Seguridad web-ftp-smtp

Seguridad Web. Apache.

Directivas

Instalación de mayor seguridad.

Mover httpd al directorio /sbin
Mv /apache/bin/httpd /sbin/httpd
Actualizar permisos
Chown 0 /sbin/httpd → Propietario root
Chgrp 0 /sbin/httpd → Grupo root
Chmod 511 /sbin/httpd → Permiso de lectura y ejecución para root y ejecución para el resto.
Colocar el valor de la variable httpd en la nueva ubicación.
Vi /apache/bin/apachectl
httpd = /sbin

TimeOut → Para evitar que se colapse la red Apache no mantendrá las conexiones del cliente activas indefinidamente.

MaxClients → Evita ataques de denegación de servicio.

Listen, Port → Para cambiar el puerto por el que escuchar el servidor.

LoadModule, AddModule → Cargar y activar exclusivamente los modulos necesarios para evitar posibles agujeros de seguridad.

ServerName → No especificar el nombre de la máquina donde esta el servidor para evitar dar más información de la necesaria al exterior.

User, Group → No debe aparecer ningún grupo o usuario del sistema para ejecutar el servicio. Se usará nobody, nogroup o apache como usuarios y grupo.

UserDir enable/disable→ Deshabilitar usuarios específicos para administrar el sistema.
Ej. UserDir disable root ftp → Deshabilita las paginas personales para estos dos usuarios.

Un ejemplo sobre los permisos que deberian tener las carpetas seria el siguiente.
/home/pablo/web/*
El directorio pablo permisos pablo:pablo 711
El directorio web permisos pablo:apache 750
El resto de los directorios pablo:apache 644

DirectoryIndex → Asegurar que en los directorios existen los ficheros especificados para que no pueda verse el contenido del mismo.

Control de acceso

```
<Directory directorio_a_restringir>
#Autentificación basica. Restriccion para permitir a pocos
        order deny, allow
```

```
        deny from all
        allow form 192.168.0.   apache.org
        options indexes
        DirectoryIndex index.html
</Directory>

<Directory directorio_a_restringir2>
<Files restringido.html>
#Autentificación basica. Restriccion para permitir a muchos
      order allow, deny
      allow from all
      deny from 192.168.
      options indexes
            DirectoryIndex index.html
</Files>
</Directory>
```

Ficheros de log

ErrorLog, LogLevel, CustomLog y LogFormat → Es importante para administrar y verificar la seguridad del sistema realizar un buen control de los ficheros de logs.

```
ErrorLog /var/log/error_log
Loglevel warn
Logformat “Ip del cliente: %h; hora %t” miformato
Customlog logs/access_log miformato
```

Alias de directorios

Aliases (Redirección interna)→ Es buena política no usar rutas reales y si rutas relativas para no dar pistas de nuestra estructura de directorios.

Autentificación básica

```
<Directory /private/projects>
      Authtype Basic → autentificacion basica
      AuthName “Fichero empleados” →se mostrara en el equi remoto
      AuthUserFile /etc/httpd/usuarios-empleados → fichero con
      usuarios
      Require valid-user
</Directory>
```

El control de los usuarios se hará creándolos y manteniéndolos usando el fichero httpasswd. Es necesario almacenar el fichero de claves en un lugar seguro.

```
Httpasswd –b –c fichero usuario password
```

Autentificación básica php

```
$username = “tribilin”
$pwd = “secreta”
if (!isset($PHP_AUTH_USER)){
      Header(“WWW-Authenticate: Basic realm=\”Zona protegida\””);
      Header(“HTTP/1.0 401 Unauthorized”);
```

```
      Echo “Imposible ejecutar la autorizacion\n”;
      Exit
Else
      If (($PHP_AUTH_USER == $username) && ($PHP_AUTH_PW == %pwd)){
            Echo “Autorizacion ejecutada para $username”;
      Else
            Echo “$username no tiene autorización”;}
}
```

