Zimbra como solución libre a un servidor de correo

por Alejandro Moreno
amperisblog[@]gmail.com
http://amperis.blogspot.com

28 de enero de 2008

..para todos aquellos que quieran instalar un servidor de correo open-source profesional y tengan la suficiente destreza para ver que puñeta de librería le falta cuando esta compilando.

Introducción

Cuando trabajas en un empresa pequeña y con pocas ganas de gastar dinero soluciones como montar un Microsoft Exchange quedan descartadas, por eso un administrador se tiene inclinar por soluciones más baratas; y más barato que un Linux no hay nada.
El único problema que encuentro es el esfuezo en el aprendizaje-instalación-configuración que tiene Linux (Postfix, Sendmail, QMail., etc) en comparación con Exchange. Eso si, una vez montado tú sistema de correo en Linux olvídate de el.

En este documento se explica el proceso de instalación, las impresiones y las modificaciones hechas durante más de un año a un sistema de correo Linux basado en la versión libre de Zimbra Collaboration Suite que implementé en un trabajo.

Zimbra te permite montar todo un sistema de correo electrónico basado en paquetes libres. Se trata de una recopilación de paquetes ya existentes y probados junto con una programación de Webmail. El mérito de Zimbra esta en su interfaz de Webmail (con soporte a Ajax) y en el empaquetado de todas las aplicaciones de terceros. Hace poco Yahoo compró a Zimbra, y si los compró por algo sería.

Todo el conjunto de aplicaciones de terceros es:

- Apache + Tomcat (servidor Web),
- Clamav (antivirus),
- Spamassessin + DSpam (filtro de spam),
- Amavis-new (conector entre los antivirus y el servidor de correo),
- Jdk (Maquina Java),
- MySQL (servidor de base datos),
- OpenLdap (servidor de directorio),
- Postfix (servidor de correo SMTP) y
- Cyrus (servidor de correo POP/IMAP).

Primero empezaremos explicando el método de instalación de la 4.5 (mientras escribo esto estoy preparando una maquina de prueba para migrar a la 5.0), la configuración estándar y terminaremos por las modificaciones hechas por mí como son:

- Instalación de un segundo antivirus,
- traducción al castellano,
- método común de actualización del motor de antivirus por defecto,
- método de copia de seguridad,
- recogida externa del correo con Fechmail,
- método renovar los certificados,
- cambio de los puertos estándar del Zimbra,
- añadir más soporte PHP a Zimbra y
- adjuntar un mensaje de confidencialidad a todos los mails salientes.

El tiempo estimado para montar este sistema de correo perfectamente funcionando es de unas dos jornadas de trabajo. Tal como esta funcionando ahora mismo, se está dando soporte a 200 cuentas de correo, más de 3000 mensajes diarios y un relay de backup externo.

Para los que realmente se decidan a instalar este sistema en producción les recomendaría ver soluciones open-source más sencillas como Scalix y sobre todo trabajar una temporada con un sistema de correo montado a “pelo”. Es decir, montarse un sistema de correo Postfix con

antivirus, antispam, webmail y servidor de base de datos. Esto te permitirá tener más experiencia en ver como funcionan estos paquetes por separado y como se enlazan unos con otros.

Antes de tener Zimbra funcionábamos con esta arquitectura, instalando los paquetes por separado: Potfix como servidor de correo SMTP, Cyrus-SASL como servidor POP/IMAP con SSL, Squirrelmail como servidor de Webmail, Postfixadmin+MySQL para almacenar los usuarios, Clamav+Amavis-new como sistema de antivirus, y Postfix VDA para dar cuotas de disco a los buzones.

