. Otros métodos de hacerlo implican a **PAM**, el módulo de autentificación de **Linux** que permite autentificar contra el fichero de passwd de **linux** o mediante algún parche contra una base de datos **MYSQL** o un servicio de directorio como **LDAP**. Esto es adecuado cuando tenemos muchos usuarios, más si utilizamos esa misma base datos para guardar los alias de direcciones y bandejas de entrada del sistema para **Postifx**. La contrapartida es que hacer una instalación chroot usando este método se vuelve complicada e insegura, ya que dejar nuestro archivo de contraseñas en la jaula es

una buena forma de hacerla inútil.

2. Instalacion de Cyrus Imap y Cyrus Sasl

Para instalar cyrus imap debemos instalar primero la base de datos de Berkeley, openssl y cyrus sasl. Normalmente esto ya está incluído en vuestra distribución pero aquí van unos comandos (vilmente copiados de un howto oficial) para instalar los requisitos por si no fuese así. Hay que tener en cuenta donde instala vuestra distribución las cosas, ya que algunas las instalarán por defecto en /usr/local/... y otras en /usr/..., por eso si nuestra instalación instala **Berkeley** o ssl en un path que no es el de defecto tendréis que indicárselo en el script configure.

Instalación de Berkeley DB

```
cd dist
./configure --prefix=/usr/local/bdb
make make install
echo /usr/local/bdb/lib >> /etc/ld.so.conf
ldconfig
```

Instalación de OpenSSL

```
./config shared
make
make test
make install
echo "/usr/local/ssl/lib" >> /etc/ld.so.conf
ldconfig
```

Instalación de cyrus sasl

Configuramos los métodos de autentificación(**plain,login**,etc y contra que vamos a autentificar(**mysql,berkeley**), en el caso que nos ocupa de autentificar contra la base de datos de **Berkeley** no es necesario indicar --*with-pam* ni --*with-saslauthd*

```
 ./configure \
--enable-anon \
```

```
--enable-plain \
--enable-login \
--disable-krb4 \
--disable-otp \
--disable-cram \
--disable-digest \
--with-saslauthd=/var/run/saslauthd \
--with-pam=/lib/security \
--with-dblib=berkeley \
--with-bdb-libdir=/usr/local/bdb/lib \
--with-bdb-incdir=/usr/local/bdb/include \
--with-openssl=/usr/local/ssl \
--with-plugindir=/usr/local/lib/sasl2
make
make install
```

Instalación de saslauthd

El demonio saslauthd se encarga de realizar la autentificación cuando ésta se realiza a través de **PAM**

```
mkdir -p /var/run/saslauthd
cd saslauthd
make testsaslauthd
cp testsaslauthd /usr/local/bin
echo /usr/local/lib/sasl2 >> /etc/ld.so.conf
ldconfig
```

Instalación de Cyrus Imap

```
export CPPFLAGS="-I/usr/include/et"
./configure \
--with-sasl=/usr/local/lib \
--with-perl \
--with-auth=unix \
--with-dbdir=/usr/local/bdb \
```

```
--with-bdb-libdir=/usr/local/bdb/lib \
--with-bdb-incdir=/usr/local/bdb/include \
--with-openssl=/usr/local/ssl \
--without-ucdsnmp \
make depend
make
make install
```

Instalación en Gentoo

La instalación con Gentoo es realmente sencilla con una salvedad. No podemos instalar usando el ebuild el método de autentificación APOP, por lo que si queremos instalarlo deberemos descomprimirlo primero y configurarlo con las opciones que necesitemos con las herramientas que Gentoo provee para ello.
Una vez elegido contra que autentificaremos a los usuarios sólo tenemos que seleccionar las flags adecuadas:

```
USE="-ldap -mysql -pam -static berkdb ssl" emerge cyrus-sasl
```