Autentificación basica con PHP y mysql

```
<?function autentificar() {
header("WWW-Authenticate: Basic realm='Sistema de autenticación'");
header("HTTP/1.0 401 Acceso Denegado");
echo "<font 'verdana,arial' size='2' color='red'><center>Acceso
denegado al Centro de Soporte Tecnico de NCT Informatica.\n
</center></font>";
exit;
}

if(!isset($PHP_AUTH_USER)){ autentificar(); }
else {
 $link=mysql_connect("localhost", "root");
     mysql_select_db("gestion",$link);
     $sql=sprintf("SELECT * FROM clientes WHERE nif='%s' AND
clave='%s'",$PHP_AUTH_USER,$PHP_AUTH_PW);

 $sql=mysql_query($sql);
       if(mysql_num_rows($sql)==0){ autentificar(); }


      session_start();

      $usuario[0] = $PHP_AUTH_USER;
      $usuario[1] = $PHP_AUTH_PW;
      session_register(usuario);

}
?>
```

SSL. (Secure Socket Layer)

SSL es una especificación propietaria de Netscape puesta en dominio público para la definición de canales seguros sobre TCP cuyo objetivo es la realización de conexiones seguras a servidores www que permitiera, por ejemplo, enviar números de tarjetas de crédito a través de un formulario.

Debe asegurar

- La integridad: La garantía de que los mensajes que enviamos o recibimos no han sido modificados.

- La confidencialidad: Nadie sin autorización puede leer la información transmitida, y garantiza que efectivamente la recibe quien debe recibirla.

2.- Puesta en Marcha:

2.1.- Requerimientos:

Apache. www.apache.org

Librerias SSL. www.openssl.org

Modulo para el Apache. www.modssl.org

Instalación del Apache+SSL:

Se puede realizar este proceso de dos formas:

1. Instalación de los diferentes programas por separado
    a. Instalar Apache, openssl y modssl de forma independiente (./config && make && make install)
    b. Crear el certificado
        i. **Openssl req –new > pepe-informatica.csr** (Crea un certificado con los datos del usuario y una clave privada (PEM passphrase) de 1024 bits. Cuando pregunte por Common name hay que darle el nombre del servidor web (www.pepe-informatica.com). Se debe responder sobre la organización.
        ii. **Openssl rsa –in privkey.pem –out pepe-informatica.key** (Elimina la frase de paso de la clave privada. Pepe-informatica.key solo debe ser leído por el administrador y apache. Se elimina cualquier información que de pistas sobre como averiguar la clave privada.
        iii. **Openssl x509 –in pepe-informatica.csr –out pepe-informatica.cert –req –signkey pepe-informatica.key –days 365**. Con eso se obtiene el certificado que podemos usar hasta obtener uno de la autoridad de certificados. (Opcional)
    c. Ahora debemos configurar Apache para que pueda responder ante peticiones https (HiperText Transfer Protocol Secure). Las directivas son :
        i. Addmodule mod_ssl.c
        ii. …
        iii. LoadModule ssl_module modules/mod_ssl.so (libssl.so)
        iv. …
        v. SSLLog logs/ssl.log
        vi. SSLLoglevel warn
        vii. …
        viii. <VirtualHost www.pepe-informatica.com:443>
        ix. SSLEngine on #Activa el protocolo SSL/TSL
        x. SSLCertificateFile /rutacertificado/pepe-informatica.cert
        xi. SSLCertificateKeyFile /rutacertificado/pepe-informatica.key
        xii. ...
        xiii. </VirtualHost>
    d. Reiniciar Apache. Puede ser necesario reiniciar apache con la opcion –D SSL o bien con /usr/sbin/apachectl startssl;
    e. Podemos comprobar que la conexión es correcta mediante :

i. **Openssl s_client –connect pepe-informatica.com:443** (Si no se produce ningun error es que esta correctamente instalado).

El puerto estandard es el 443. De todas maneras, se puede configurar para que sea otro, y lo mas importante: A la hora de navegar no es http://localhost:443 (o cualquier otro), sino que es httpS://localhost, o httpS://localhost:(cualquier otro) (Hyper Text Transfer Protocol Secure).

La otra forma de instalar es configurando la instalación:

- Descomprimir e instalar openssl. Se instala con la configuración básica ./configure
- Posteriormente mod_ssl. Se le pasará la ruta donde se descomprimió Apache. ./configure –with-apache=../apache_x.y.z
- Y Finalmente se compila Apache
  - SSL_BASE =../openssl
  - ./configure –enable-module=ssl –prefix=/www –enable-module=so
  - make
  - make certificate (Completar los datos solicitados para la creación del certificado)

Seguridad FTP. ProFTPd.

FTP son las iniciales de File Transfer Protocol (Protocolo de Transferencia de Ficheros o archivos) y aluden a uno de los protocolos más utilizados en Internet. Su función es permitirnos bajar (o subir) información desde (o hacia) los servidores que se encuentran en Internet.