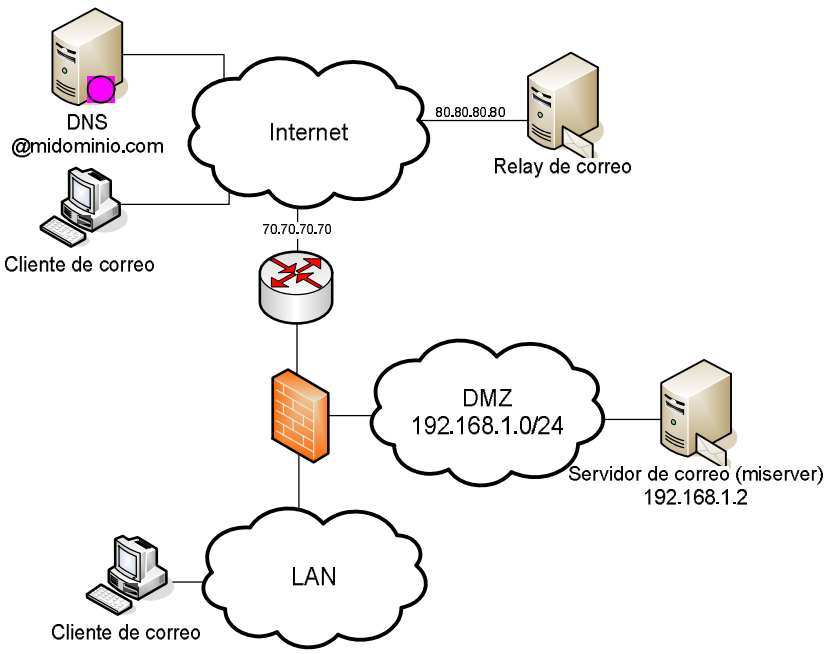
Panorama de partida

Para implementar nuestro servidor de correo utilizaremos un servidor Xeon 3Ghz, con 3Gb de RAM y dos discos de 200Gb minino en RAID 0+1 (por hardware). Esta maquina la colocaremos en una DMZ con salida a Internet garantizada 100% con simetría. Lógicamente esta configuración dependerá mucho de las necesidades de cada uno y de su presupuesto. Lo que tenemos que tener 100% garantizado es la salida a Internet. Si no hay Internet no hay correo, y puedo a segurar que un usuario puede estar 1 hora si navegar por la Web, pero no puede estar ni 10 minutos sin correo; con lo cual llaman diciendo "*parece que el correo no va, ¿no?*"

Lo de la simetría a Internet depende de cada uno. Por regla general se recibe más correo del que se envía (por culpa del Spam), pero tenemos que tener en cuenta que no solo es posible implementar el correo, sino que también podemos tener una Web o que nuestros usuarios se conecten por VPN a la oficina.

Aun así si no tenemos garantizada la conexión a Internet necesitaremos un relay de correo con otra empresa para que nos recoja el correo que nos envían hasta que nuestra conexión a Internet este levantada. Esto lo explicaremos luego.

Para terminar necesitaremos control sobre nuestro dominio (para los ejemplos utilizaremos midominio.com) y acceso a la configuración del DNS para poder modificar los registros según nuestras necesidades.



Instalación del SO

Una vez tengamos el servidor listo y montado el RAID de los disco instalaremos el sistema operativo. En este caso para la versión 4.5 de Zimbra Open-Source utilizaremos Fedora Core 4 para 32 bits (x86). Me hubiera gustado instalarlo para 64 bits (x86_64) teniendo un Xeon pero a dia de hoy la versión 5.0 de Zimbra no esta (ni estará) para 64 bits y esto me hubiera complicado.

Nota: antes de comenzar a instalar recomiendo comprobar el BIOS del Server para ver las posibilidades. Si es un servidor HP con soporte iLO podemos configurarlo para tener el control remoto del hardware si el SO cae. En este caso necesitaremos dos bocas ethernets. Una para iLO y otra para el servidor. El servidor e iLO van por interfaces de redes diferentes.

En esta fase colocaremos la maquina dentro de la DMZ. Esto no debería ser así mientras estamos montando la maquina. Aun así, si la colocasemos tampoco seria muy grave porque aun no tenemos hechos los PATs necesarios en el router y en el firewall para tener acceso desde el exterior al servidor. Si eres un administrador paranoico deberías tenerlo todo configurado y probado antes de poner la maquina en la DMZ.

La instalación del SO no la explicaremos porque no tiene más misterio que botar del DVD e instalar. Dos cosas: al seleccionar los paquetes, seleccionaremos el mínimo posible (si que seleccionaremos las 'Development Tools", las "Administration Tools" y las "System Tools", el resto de paquetes necesarios los instalara Zimbra) y configuraremos la IP con un dirección libre de la DMZ.

Nota: la instalación de servidores en producción de cualquier tipo siempre debería realizarse en ingles (salvo manías de cada uno). Sobre todo porque los primeros parches salen en inglés y a la hora de buscar información (sobre todo mensajes de errores)..

Una vez botado y comprobado que tenemos salida Web, actualizaremos todos los paquetes con un "`yum update`". Aquí nos podemos ir a tomar el primer café mientras se instalan los 300Mb de actualizaciones.

El siguiente paso es habilitar y deshabilitar con "`ntsysv`" los servicios que necesitemos. Hay que deshabilitar: bluetooth, cups, isdn, pcmcia y sendmail. Habilitaremos los servicios: iptables y sshd.

El siguiente paso es configurar el firewall de la propia maquina. No confundir con el firewall de nivel superior como pueda ser por ejemplo un PIX, un ISA o simplemente las funciones de firewall del router.
El firewall que viene con los Fedoras es iptables. La configuración del firewall se encuentra en "`/etc/sysconfig/iptables`". Listo la configuración básica del firewall.

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
:RH-Firewall-1-INPUT - [0:0]
:LOGDROP - [0:0]

# Buscar en "cat /var/log/messages | grep IPTABLES"
-A LOGDROP -j LOG --log-prefix "IPTABLES "
-A LOGDROP -j DROP
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT

# puertos especiales solo para el administrador
-A    RH-Firewall-1-INPUT    -m    state    --state    NEW    -m    tcp    -p    tcp    -s
192.168.1.3/255.255.255.255 --dport 22    -j ACCEPT
-A    RH-Firewall-1-INPUT    -m    state    --state    NEW    -m    tcp    -p    tcp    -s
192.168.1.3/255.255.255.255 --dport 7071  -j ACCEPT
# Servicios publicos
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp  --dport 995  -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp  --dport 993  -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp  --dport 110  -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp  --dport 143  -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp  --dport 25   -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp  --dport 80   -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp  --dport 8443 -j ACCEPT
# Denegamos el resto de paquetes. Lo paquetes UDP denegados no los logeamos
-A RH-Firewall-1-INPUT -m tcp -p tcp -j LOGDROP
-A RH-Firewall-1-INPUT -j DROP