Esto instalaría SASL con soporte para SSL/TLS y con soporte para usar la autenficación contra sasldb2. La flag -static indica que la librería debe ser compilada como dinámica, es decir compartida por el sistema. Si la hacemos estática será cargada cuando se cargue SASL y no podremos hacer uso de la flag etdyn que nos provee PAX en Gentoo Hardened para hacer más difícil explotar un desbordamiento de pila en caso de que hayamos configurado la pila como no ejecutable. Para instalar cyrus imap utilizaríamos estas flags:

```
USE="-snmp -afs ssl" emerge cyrus-imapd
```

También es necesario crear un usuario cyrus en el sistema, pero en un sistema Gentoo ya está creado por defecto.,si no es así creadlo con el comando *adduser*

3. Configuracion de SASL y el servidor Imap

Las opciones **SASL** a configurar en el archivo /etc/imapd.conf no son muchas y lógicamente van relacionadas con los mecanismos de autentificación y la forma de hacerla. Vamos a ver algunas opciones interesantes que podemos especificar , para ver las lista completa consultad man

imapd.conf

srvtab: path hasta las claves utilizadas en la autentificación **Kerberos**, por defecto */etc/srvtab servername:* el nombre **DNS** que se mostrará cuando un cliente conecte con el server, por defecto lo obtenido con la llamada *gethostname()*

imapidresponse: indica si se debe mostrar el nombre del programa **imap** y la versión, por defecto sí. Yo personalmente cambiaría esto si es un servidor accesible desde internet.

maxmessagesize: el tamaño máximo que deben tener los mensajes para que el servidor acepte

Una vez elegido nuestro método de autentificación y nuestros mecanismos lo configuramos con dos opciones, en este caso correspondientes a autentificar contra la base de datos sasldb2

sasl_pwcheck_method: auxprop
mech_list: plain login cram-md5

Si necesitamos autentificar contra algo que necesite privilegios de root, como por ejemplo contra **PAM** en sus diferentes formas(el archivos de contraseñas de **Unix**,una base de datos, etc) deberemos utilizar saslauthd, un demonio que sí tiene privilegios para hacerlo

sasl_pwcheck_method: saslauthd

En el caso de que nuestro servidor **imap** aloje varios dominios virtuales, es necesario especificarlo en el archivo también y especificar el dominio que será considerado el de defecto de la máquina

virtual_domains: yes
default_domain: nauto.com

Los usuarios del dominio por defecto de la máquina son reconocidos por cyrus con sólo el nombre, mientras que el resto pertenecientes a los dominios virtuales deben ser refereciados por nombre@dominio al que pertenecen, tanto a la hora de autentificar como a la hora de crearlos. Debemos especificar un administador global que administrará el servidor

admins: cyrus

Podemos especificar también administradores de dominio con permisos en el dominio en el que pertenecen especificando el nombre@dominio

```
admins: cyrus elputoamode@dominio.es
```

Para crear los usuarios utilizamos el comando *saslpasswd2*, siendo el primer usuario que tenemos que crear el que hemos especificado como admin global

```
echo contraseña_fuerte_de_cojones | saslpasswd2 -c -p cyrus
```

En este momento la base de datos salsdb2 se crea si no está creada ya. En Gentoo Linux si está creada, pero si instalamos **Postfix** no tenemos permiso para leerla como **cyrus** por lo que no podremos autentificar contra **SASL**, por lo que deberemos cambiarle los permisos con

```
chmod 644 /etc/sasl2/sasldb2
```

o hacerla propriedad de **cyrus**

```
chown cyrus:mail /etc/sasl2/sasldb2
```

Permtir acceso de lectura a todos los usuarios no es lo más adecuado, por lo que se hace útil contar con un sistema de ficheros que tenga la posibilidad de definir acls más complejas que los simples permisos **POSIX de linux**. A continuación podemos empezar a crear los usuarios que vamos a utilizar especificando el dominio al que pertenecen si este no es el de defecto

```
echo "contraseña" | saslpasswd2 -c -p usuario -u dominio_virtual
```

Para listar los usuarios que vamos creando y si pertenecen al dominio correcto ejecutamos el comando
sasldblistusers2