Los objetivos del FTP son:

1) promocionar el uso compartido de ficheros (programas y/o datos),
2) animar al uso indirecto o implícito (a través de programas) de servidores remotos,
3) hacer transparente al usuario las variaciones entre la forma de almacenar ficheros en diferentes ordenadores, y
4) transferir datos fiable y eficientemente.

El FTP, aunque puede ser utilizado directamente por un usuario en un terminal, está diseñado principalmente para ser usado por programas.

El ProFTPd

ProFTPd es una de las alternativas para Linux en el campo de los servidores FTP. Presenta cualidades muy apreciadas en este terreno, como son escalabilidad, rendimiento, seguridad y facilidad de configuración. Permite definir dominios virtuales, FTP anónimo y control de permisos.

Proftpd es un software potente y seguro. Su fichero de configuración utiliza una sintaxis similar a la de Apache, lo que permite homogeneizar su configuración. Ha sido elegido como servidor ftp oficial por la distribución Debian/GNU Linux. Su elección se debe a sus numerosas funcionalidades y por la gran cantidad de programadores que trabajan en él corrigiendo errores.

El uso de módulos permite fácilmente extender sus funcionalidades. Existen módulos para:

- Gestión de quotas (de usuario)
- Otros sistemas de authenticación como LDAP, Mysql, Postgresql, ...
- Gestión de ratios (por ejemplo, el usuario no puede descarga mas de 1 MB diario)
- Gestión de ancho de banda

ProFTPD ofrece las siguientes características:

- Un único fichero de configuración, con directivas y directivas de grupo intuitivas para cualquier administrador que haya usado el servidor web Apache.
- Configuración de acceso por directorio: ".ftpaccess", similar al ".htaccess" de Apache.
- Fácil de configurar múltiples servidores FTP virtuales y servicios de FTP anónimos.
- Diseñado para correr tanto como servidor “stand-alone”, como desde el inetd o xinetd, dependiendo de la carga del sistema.
- Los directorios del servicio de FTP anónimo no requiere de ninguna estructura de directorios específica, sistemas de ficheros binarios u otro tipo de sistema de ficheros.

- ProFTPD no ejecuta ningún programa externo en ningún momento. En entornos modernos de Internet, tales programas pueden ser una pesadilla en cuanto a seguridad se refiere. El código fuente está disponible (y siempre lo estará) para así poder comprobarlo.
- Directorios y ficheros ocultos, basados en el estilo de permisos de Unix o propiedad por usuario/grupo.
- Corre como un usuario no privilegiado configurable en modo "stand-alone", para disminuir la probabilidad de ataques que proporciona su actitud como "root". **Nota: Esta característica depende de la capacidad del sistema Unix.**
- Soporta autenticación de usuarios y utmp/wtmp. La autenticación es compatible con el estándar wu-ftpd, con posibilidad de ampliar esta característica.
- Soporta contraseñas ocultas (shadow password), incluyendo soporte para cuentas que han caducado.

Se obtienen los fuentes de la dirección http://proftpd.linux.co.uk/. Se trata de la versión 1.2.4 que guardamos como */root/proftpd-1.2.4.tar.gz*.

Se descomprime el fichero con el comando:

tar xvzf proftpd-1.2.4.tar.gz

A continuación se configura para la compilación:

./configure –sysconfdir=/etc –localstatedir=/var –prefix=/usr

con *sysconfdir* se coloca el archivo de configuración en */etc*; con *localstatedir* ubicamos los ficheros del estado en tiempo de ejecución en */var*; por último se utiliza *prefix* para colocar todos los directorios del programa bajo */usr*.

Resta compilar los fuentes e instalar, respectivamente con los comandos **make** y **make install**.

Al igual que en Apache lo primeroQue el demonio se ejecute con permisos de root no es adecuado por eso hay que ejecutarlos con usuario ftp o nobody y grupo ftp o nogroup.

Iniciar el servicio
Cd /usr/local/etc/rc.d
Cp proftpd.sh.sample proftpd.sh
/usr/local/etc/rc.d/proftpd.sh Start

• *Control de acceso remoto únicamente*

Entendemos por acceso remoto el acceso desde otro equipo distinto a aquel en el que se encuentra el servidor. Inhabilitamos el acceso al servidor desde dicha máquina :

```
# Para que solo se pueda acceder remotamente
<LIMIT LOGIN>
  Deny 192.168.10.1
</LIMIT LOGIN>
```

donde *192.168.10.1* es la dirección IP del citado equipo. El bloque *LIMIT* es de aplicación global.