COMMIT
```

Nota: para el que no este familiarizado con las reglas de control de acceso decir que las ACL se procesan de arriba abajo. Cuando una regla es cierta el paquete se deniega o acepta y ya no se continua con ninguna regla más. Por Internet hay montones de scripts para generar configuraciones para Iptables.

Esta configuración de ACLs es muy sencilla pero ya nos sirve. Lo que haremos es configurar el puerto 22 y el 7071 para que solo el administrador (o el departamento informático) tenga acceso.
Cualquier otro paquete de deniega y se logea en "`/var/log/messages`" con la etiqueta IPTABLES. Con ayuda de esta etiqueta podremos buscar rápidamente entre todo el contenido de "`/var/log/messages`".

La descripción de los puertos que hemos abierto es la siguiente:

- 22/TCP: puerto de SSH para conectarnos remotamente al servidor,
- 7071/TCP: puerto HTTPS de administración del Zimbra,
- 143/TCP 110/TCP: puerto IMAP y POP del correo,
- 995/TCP y 993/TCP: puerto IMAP seguro y POP seguro del correo,
- 80/TCP: puerto HTTP del Zimbra (luego veremos para que lo utilizamos) y
- 8443/TCP: puerto HTTPS del Zimbra (luego veremos para que lo utilizamos).

Esto es lo necesario para dejar el servidor decente antes de comenzar a instalar el Zimbra. Ahora solo queda reiniciar el servidor (manías mías) para continuar el desarrollo.

Instalación de Zimbra

Bajaremos de www.zimbra.com la versión 4.5 que es la que nos interesa para nuestra instalación. Desempaquetaremos la aplicación e intentaremos leer el documento "`zcs/doc/quick_start.pdf`" porque es allí donde encontraremos el proceso de instalación y los pre-requisitos necesarios antes de instalar. Si miramos el README.TXT encontraremos la distribución de las carpetas de instalación así como del propio instalador (install.sh).

Explico a continuación las modificaciones necesarias para cumplir los pre-requisitos:

- Deshabilitar SELinux. Para ello modificaremos "`/etc/selinux/config`" con SELINUX=disabled,
- Paramos el servicio de sendmail con "`chkconfig sendmail off`" y "`service sendmail stop`",
- Modificaremos el archivo "`/etc/hosts`" con el siguiente contenido:

```
127.0.0.1               localhost.localdomain localhost
192.168.1.2             correo.midominio.com correo
```

Ya podemos comenzar el proceso de instalación con "`./install.sh`". Lo primero que hará es buscar en el sistema si tenemos instalado alguna versión de Zimbra.

Nota: el proceso de actualización a una versión superior es muy sencilla. Al arrancar el instalador comprueba la versión que tenemos y actualiza. Lo que hay que tener en cuenta es que todas las modificaciones que nosotros hallamos hecho internamente las perdemos. Por eso nuestras modificaciones las tenemos que tener bien documentadas para restaurarlas y ajustarlas a la nueva versión rápidamente.

En el siguiente paso comprobará si nos falta algún paquete o librería. Si nos faltara alguno lo instalamos mediante un yum y volvemos a arrancar el proceso de instalación. En mi caso siempre que instalo me falta por hacer un "`yum install fetchmail`" y un "`yum install libstdc++.so.5`"

Ahora afirmamos que queremos instalar todos los paquetes de Zimbra (core, ldap, logger, mta, snmp, store, apache y spell) y comenzará las instalación de todos los RPMs que hay dentro de "`zcs/packages`".
Una vez terminada la instalación de los paquetes empieza el proceso de configuración. Desde el menú que aparece podemos cambiar muchos parámetro de Zimbra pero por ahora solo cambiaremos la contraseña del "admin" tal como nos recomienda al estar marcado con asteriscos. Luego aplicamos los cambios y guardamos la configuración tal como nos recomienda.

Tras la configuración empezará un proceso automático que creará los certificados SSL, creará la estructura LDAP, creará los usuarios por defecto del MTA y alguna cosilla más.
Luego nos preguntará si queremos notificar a Zimbra sobre nuestra instalación. Diremos que no…¿no?

Y para finalizar, arrancaran todos los servicios.

Si durante el proceso de instalación (después de instalar los RPMs) tenemos algún problema y la instalación se detiene por algún error nos tocará buscarnos la vida como siempre. Nos toca

buscar en Google y sobre todo y de gran ayuda pasearse por los foros de Zimbra en http://www.zimbra.com/forums.
Una vez resuelto el problema podemos continuar con la instalación arrancado la utilidad "`/opt/zimbra/libexec/zmsetup.pl`".
Si queremos desinstalar y volver a empezar otra vez desde cero haremos un "`./install.sh -u`"

El proceso de instalación creará el usuario zimbra ("su – zimbra") para utilizar cierto conjunto de scripts. Entre estos scripts esta "`./opt/zimbra/bin/zmcontrol stop`" y "`./zmcontrol start`" para arrancar y para todos los servicios de Zimbra.

Ya lo tenemos todo instalado. Ahora solo nos queda entrar en la administración de Zimbra (https://correo.midominio.com:7071/zimbraAdmin/) con admin@midominio.com y la contraseña que hemos puesto durante la instalación y crear las cuentas de usuario que necesitemos. También configuraremos algún cliente de Outlook para comenzar hacer pruebas.

Cuando estemos convencidos que nuestro servidor esta listo podremos abrirlo al público configurando el DNS para comenzar a recibir correo. Esto no debería ser aun así porque faltaría añadir todas las modificaciones internas que uno quiera hacer (doble antivirus, traducciones al castellano, etc).

Configuración del DNS

El siguiente paso es configurar el DNS de midomnio.com para crear los registros A y MX. Crearemos un registro A con correo.midominio.com que apunte a mi IP publica 70.70.70.70 y luego dos entradas MX. Una entrada MX con peso 10 que apunte a correo.midominio.com y otra con peso 20 que apunte al relay del backup. La forma de configurar esto depende de con quien tengamos contratado la compra de nuestro dominio. Si lo hemos contratado nosotros mismos desde un proveedor como register.com, network-solutions.com, etc, es tan fácil como entrar en su Web con nuestro login y administrar el dominio. Si lo tenemos contratado con una tercera empresa, pues le decimos lo que queremos y que nos lo hagan ellos.

Toda modificación en un DNS implicará esperar 24 o 48 horas hasta que todos los DNS se actualicen.

Si hacemos un nslookup a nuestro dominio, la cosa tendría que quedar de esta manera:

```
C:\>nslookup
Servidor predeterminado:  oculto.midominio.com
Address:  192.168.1.8:53

> server 194.179.1.100
Servidor predeterminado:  100.red-194-179-1.static.ccgg.telefonica.net
Address:  194.179.1.100

> set type=mx
> midominio.com
Servidor:  100.red-194-179-1.static.ccgg.telefonica.net
Address:  194.179.1.100

Respuesta no autoritativa:
midominio.com      MX preference = 10, mail exchanger = correo.midominio.com
midominio.com      MX preference = 20, mail exchanger = smtp.empresaderelay.net

correo.midominio.com      internet address = 70.70.70.70
smtp.empresaderelay.net  internet address = 80.80.80.80
>
```

Lo que conseguimos poniendo los pesos de 10 y 20 es que por defecto todo el correo entrante se enviará a correo.midominio.com y solo si no responde (se cae nuestra linea de datos, nuestro server cae o un ratón a mordido el ethernet) el correo se reenviara a nuestro relay de backup contratado en smtp.empresaderelay.net.

Lógicamente si nuestro correo cae y va a parar todo al relay, tendremos que recoger el correo depositado en este relay más tarde una vez nuestro sistema de correo vuelva a funcionar. El método y el momento de recoger este correo lo veremos luego.

Nota: La opción de utilizar un relay de backup de correo es opcional lógicamente, eso depende del aprecio que tenga uno a su correo. La verdad es que vale muy poco dinero y nos evita que al emisor le llegue un mail de respuesta con eso de que no ha podido entregar el correo a midominio.com. Esto es como las copias de backup, todo va bien hasta que falla y cuando falla es cuando lo hechas de menos.

Administración diaria

Antes de continuar con las modificaciones hechas en el Zimbra para ajustarse a mis necesidades haremos una pequeña parada explicando comandos básicos.

- Como ya hemos dicho antes el comando básico para parar y arrancar Zimbra es "`./zmcontrol stop`" y "`./zmcontrol start`". Si queremos detener o arrancar el antivirus, tenenos "`./zmavavisctl stop/start`" y para parar el servidor Web tenemos "`./zmapachectl stop/start`". Estos comandos son necesarios por ejemplo cuando queramos actualizar el antivirus o el certificado SSL sin necesidad de detener el servidor de correo.

- Si miramos las tareas programadas de Linux veremos que hay un montón de tareas creadas para Zimbra. Entre ellas una tarea que arranca el script "zmdailyreport". Esta nos genera un report con la cantidad de correo procesados ese dia, cuantos correo han llegado a cada persona y cuantos a enviado, los errores en el MTA, etc. Es interesante ir controlándolo.
  Si queremos un report de un mes entero, podemos modificar los parámetros startTime y endTime del script para ajuntarlo a nuestras necesidades.
  También podemos hacer "`./zmlocalconfig -e zimbra_mtareport_max_recipients=10`" y "`/zmlocalconfig -e zimbra_mtareport_max_senders=10`" para no hacer el report demasiado largo.

- Podemos utilizar el comando "`postqueue -p`" para ver la cola de correos en el MTA.
- Con "`./opt/zimbra/clamav/freshclam --config-file=/opt/zimbra/conf/freshclam.conf`" podemos forzar la actualización del antivirus.
- Se pueden aprender más cosas en el "zcs/doc/admin.pdf"

Ahora que ya tenemos nuestro servidor de correo funcionado y tenemos un conocimiento básico de la administración podemos comenzar a modificarlo y añadir funcionalidades nuevas.

Recogida externa del correo con Fetchmail

Como ya comentamos es recomendable tener contratado nuestro servicio de backup de relay por si nuestra linea de datos o servidor cae. Es hora de configurar nuestro servidor para recoger este correo. Según la política que seguiremos recogeremos el correo de la cuenta de relay cada hora. Lo primero que necesitamos saber es la dirección de correo que nos ha proporcionado la empresa de relay así como el usuario y contraseña. Independientemente de si nuestro correo cae o no cae, nosotros recogeremos el correo cada hora. Como es lógico si nuestro correo siempre va bien, el buzón del relay siempre estará vacío en tería. En la práctica no es cierto.

El archivo de configuración de Fetchmail esta en "`/root/.fectchmail`". Lo editaremos con algo parecido a esto:

```
set logfile "/var/log/fetchmail.log"
set spambounce

poll smtp.empresaderelay.com
        proto pop3
        no dns
        localdomains
        midominio.com
        user "mimominio@empresaderelay.com"
        pass "micontraseña"
        is *
        fetchall
```

Lo que estamos diciendo en este archivo de configuración es que tiene que recoger todo el correo (fecthall) de la cuenta midominio@empresarelay.com que esta en el servidor smtp.empresarelay.com.

Ahora solo queda arrancar Fetchmail. Tenemos dos soluciones, crear un servicio para Linux o crear una tarea para ejecutar Fetchmail cada hora. Si queremos crear una tarea programaríamos la tarea haciendo un llamada a "fechmail –F" y si queremos crear un servicio para Linux podríamos utilizar el siguiente script:

```
#!/bin/sh

case "$1" in
'start')
   echo -n "Starting fetchmail..."
   fetchmail -F -d 3600
   touch /var/lock/subsys/fetchmail
   echo
   ;;
'stop')
   echo -n "Shutting down fetchmail..."
   fetchmail -q
   rm -f /var/lock/subsys/fetchmail
   echo
   ;;
*)
   echo "Usage: $0 { start | stop }"
   ;;
esac
exit 0
```

En este ejemplo cuando hacemos "`service fetchmail start`" arrancamos el daemon de fetchmail haciendo polling cada hora. Si tenemos problemas con fetchmail podemos utilizar el parámetro –v para arrancarlo en modo debug y mirar dentro de "`/var/log/fetchmail`" para ver cual es el problema.

Actualización del motor del antivirus

Uno de los defectos que tiene Zimbra es que no es posible actualizar el motor del antivirus automáticamente. Hay que bajarse la nueva versión del ClamAv instalarla y volver a enlazarla con Zimbra. Nos daremos cuenta porque al actualizar el antivirus nos saldrá un warning del estilo "Your ClamAV installation is OUTDATED!".
Siempre es importante tener la última versión del motor para asegurarnos que pillamos el mayor número de virus posibles.

Bajaremos de la Web de ClamAv (http://www.clamav.org) la última versión. En nuestro caso la versión es la 0.91.2. Copiaremos el paquete a nuestra carpeta de instalaciones y lo desempaquetaremos. El proceso de actualización esta documentado en el foro de Zimbra (http://www.zimbra.com/forums/administrators/1015-solved-howto-update-clamav.html) pero explicare los pasos básicos.

```
# tar -xzvf clamav-0.91.2.tar.gz
# cd clamav-0.91.2
#    ./configure    --prefix=/opt/zimbra/clamav-0.91.2    --with-user=zimbra    --with-
group=zimbra
# make
# make check
# make install
```

Como siempre que compilamos comprobamos que configure y make no da ningún error. Si lo diera, pues a buscarse la vida tirando de Google y del foro de ClamAv. Una vez instalado aparece la capeta "`/opt/zimbra/clamav-0.91.2`". Ahora solo queda enlazar este nuevo antivirus con Zimbra.

```
# ./zmamavisdctl stop
# rm /opt/zimbra/clamav
# ln -s /opt/zimbra/clamav-0.91.2 /opt/zimbra/clamav
# mkdir /opt/zimbra/clamav/db
# chown -R zimbra:zimbra /opt/zimbra/clamav-0.91.2
# zmamavisdctl start
# ./opt/zimbra/clamav/bin/freshclam --config-file=/opt/zimbra/conf/freshclam.conf
```

Lo que hacemos es enlazar el nuevo antivirus con Zimbra, dar los permisos correctos, arrancar de nuevo el servicio de antivirus y forzar la actualización de las bases de virus.
Para asegurarnos que lo tenemos todo correcto, tenemos que hacer dos pruebas. La primera es ver que Amavis (recordar que es el paquete que enlaza los antivirus con Postfix) a encontrado el antivirus. Para ellos editamos "`/var/log/zimbra.log`" y miramos que parece algo como:

```
Jan 21 12:23:40 correo amavis[20675]: Using internal av scanner code for (primary)
ClamAV-clamd
```

La segunda prueba es coger algunos de los virus de nuestra colección he intentar enviarlos por Zimbra y ver que los detecta correctamente. Quien no tenga que pruebe un Eicar (http://www.eicar.org/anti_virus_test_file.htm).

Nota: Cuando se pasa de una versión a otra de ClamAv es posible que el archivo de configuración clamav.conf traiga algún cambio. Ese es el caso de pasar de la 0.88.7 a la 0.91.2. Si miramos /opt/zimbra/log/clamav.log veremos si el antivirus arranca bien o no. Si hay algún problema en el archivo de configuración encontraremos el error en ese log. Para resolverlo modificaremos la configuración de /opt/zimbra/conf/clamd.conf.in

Instalación de un segundo antivirus

En los tiempos que corren seria de valientes tener un servidor de correo con solo un paso de antivirus. Por regla general el 99% de los virus que entran en una empresa nos entran por correo. Así que instalaremos un segundo antivirus, los virus que no detecte ClamAv, esperamos que si los detecte el segundo antivirus.

Hay que instalar una versión de antivirus compatible con Amavis (en su pagina Web está la lista de antivirus compatibles). En nuestro caso instalaremos una versión gratuita de BitDefender 7.1-3. No es que ClamAv y BitDefender me apasionen (prefiero un McAfee o un Kaspersky) pero es lo único que he encontrado gratuito y que me funcione. ¿Alguien utiliza otros?
En dos años que llevamos con Zimbra o Postfix con estos dos antivirus solo han habido tres correos electrónicos con virus que no fueron detectados. El virus lo para el antivirus del usuario.