3.1 Cyradm

Una vez creados los usuarios para **SASL** ,es necesario crear para **Imap** las carpetas que alojarán el correo del cliente. **Imap** utiliza una jerarquía de dominios para identificar a los clientes, como ya hemos explicado y las carpetas se crean debajo de /var/imap/user/primeraletradelnombre. Para crear las carpetas y configurar los accesos y quotas podemos utilizar **cyradm**, un comando que

conecta con el servidor y para el que también hay una interfaz web que no vamos a explicar en este documento(más que nada porque ya me conozco la seguridad de las aplicaciones web). Primero tenemos que instalar el paquete **cyradm**

```
emerge cyrus-imap-admin
```

Para conectar con nuestro servidor **imap** , siendo root ejecutamos

```
cyradm -user cyrus localhost -pass la_de_cyrus
```

Una vez dentro del servidor autentificados como administrador global, podemos proceder a la creación de las carpetas y su configuración. Algunos comandos de **cyradm** son:

- cm user.usuario (crea un nuevo usuario para imap asignándole una carpeta)

- lam carpeta (muestra los permisos asociados con esa carpeta)

- sam carpeta usuario permisos (asigna o quita permisos a carpeta para el usuario especificado)

Los permisos que podemos utilizar son:

l listar carpeta
r leer carpeta
d borrar carpeta
c crear carpeta
a establecer permisos
i insertar
p mandar correo
s mantener información entra sesiones
sq carpeta número (asigna una cuota a carpeta especificada para que el usuario no pueda almacenar en su carpeta más de número KB)

Podemos especificar comodines a la hora de usar los comandos, por ejemplo para listar todos los atributos de todos los usuarios creados podemos ejecutar

```
lam user.*
```

Los pasos ha seguir para crear un usuario serían estos , con la salvedad de que si estamos creando un usuario perteneciente a un dominio virtual debemos especificar el nombre completo(la dirección de email)

```
cm user.david
sam user.david root all ( nos asigna permisos para administrarla)
sq david 1000
```

Si algún usuario tiene varias cuentas de correo, podemos configurar las carpetas para que con una sóla autentificación **imap** pueda ver las dos bandejas de entrada. Para ellos sólo tenemos que darle permiso a los dos nombres de usuarios para que puedan ver las carpetas a las que se descarga el correo de las diferentes cuentas. **Probando la instalación** En cuanto tenemos creado un usuario es conveniente probar a ver si el servidor está correctamente instalado, para ellos podemos usar el comando **imtest**

```
imtest -m login -u usuario nombre_dns_del_server -t ""
```

Vemos que el servidor nos muestra los mecanismos soportados y como **imtest** ha intentado realizar un login utilizando TLS (opción -t "") con nuestro servidor. Si el login se ha realizado con éxito, veremos que la conexión se ha cifrado a 256Kbs. Si es así, ya hemos terminado de configurar nuestro servidor imap, con la salvedad de tener que crear el resto de usuarios y que tenemos que especificar en nuestro **MTA**(Mail Transport Agent) a que carpeta exactamente tenemos que descargar el correo. Si todo ha ido mal, consultad los logs en **/var/log/messages** para obtener una idea que ha fallado

4. TLS y generacion de certificados

Para utilizar los mecanismos de **SSL/TLS** debemos usar certificados para el servidor y para los clientes si los queremos usar en la autentificación. Para crear un certificado válido necesitamos

una autoridad certificadora(**CA**) que nos de un certificado raíz o nos firme los certificados a usar. Si queremos que nuestros certificados tengan validez en todo el mundo sin interacción del usuario debemos contratar una entidad certificadora. Si es para uso interno podemos crear nosotros un certificado raíz **CA** para nuestra red e instalarlo y/o los certificados creados con ella en el almacen de certificados de la red. Otra opción es utilizar CACert (*http://www.cacert.org/*), una entidad sin ánimo de lucro que provee certificados digitales gratuitos.

