ARCHIVING EMAILS - POSTFIX

Many countries in the world currently have some form of cyberlaw. In many of these countries, email messages are regarded as proper legal documents which have legal weight and meaning. For this reason and others, many companies now found themselves in the situation where they have to archive all their emails. Now leaving aside the problem of storage space which can solved by throwing disks at the issue, the one problem people seem to struggle with, is how to actually get the emails into some sort of archive. Well, Postfix has the easiest method of getting this done..

What is Needed

A linux box, oodles of space and a version of Postfix 2 or higher. The glorious thing about postfix is that even if you are a Microsoft house, you can setup postfix as an email-relay and thus still archive your mails. We will not be covering how to get postfix running here, but rather the archiving bit. So lets assume you already have a working postfix setup, lets edit your *main.cf* file, and add the following at the bottom..

```
always_bcc = <users-email-address>
```

Now you can use an email address on a different server in which case your line may look like..

```
always_bcc = bob@greengrocer.co.za
```

Or if you are delivering it to user homed on the postfix server (our preferred method), your line will look something like this..

```
always_bcc = bob
```

Now after you have restarted postfix, every single email your server sends or receives will be blind carbon copied to "*bob*". Think of "*bob*" as your archive. But you will also not want to use the default *mailbox* method of storing emails (where are emails reside in one big file) as this makes working with specific emails impossible. You rather want to use the *maildir* method (where every email is stored as a separate file). So in your *aliases* file, make "*bob*'s" entry look like..

```
bob:   /var/archive/maildir/
```

Now under that mail folder, you should have a folder structure like this..

```
# pwd
/var/archive/maildir
# find ./ -type d
./
./tmp
./cur
./new
```

All new emails are delivered to the "*new*" subfolder. Told you it was easy.

Managing the archive

Ok, now you have your archive, each email is a separate file, how do you make the storage more meaningful? I use these two script files, please feel free to use them. As always, use of them at your own risk and I do not consider them works of art. Tweak and fiddle as much as you want. Firstly I have the *filemail* script (change the variables to match your environment)..

```
STORE=/var/archive/maildir/new
KEEP=/var/archive/keep
```

```
DOMAIN=greengrocer.co.za

for x in `find $STORE -type f`
 do
  echo "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"
  RSLT=`cat $x | grep "Return-Path"`
  PERSONTMP=`echo $RSLT | cut -f 2 -d "<"`
  PERSON=`echo $PERSONTMP | cut -f 1 -d ">"`
  echo "-checking email.. $x"
  echo "-get sender.. $PERSON"
  RSLT1=`echo $PERSON | grep -i -e "@$DOMAIN" 1> /dev/null ; echo $?`
  if [ "$RSLT1" == "0" ]
   then
    NAME=`echo $PERSON | cut -f 1 -d "@"`
    echo "-sender is a $DOMAIN person.. $NAME"
    if [ -d $KEEP/$NAME ]
         then
          echo "--archive folder exists"
         else
          echo "--archive folder does not exist .. so create it"
          mkdir $KEEP/$NAME
        fi
   echo "-so lets move the email.."
   mv -uv $x $KEEP/$NAME
   else
    echo "-sender is not a $DOMAIN person.."
        NAME=`echo $PERSON | cut -f 2 -d "@"`
        echo "-external senders domain is $NAME"
        if [ -d $KEEP/external.domains ]
     then
       echo "--archive folder exists"
     else
       echo "--archive folder does not exist .. so create it"
       mkdir $KEEP/external.domains
    fi
        echo "-so lets move the email.."
    mv -uv $x $KEEP/external.domains
  fi
  echo "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"
 done
```

This script will parse through the *new* subfolder and extract the sender from each email (we use *reciepient* as there can be multiple *recipients* but only one *sender*), if it comes from our domain it creates a folder in a specified area with the sender's name and then copies the email to that folder. If the sender is an external person, it copies it to the *external.domains* folder. Ok, now you have the emails sorted of senders, but can we do better? Lets sort by date. This is the *sortemail* script (again, update variables as needed)..

```
BSE=/var/archive/keep
TMPF=/tmp/sort.f.lst
TMPL=/tmp/sort.l.lst

rm -rf $TMPF
rm -rf $TMPL

echo "-Get folder names.."
for FLIST in `ls $BSE/`
 do
  echo -n "."
  echo $FLIST >> $TMPF
 done
echo "" ; echo ""
```

```
for x in `cat $TMPF`
 do
  rm -rf $TMPL
  echo "-lets do $x"
  cd $BSE/$x
  find ./ -type f -maxdepth 1 > $TMPL

  for y in `cat $TMPL`
   do
    echo $y
        FILEDATE=`date -r $y | gawk '{ print $1"-"$2$3"-"$6 }'`

    if [ -d $BSE/$x/$FILEDATE ]
         then
          echo "--folder exists"
         else
          echo "--folder does not exist..create it"
          mkdir $BSE/$x/$FILEDATE
        fi
        mv $BSE/$x/$y $BSE/$x/$FILEDATE
   done
  echo ""
 done
echo ""
```

This script will parse through all the sender folders, and sort the email files into subfolders named after the day the email was sent. So after running both scripts (which you can schedule via *crontab* to run nightly), you should have a structure like this..

```
# find ./lucy -type d
./lucy
./lucy/Tue-Jul18-2006
./lucy/Wed-Jul26-2006
./lucy/Thu-Jul27-2006
./lucy/Wed-Jul19-2006
./lucy/Mon-Jul17-2006
./lucy/Thu-Jul20-2006
./lucy/Fri-Jul21-2006
./lucy/Mon-Jul24-2006
./lucy/Tue-Jul25-2006
./lucy/Fri-Jul14-2006
./lucy/Fri-Jul28-2006
./lucy/Mon-Jul31-2006
./lucy/Wed-Aug2-2006
./lucy/Tue-Aug1-2006
```

Much better. Now you can search a lot easier.

Final Words
Obviously this paper does not take into account setting up postfix or locking down the postfix server so as to properly secure your email archive, but I hope it does show that the task of archiving your emails is not beyond your reach, and can be accomplished fairly easily. As always, have fun and learn.