- ***Posibilidad de acceso en modo usuario normal (Usuario con una cuenta creada en el sistema, p.e. prueba), de forma que dicho usuario no pueda salir de los directorios de los cuales es propietario (chroot).***

Aqui necesitamos que se verifiquen unas condiciones previas:

1) la cuenta del usuario existe previamente en el sistema;
2) tiene asignado un shell compatible con el sistema;
3) todos los directorios de los cuales es propietario están bajo su directorio personal.

```
#Para que el usuario no pueda subir a un directorio de
nivel mas alto que el suyo

DefaultRoot                ~
```

El carácter **~** apunta al directorio personal del usuario dentro del sistema. La directiva ***DefaultRoot*** señala el directorio a partir del cual no puede ascender el usuario.

- ***Posibilidad de acceso como usuario anónimo.***

Para habilitar el acceso en modo anónimo añadimos un bloque *Anonymous* al fichero de configuración, seguido del directorio raíz para el mismo. Es necesario asimismo crear un usuario y un grupo en el sistema para dicho usuario. Además añadimos un alias para facilitar su uso. Dentro de este bloque se configurará el acceso para este tipo de conexión.
Por defecto proftpd lee del fichero /etc/passwd para el login de usuarios. Por lo que para validar usuarios ftp solo basta tener una cuenta en el sistema. En caso de no querer tener usuarios de sistema para acceder a ftp estos se podrían validar usando por ejemplo mysql, ldap, radius
Useftpusers off directiva para que no se lean los usuarios del fichero /etc/ftpusers usado por el wu-ftpd.

```
<Anonymous /home/ftp>
  User                          ftp
  Group                         ftp
  # We want clients to be able to login with "anonymous" as well as "ftp"
  UserAlias                     anonymous ftp

</Anonymous>
```

- ***Limitado el acceso a 3 usuarios simultáneamente.***

  Con la directiva ***MaxClients*** especificamos el número máximo de usuarios que se pueden conectar. Al encontrarse dentro del bloque *Anonymous* solo se aplica a aquellos conectados en tal modo. Evita ataques de denegación de servicio.

```
        # Limit the maximum number of anonymous logins

             MaxClients          3
```

- ***Limitada la conexión sin actividad en 5 minutos.***

  TimeoutIdle no puede incluirse directamente dentro del bloque *Anonymous*, así que lo incluimos en el bloque principal, siendo válido para todos los bloques que aparecen posteriormente. El tiempo viene dado en segundos.

```
        # Tiempo tras el cual el servidor desactiva la conexión si no hay
        actividad

             TimeOutIdle         300
```

Se puede limitar el uso intensivo de cpu/memoria de sesiones ftp mediante el uso de directivas.

```
RLimitCPU session 10
RLimitMemory session 4096
RLimitMemory daemon 8192 max
```

Allowoverride off indica a proftpd deshabilita el chequeo de los directorios en busca del fichero .ftpaccess lo que reduce accesos de lectura escritura en disco.

- ***Ficheros de logs***

Existen tres tipos de ficheros de logs que puede crear el demonio proftpd. *Transferlog* aquí es el más común y almacena un registro de la transferencia de los ficheros. *Systemlog* que almacena respuestas del demonio. Y *extendedlog* que es usada para crear ficheros configurales de los usando logformat. No dar permisos de escritura donde esten los ficheros de logs para que estos solo puedan ser modificados por el root.

- LogFormat nombre “formato”
  - Logformat default “%h %i %u %t”
- TransferLog /ruta/fich
  - TransferLog /Var/log/profptd/transfer
- ExtendedLog /ruta (AUTH|WRITE|READ|ALL) nombre.
  - ExtenderLog /var/log/proftpd.all.log ALL default.