```
# rpm -i BitDefender-Console-Antivirus-7.1-3.linux-gcc3x.i386.rpm
# bdc --update
```

Una vez instalado y actualizado tenemos que crear una tarea en Linux para actualizar el antivirus por lo menos cada dos horas. Haríamos una llamada a "`bdc -update | mail -s 'Actualizacion Bitdfender' admin@midominio.com`", de forma que el resultado de la actualización se envíe por correo y pueda ir viendo que las actualizaciones se hacen correctamente.

El siguiente paso es reiniciar Amavis y ver si en "`/var/log/zimbra.log`" aparece algo como:

```
Jan 21 13:35:53 correo amavis[12354]: Found primary av scanner BitDefender at
/usr/bin/bdc
```

Ya tenemos segundo antivirus.

Copias de seguridad

Otro defecto que trae la versión libre de Zimbra es que no trae ningún proceso de copia de seguridad, ni en forma de script ni documentado. Así la única solución es parar el servicio y hacer un paquete de todo el contenido de /opt/zimbra. Podríamos hacer un script para realizar todo esto, pero ya existe un script en Sourceforge hecho para tal propósito llamado zimbraColdBackup.pl (http://sourceforge.net/projects/zcstools/).

Se trata de un script hecho el Perl y simplemente hay que ejecutarlo con "`./zimbraColdBackup.pl –confirm`". El script parará Zimbra y hará una copia en /backup.

Nota: dado que es un programa en Perl posiblemente nos de errores porque nos falta instalar algún modulo de Perl. Por ejemplo a mi me falta instalar el módulo `Proc::ProcessTable`, para instalarlo haremos:

```
#perl –MCPAN –e shell
Cpan> install Proc::ProcessTable
```

Para hacer un copia de seguridad diaria, basta con programar una tarea diaria. Durante el proceso de la copia de seguridad el sistema de correo no funcionará, pero podremos seguir recibiendo correo gracias al backup del relay. Otra consideración será la hora y el dia en el que realicemos la copia. Deberemos tener en cuenta la cantidad de gente que se puedan conectar tanto en horas de oficina como el posible grupo de empleados que se pueden conectar desde Internet (vía Webmail) a cualquier hora del dia.

Mensaje de advertencia en todos los correos salientes

Para mi es la parte más entretenida porque no depende de la instalación de Zimbra que tengamos sino que hay que meter mano a la configuración de Postfix.
Lo que haremos para colocar un mensaje (disclaimer) en todos los mails de salida es hacer un bypass en el flujo de Postfix. Crearemos un script el cual sacará el mail de Postfix, le introducirá un pie de pagina de disclaimer y luego lo volverá a introducir dentro de Postfix para que siga su camino.

Recordar que los disclaimers son esos textos que se ponen al final del mail diciendo "Este mensaje es confidencial…." y que últimamente las grandes empresas lo tienen. Y como nuestra empresa es grande pues nosotros también se lo pondremos.

Los pasos que seguiremos son:

- Hacer un "`yum install altermime`" para instalar un paquete que es capaz de modificar el cuerpo MIME de un correo electrónico.
- Crear el directorio "`/opt/zimbra/postfix/disclaimer`" con permisos 770 y propietario root:zimbra.
- Crear dentro de este último directorio tres archivos de texto: add_disclaimer.sh, disclaimer_txt.txt y disclaimers_html.txt. Luego comentaremos el contenido de estos archivos.
- Crear el directorio "`/var/spool/altermime`" con permisos 777.

Ahora tenemos que cambiar el archivo de configuración de Postfix que define todos los procesos que componen el MTA. Original mente la cabecera del archivo /opt/zimbra/postfix/conf/master.cf empieza por:

```
# Postfix master process configuration file.  For details on the format
# of the file, see the Postfix master(5) manual page.
#
# ==========================================================================
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)   (yes)   (yes)   (never) (100)
# ==========================================================================
smtp      inet  n       -       n       -       -       smtpd
```

Y la tenemos que cambiar por esta otra:

```
#
# Postfix master process configuration file.  For details on the format
# of the file, see the Postfix master(5) manual page.
#
# ==========================================================================
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)   (yes)   (yes)   (never) (100)
# ==========================================================================
smtp      inet  n       -       n       -       -       smtpd
       -o content_filter=disclaimer

disclaimer  unix  -       n       n       -       -       pipe
    flags=Rq    user=zimbra  argv=/opt/zimbra/postfix/disclaimer/add_disclaimer.sh  -f
${sender} -- ${recipient}
```

El contenido de add_disclaimer.sh es:

```
#!/bin/sh

#System dependent settings
ALTERMIME=/usr/bin/altermime
ALTERMIME_DIR=/var/spool/altermime
SENDMAIL="/opt/zimbra/postfix/sbin/sendmail -G -i"
MIDOMINIO=@midominio.com