Creación de una autoridad certificadora

Para crear un certificado raíz utilizaremos OpenSSL cuya herramienta de la línea de comandos del mismo nombre nos permite realizar todas las operaciones necesarias con certificados. Para configurarlo modificaremos el archivo /etc/ssl/openssl.cnf Primero rellenaremos información relativa en la sección CA_default a la ubicación de los archivos relacionados con la **CA**,como el directorio de la **CA** o el path y nombre de los archivos de claves y certificados creados por defecto. La opción *unique_subject* que viene comentada y puesta a "no", implica que no podemos crear más de un certificado con una misma petición de certificado. La sección **req_distinguised_name** nos permite especificar datos que serán incluídos en el certificado al firmar, como el nombre de la empresa, dirección de email ,etc
Una vez hecho esto nos vamos hasta la carpeta ráiz de la instalación de OpenSSL , típicamente / etc/ssl y ejecutamos el script CA.pl o CA.sh (como usuario root) para crear las carpetas y el certificado raíz
misc/CA.pl -newca

y respondemos a las preguntas que nos haga. Pulsaremos enter para crear un certificado y le introducimos una contraseña al certificado, necesaria cada vez que tengamos que firmar un certificado cliente. Introducimos información personal y tras eso vemos que se han creado una carpeta demoCA/private con un archivo cakey.pem que contiene la clave RSA de 1024 bits de nuestra CA y un archivo demoCA/cacert.pem que es nuestro certificado raíz.

Generación de los certificados cliente

Para generar el certificado que usaremos con nuestro servidor imap necesitamos realizar una petición de certificado con el comando

```
misc/CA.pl -newreq
```

Nos pedirá una contraseña y se creará un archivo con la petición de certificado *newreq.pem*. Para firmar la petición y crear nuestro certificado ejecutamos

```
misc/CA.pl -sign
```

Con esto se habrá creado nuestro certificado *newcert.pem* y un archivo en *demoCA/newcerts* de extensión .pem que es nuestro certificado(son iguales). Si queremos hacer una instalación desatendida de servidores necesitamos también la clave **RSA** sin cifrar en un archivo a parte, para ello la creamos con

```
openssl -rsa < newcert.pem > newkey.pem
```

Por supuesto los nombres *newcert.pem* y *newkey.pem* pueden ser cualquiera con el que hayais renombrado el certificado.

Instalación en Gentoo Linux

Al instalar **cyrus-imap**, Gentoo ya genera un certificado y una archivo de claves situados en /etc/cyrusimapd para nuestro servidor, por lo que la instalación de reduce a ejecutar emerge, configurar el servidor y generar los certificados cliente.

```
emerge -v cyrus-imapd
```

Configuración de Cyrus con SSL/TLS

Para que **SASL** utilice **SSL/TLS** a la hora de autentificar debemos indicarle donde se encuentran el archivo con los certificados del servidor imap, sus archivos de clave y el archivo donde se encuentran los certificados de las autoridades certificadoras para que puedan ser verificados los certificados. Esto se hace mediante estas opciones:

tls_cert_file: /etc/cyrusimapd/server.cert
tls_key_file: /etc/cyrusimpad/key.file
tls_ca_file: /etc/postfix/certs/server.pem

Si queremos obligar a los clientes de correo a autenficarse con un certificado incluímos la directiva

tls_require_cert = 1

También podemos especificar la mínima capa bajo la que se autentificará a los clientes para exigir cifrado en ellas

sasl_minimun_layer: (0 por defecto)

5. Solución de problemas

En los logs veo un error de method not supported al utilizar la herramienta imtest

Si has configurado el servidor con **SSL/TLS** añade la opción -t "" a **imtest** para que haga la prueba con autentificación cifrada.

No me puedo logear a cyradm

Comprueba con el comando sasldblistusers2 que tu usuario administrador está creado y su dominio es el de defecto de tu máquina que administra y que está configurada la opción admin en /etc/imapd.conf con los usuarios que administran los dominios.

Al intentar crear una carpeta en cyradm me devuelve "permission denied"

Tienes que ser root para poder crear carpetas imap

¿Qué método utiliza lo que Microsoft llama autentificación con contraseña segura(SPA) en cyrus?

Los clientes Outlook pueden utilizar logins planos o SPA, que utiliza CRAM-MD5

Este manual también puende verlo en: cyrus imap sasl tls

año 2004