- ***Existencia de un subdirectorio con posibilidad solamente de descarga de ficheros.***

En nuestra configuración del usuario anónimo negando por defecto la posibilidad de escritura, solo es posible la descarga de ficheros en todo el árbol de directorios:

```
        # Limit WRITE everywhere in the anonymous chroot
        <Limit WRITE>
            Deny all
        </Limit>
```

- ***Existencia de un subdirectorio oculto (no visible con un ls) con posibilidad únicamente de escritura (sin sobreescritura).***

  Creamos el subdirectorio *privado* y lo hacemos propietario. Incluimos un bloque dentro de la configuración del usuario anónimo para este directorio, con la directiva ***HideUser*** que oculta los ficheros cuyo propietario sea el usuario indicado. Desactivamos la sobreescritura con el comando ***AllowOverwrite off*** y añadimos un subloque ***Limit WRITE*** con el que habilitamos la escritura:

```
      # Restringir el acceso al directorio 'privado'
      <Directory privado>
            AllowOverWrite      off
            HideUser usuario
            <Limit WRITE>
                  Allow all
            </Limit>
     </Directory>
```

- ***Creación de un subdirectorio para acceso restringido del usuario prueba (a crear) con acceso en modo anónimo.***

Para reconocer al usuario *aso* es necesario que se identifique; como además queremos que utilice acceso anónimo creamos un nuevo bloque ***Anonymous*** donde requerimos la contraseña del usuario, utilizando ***AnonRequirePassword on***. Localizamos la entrada del usuario en el mismo directorio que el usuario anónimo anterior ***/home/ftp*** y lo configuramos de manera similar.

```
# Usuario 'prueba' con acceso anónimo
<Anonymous /home/ftp>
User                        aso
Group                       ftp
AnonRequirePassword         on
HideUser usuario
<Directory privado>
  AllowOverwrite            off
  <Limit WRITE>
    AllowAll
  </Limit>
</Directory>
</Anonymous>
```

Además, en la configuración del usuario anónimo básico hemos de limitar el acceso al directorio *prueba*, utilizando la directiva ***IgnoreHidden on***; este directorio parecerá no existir para el usuario anónimo:

```
# Restringir el acceso al directorio 'prueba'
<Directory prueba>
  HideUser prueba
  <Limit ALL>
     IgnoreHidden          on
  </Limit>
  </Directory>
```

Seguridad SMTP. Postfix.

1.- Introducción

El servicio de correo electrónico consta de dos partes bien diferenciadas: aquella con la que trata el usuario, y aquella que se encarga de transportar los mensajes del origen al destino. A menudo hay un componente adicional encargado de distribuir el correo que llega a la maquina destino a una ubicación especial dentro de esta, propia de cada usuario. Los nombres de estos componentes son:

- MUA (Mail User Agent): es un programa que permite leer y escribir correos. Suelen tener muchas funcionalidades que superan la estricta lectura y composición de mensajes, como el mantenimiento de libretas de direcciones, gestión de anexos, gestión de múltiples carpetas determinadas, todo ello automáticamente y en funcion de las características del mensaje, etc. Nombres habituales de Musa son: mail, elm, pine, kmail, eudora, outlook express, pegasus, etc
- MTA (Mail Transfer Agent): es un programa encargado de recoger los mensajes y enviarlos, comunicando para ello con otros MTA según sea preciso. Lo normal es que funcione como servicio. En linux se implementan como uno o más demonios. El MTA más famoso y utilizado es sendmail; otros MTAs son Postfix, Qmail...
- Utilidades diversas: dependiendo de las circunstancias, se usaran otras pequeñas utilidades adicionales, que se encargaran de colocar los mensajes en el buzón de cada usuario, de recoger el correo de servidores externos, etc. Postfix usa procmail para la primera función, aunque también esta fetchmail o vpopmail...

Instalación.

Descargar de la web oficial la última versión del servidor de correo Postfix(http://www.postfix.org) y descargarnos su última versión, la 2.0.10. Una vez descargada la aplicación, pasamos a su instalación, realizando los siguientes pasos:

• Antes de iniciar la instalación, lo que hacemos es desactivar el demonio Sendmail para que no nos provoque conflictos. Esto lo hacemos con la orden setup.

```
# setup (y deshabilitar el demonio sendmail)
```

• A continuación desinstalamos Sendmail de la siguiente forma:

```
# rpm –e sendmail nodeps
```

• Luego creamos un usuario llamado postfix y el grupo postdrop.

```
Groupadd postdrop
Useradd –g postdrop postfix
```

• verificar en /etc/aliases, que el usuario postfix se corresponda con root:

```
postfix: root
```

• Tras a ver realizado estos pasos, descomprimimos los fuentes en algún directorio:

```
# tar zxvf postfix-2.0.10.tar.gz
```

• Nos situamos en el directorio en donde hemos descomprimido los fuentes y ejecutamos las siguientes ordenes para realizar la instalación en el sistema:

 # make
 # make install

• Si todo ha ido bien tendremos instalado nuestro servidor PostFix.