TEMPFAIL=75
UNAVAILABLE=69

cd $ALTERMIME_DIR || { echo $ALTERMIME_DIR does not exist; exit $TEMPFAIL; }

trap "rm -f in.$$" 0 1 2 3 15

cat >in.$$

case "$2" in
   *$MIDOMINIO*)
      $ALTERMIME --input=in.$$ \
      --disclaimer=/opt/zimbra/postfix/disclaimer/disclaimer_txt.txt \
      --disclaimer-html=/opt/zimbra/postfix/disclaimer/disclaimer_html.txt \
      --xheader="X-Copyrighted-Material: MI EMPRESA S.L." || \
      { echo Message content rejected; exit $UNAVAILABLE; }

esac

$SENDMAIL "$@" <in.$$

exit $?
```

Lo que estamos definiendo aquí es un nuevo proceso llamado disclaimers que será implementado por el script add_disclaimer.sh.

Dentro de disclaimet_txt.txt colocaremos el texto que queremos que aparezca cuando el mail sea en formato texto y dentro de disclaimers_html.txt lo mismo para los mails que se envíen en formato HTML.

Si no funciona a la primera es normal. Normalmente es problema de permisos. Si consultáis "`/var/log/zimbra.log`" encontrareis cosas como:

```
Jan 21 16:59:38 server pipe[28439]: fatal: pipe_comand: execvp
/opt/zimbra/postfix/disclaimer/add_disclaimer.sh: Permission denied
```

Traducción al castellano

Con la nueva versión 5.0 de Zimbra la traducción al castellano es instantánea porque ya viene instalada con la versión, pero para la version 4.5 no es así y habrá que bajar las traducciones de Internet, descomprimirlas y decirle a Zimbra que las utilice.

Para encontrar las traducciones en castellano para la versión 4.5 hay que tirar del foro de Zimbra. Existe un post donde esta colgada estas traducciones: http://www.zimbra.com/forums/i18n-l10n-translations/6379-spanish-translation-zimbra-4-5-ge.html

Los pasos a seguir son los siguientes:

- Descomprimir el Zip y copiar los cuatro archivos de texto en "`/opt/zimbra/tomcat/webapps/zimbra/WEB-INF/classes/msgs/`",
- Una vez copiados dar a estos 4 archivos permisos 664 y propietario zimbra:zimbra al igual que el resto de archivos,
- Reiniciar Zimbra.

Una vez entremos en el Webmail veremos que ya sale en castellano. Estas traducciones solo nos servirán para el Webmail y no para la consola de administración Web.
El único problema que nos podemos encontrar en este punto es que tengamos que borrar la cache de nuestro Internet Explorer porque crea conflicto con las páginas cacheadas que teníamos antes en inglés.

Cambio de puertos

Dado que tenemos un servidor conectado a Internet y por tanto un servidor Web como Apache, vamos a sacarle provecho y poder colgar nuestras paginas PHP que necesitemos.

Si hacemos un "`./zmprov gs correo.midominio.com`" podemos ver la distribución de los puertos (zimbraAdminPort=7071, zimbraMailPort=80, zimbraMailSSLPort=443).
Los puertos 80 y 443 no corresponden a Apache como seria lo lógico, sino a Tomcat. Por defecto Apache esta corriendo en el puerto 7780 (probadlo!).

Lo que haremos es reasignar lo puertos para dejar el 80 y el 443 libres para Apache. Dejaremos zimbraAdminPort al 7071, el zimbraMailPort lo cambiaremos por el 8081 y el zimbraMailSSLPort lo cambiaremos por el 8443.

Para ello haremos lo siguiente:

```
#./zmprov ms correo.midominio.com zimbraMailPort 8081 zimbraMailSSLPort 8443
#./zmprov ms correo.midominio.com zimbraMailMode mixed
#./zmprov ms correo.midominio.com zimbraSpellCheckURL
http://webmail.midominio/aspell.php
#./zmprov gs correo.midominio.com
```

Ahora editamos "`/opt/zimbra/conf/httpd.conf`" para cambiar el valor del parámetro Listen al 80. Este archivo de configuración de Apache no tiene permisos de escritura, así que habrá que dárselo y luego dejarlo como estaba. También podemos modificar el parámetro DirectoryIndex con "`DirectoryIndex index.php index.html`" para cargar paginas PHP como veremos más tarde.

Solo queda reiniciar Zimbra. Si arrancamos veremos que se produce un error. No es posible arrancar con el usuario zimbra el Apache y por esos se produce un conflicto en el puerto 80. Más concretamente se produce cuado "/opt`/zimbra/bin/zmspellctl`" intenta arrancar el "`/opt/zimbra/bin/zmapachectl`". Para solucionarlo tendremos que arrancar el Apache a mano con "`/opt/zimbra/bin/zmapachetl`" como root. ¿Alguien tiene una solución más elegante?

Ahora ya tenemos libre el puerto 80 para colgar lo que queramos. Como política de empresa todo el mundo que se conecte al Webmail deberá ser redirigido por pagina segura, es decir quien entre por http://webmail.miempresa.com será redirigido a https://wemail.miempresa.com:8443.
Por tanto necesitamos una pagina index.php en "`/opt/zimbra/httpd/htdocs`" que redireccione. El contenido de index.php será algo como:

```
<?php
   header("Location: https://correo.midominio.com:8443/");
?>
```

Soporte PHP a Zimbra

Ahora que ya tenemos Apache a nuestro gusto es hora de probar alguna pagina PHP. Directamente probamos una pagina test.php con una llamada a la funciona phpinfo() para que nos enseñe la configuración de PHP. Rápidamente vemos que el paquete que Zimbra nos ha dado de PHP trae pocas funciones. Por ejemplo no trae soporte para MySQL, no trae soporte para Ldap, etc (yo sin MySQL no soy nadie).

Apache carga el soporte a PHP como modulo dinámico ("`LoadModule php5_module modules/libphp5.so`"), así que lo que tendremos que hacer es cambiar libphp5.so con otro modulo de PHP que soporte MySQL, LDap y todo lo que necesitemos.

Por tanto haremos lo siguiente:

```
#yum install php-mysql php-ldap php-pear php-gd
#rpm -q -a | grep php
#updatedb
#locate libphp5.so
#cp /usr/lib/httpd/modules/libphp5.so /opt/zimbra/httpd/modules/
```

Al finalizar el locate vemos que el nuevo libphp5.so esta en "`/usr/lib/httpd/modules/`". Por tanto solo tenemos que copiar esta librería en /opt/zimbra/httpd/modules. Una vez copiado el modulo solo queda reiniciar el Apache con el zmapachectl.

Si probamos ahora nuestra pagina test.php veremos como ahora si que tenemos soporte para MySql, Ldap y más cositas.

Renovar el certificado

Zimbra por defecto nos genera un certificado SSL para dos años. En la versión 3.0 creo que solo era para una año. Por tanto pasado este tiempo tendremos que generar nosotros mismos un nuevo certificado.

La configuración de los certificados de seguridad esta dentro de "`/opt/zimbra/conf/zmssl.conf.in`". Lo editaremos y cambiaremos "`default_days=1825`" y "`0.organizationName_default=MI EMPRESA SL`".

> *Nota: existe bug documentado por Zimbra (http://bugzilla.zimbra.com/show_bug.cgi?id=12228) por el cual el parámetro default_days es ignorado. La solución es modificar a pelo el zmcreatecert y el zmcreateca. Donde pone 365 lo cambiaremos por ejemplo por 1825.*

Para regenerar los certificados haremos:

```
# cd /tmp
# tar -cf /tmp/zimbra-ssl.tar /opt/zimbra/ssl/
# rm -rf /opt/zimbra/ssl
# mkdir /opt/zimbra/ssl
# chown zimbra:zimbra /opt/zimbra/ssl
# chown zimbra:zimbra /opt/zimbra/java/jre/lib/security/cacerts
# chmod 644 /opt/zimbra/java/jre/lib/security/cacerts
# su - zimbra
# keytool -delete -alias my_ca -keystore /opt/zimbra/java/jre/lib/security/cacerts -#
storepass changeit
# keytool -delete -alias tomcat -keystore /opt/zimbra/tomcat/conf/keystore -storepass
# zimbra
# zmcreateca
# zmcreatecert
# zmcertinstall mailbox /opt/zimbra/ssl/ssl/server/tomcat.crt
# zmcertinstall mta /opt/zimbra/ssl/ssl/server/server.crt
/opt/zimbra/ssl/ssl/server/server.key
```

Toda la documentación sobre este proceso está dentro de la Wiki de Zimbra en http://wiki.zimbra.com/index.php?title=SSL_Certificate_Problems.

Una vez hecho todo el proceso y sin errores basta con reiniciar Zimbra y cargar el Webmail para ver si el certificado esta bien creado.

Activar grupos de distribución automáticamente

La mayor proporción de correo que recibo es por culpa de los grupos de distribución. Es decir si algún spammer consigue la dirección valida administracion@miempresa.com será capaz de enviar un mail a todos los usuarios de administración. Por tanto si tuviéramos una dirección del estilo todolaempresa@midominio.com la cosa seria peligrosa. Por tanto lo ideal seria que este grupo de distribución estuviera cerrado y solo activarlo durante 15 minutos para poder enviar

algún tipo de comunicado y luego cerrarlo automáticamente. Lo que haremos es una pagina Web PHP que llame a un script para que abra este grupo y luego lo cierre.

Primero crearemos el script "`/opt/zimbra/bin/zmactivedl`" con el siguiente contenido:

```
#!/bin/bash

echo "Activando el grupo todalaempresa@midominio.com..."
/opt/zimbra/bin/zmprov mdl todalaempresa@midominio.com zimbraMailStatus 'enabled'
sleep 900
echo "Desactivando el grupo todalaempresa@midominio.com..."
/opt/zimbra/bin/zmprov mdl todalaempresa@midominio.com zimbraMailStatus 'disabled'

exit
```

Ahora crearemos una pagina HTML que copiaremos dentro "`/opt/zimbra/httpd/htdocs`". El contenido seria algo como:

```
<html>
<body>
El grupo de distribucion todalaempresa@midominio.com estara activado durante 15
minutos.<br>
Puede cerrar esta ventana.
<script src=http://correo.midominio.com/zmactivedl.php></script>
</body>
</html>
```

El contenido del zmactivedl.php sería:

```
<?php
   shell_exec('/opt/zimbra/bin/zmactivedl &');
?>
```