. **Configuración**: Editamos el fichero /etc/postfix/main.cf. A continuación mostramos las líneas más importantes de este fichero que corresponde a lo que nos pedía la práctica. Default_privs indica los privilegios por defecto del agente de entrega de correo para ejecutar un comando o abrir un archivo. Generalmente se usa nobody ya no se debe especificar un usuario con privilegios o el usuario postfix.

default_privs = nobody

Especificamos el nombre del host (**$myhostname**). mynetworks se usa para indicarle a Postfix que máquinas, distinguidas por IP o dirección son consideradas locales y las pueden usar el servidor de correo para envíos. mynetworks puede ser una colección de Ips o una clase completa.

myhostname = equipo1.sopa.dis.ulpgc.es
para indicar que direcciones de correo pueden enviar correo a través de nuestro servidor y cuales no pueden enviar a nuestro servidor.
mynetworks = 192.168.10.0/28 #, 127.0.0.0/8

El parámetro **mydestination** especifica a que dominios entregar localmente, en vez de enviarlo a otras maquinas.

mydestination = $myhostname, localhost.$mydomain

La opcion **relay_domains** restringe los dominios donde los clientes usan un servidor de correo para enviar correo (relay) o que destinos va a servir nuestro servidor de correo. (solo se deberá permitir ***relaying*** únicamente a las máquinas de su propia red o dominio). Esto lo logramos con la opción relay_domains.

relay_domains = $mydestination

La opción **mail_spool_directory** indica el directorio donde se almacena los buzones de correos.

mail_spool_directory = /var/spool/mail

Otro requisito podría ser que el tamaño máximo del buzón de correo será de 10 Mbytes. Esto lo logramos con la sentencia **message_size_limit** que indica el tamaño máximo de un mensaje que por defecto es de 10240000 bytes y con la sentencia **mailbox_size_limit** que nos indica el tamaño máximo del buzón de correo.

message_size_limit = 10485760
mailbox_size_limit = 51200000

Para deshabilitarlas se pueden poner a 0.

Otros Mecanismos de seguridad

Notificaciones al postmaster
Mediante la directiva notify_classes el administrador puede indicar a postfix el tipo de incidencias que puede notificar.
Bounce: si un mensaje no puede ser enrutado, se envia otro mensaje al remitente y una copia al postmaster incluyendo el original.
2bounce: si la notificación de error de encaminamiento genera tb error de encaminamiento se envia notificación al postmaster.
Delay : se informa al postmaster de que hay mensajes propuestos por problemas de encaminamiento.
Policy: se informa de peticiones rechazadas de entrega de mensajes. Normalmente se debe a que el interlocutor o el propio mensaje incumple la politica definida para la aceptación de correo. Muy util para conocer intentos fallidos de uso de la estafeta como relay.
Protocol : se informa sobre incidentes de protocolo.
´resource : se comunica a postmaster que un mensaje no ha sido encaminado por problemas de recurso del sistema.
Software: comunica que un mensaje no ha sido encaminado por problemas de software.
Lo habitual es **notify_classes = resource, software, policy**

Filtrado por cabeceras
Un mecanismo que proporciona postfix y que se esta usando ampliamente contra la propagación de virus es la posibilidad de filtrar mensajes en base a las cabeceras de los mismos y a patrones definidos mediante expresiones regulares. De forma que todo mensaje que contenga una cabecera que cumpla un determinado patrón se rechaza automáticamente. Un ejemplo de ficheros de patrones podría ser:

```
# header checks
# Virus W32/Frethem.K/J (15/7/02
/^Subject: Re: Your password!/
```

si esto estuviera en un fichero /etc/postfix/header_checks se podría activar con:

```
header_checks = regexp:/etc/postfix/header_checks
```

Filtrado por contenido
Se pueden filtrar contenidos de mensajes.

```
#Body checks
/Accept credit cards/   REJECTS
/Nude Celebrities/      REJECTS
```

Se rechazaran todos los mensajes que en cualquier parte del cuerpo aparezcan alguna de las palabras o frases indicadas. Suponiendo que el fichero es /etc/postfix/body_checks se podria activar el filtrado con
Body_checks = regexp:/etc/postfix/body_checks