HOWTO - amavisd-new + clam + Kolab

History

Version 1.0 Stephan Buys (s.buys@codefusion.co.za) 22 July 2003

Version 1.1 Stephan Buys (s.buys@codefusion.co.za) 04 August 2003

- Remove several references to sophos

Introduction

This document outlines how to integrate Anti-Spam and Anti-Virus measures into a Kolab server. The components used to accomplish this is amavisd-new which integrates well into postfix, using Spam-assasin and Clam anti-virus.

Software

amavisd-new

http://www.ijs.si/software/amavisd/

clam anti-virus

http://clamav.elektrapro.com

opa

Opa is a script which eases the adminitration of any Openpkg machine. It sets the environment in such a manner that your user will access the “Kolab” tools. Installing software, modifying permissions, etc. it is easier (and sometime necesarry) to use the root user.

You can download opa somewhere at www.openpkg.com or use the code listing below:

```
#OpenPKG Add Environment
opa () {
root="$1"
if [ ! -d $root ]; then
  echo "opa:ERROR: OpenPKG root directory $root not found"
  return 1
fi
if [ ! -f "$root/etc/rc" ]; then
  echo "opa:ERROR: does not contain OpenPKG hierarchy"
  return 1
fi
eval `$root/etc/rc --eval all env`
}
```

To activate opa:

```
#opa /kolab
```

This will set up your environment to be in Kolab.

The output of the *which rpm* command should now look as follows:

```
# which rpm
/kolab/bin/rpm
```

Important. You need to assure that you are in the Kolab environment when you do the steps in this howto. I am going to assume that you are.

Alternative ways to get into the Kolab environment as root:

```
(from the root user)
#su – kolab
(will drop you into the Kolab user)
>su
(you will be prompted to give the root password)
```

Install amavisd-new

Recommended: Read the INSTALL and AAAREADME.first files that is included in the amavisd-new distribution. This will give you insight into why you are doing the steps I list.

Download and Install prerequisites

Download the necesarry tarballs and patches from the amavisd-new website and extract them. This example assumes you are in the same folder as the archive.

```
#tar -xzf amavisd-new-<version>.tar.gz
#cd amavisd-new-<version>
```

Patch the software if necesarry:

```
#patch < amavisd-new-<version>-1.patch
```

Check that you have the necesarry software installed:

File 3.41 or later is recommended for security reasons.

```
#file -v
file-4.02
magic file from /usr/share/misc/file/magic
```

Here is a little script file that will attempt to include all the perl modules necesarry for amavisd-new to work. If it reports any errors you need make sure that the correct software is installed.

`checkmodules.pl`

```
#!/kolab/bin/perl -w
use strict;

use Archive::Tar;
use Archive::Zip;
use Compress::Zlib;
use Convert::TNEF;
use Convert::UUlib;
use MIME::Base64;
use MIME::Parser;
use Mail::Internet;
use Net::Server;
use Net::SMTP;
use Digest::MD5;
use IO::Stringy;
use Time::HiRes;
use Unix::Syslog;
use Mail::SpamAssassin;
```

Note that I used the correct version of perl (the Kolab one) in the first line of the code-listing. On my test-server I received the error:

```
#perl checkmodules.pl
Can't locate Compress/Zlib.pm in @INC ...
```

This means that I need to install the Compress::Zlib library for Perl.

Important: It is essential that you install the libraries into the correct perl folders, ie. the Kolab perl folders.

First, confirm that you are using perl in the Kolab tree.

```
#which perl
/kolab/bin/perl
```

Important: If the above command does not tell you give same output as I have listed above you are not within the Kolab environment. Do not attempt to do any further steps, please read the section called **opa** above.

An easy and convenient way to install perl modules is through the use of the CPAN module. To activate the CPAN module type the following:

```
#perl -MCPAN -e shell
```

If this the first time that you run the CPAN module you will be prompted to configure your system. Just follow the prompts until you see the prompt:

```
cpan>
```

Now install the modules that you need for amavisd-new to work. In my case I need the Compress::Zlib module, so I execute:

```
cpan>install Compress::Zlib
...
```

To test that it is installed correctly verify the output of:

```
cpan>test Compress::Zlib
...
  /kolab/bin/make test – OK
cpan>quit
```

At this stage rerun the checkmodules.pl script to see if it loads correctly.

```
 #perl checkmodules.pl
Can't locate Mail/SpamAssassin.pm in @INC...
```

I now repeat the above procedure to install Maill::Spamassassin

```
#perl -MCPAN -e shell
cpan>install Mail::SpamAssassin
cpan>test Mail::SpamAssassin
#perl checkmodules.pl
#
```

If the checkmodules.pl script just returns it means that it successfully loaded all the required modules.

To be able to scan compressed attachments you need to install the following programs:

compress, gzip, bzip2, nomarch (or arc), lha, arj (or unarj), rar (unrar),

zoo, cpio, lzop, freeze (or unfreeze or melt).

As long as they are reachable by your script during runtime it does not matter if the program is provided by your distribution or by the Kolab OpenPKG packages.

Setup amavisd-new

We will use the kolab-r (restricted Kolab) user and group to run our daemons in.

First we create the amavisd-new home directory and set appropriate permissions:

```
#mkdir /kolab/var/amavis
#chown kolab:kolab-r /kolab/var/amavis
#chmod 770 /kolab/var/amavis
```

Now we copy the executable into the Kolab tree and also set correct permissions

```
(change directory to where you extracted amavis-new)
#cp amavisd /kolab/local/sbin/
#chown kolab:kolab-r /kolab/local/sbin/amavisd
#chmod 775  /kolab/local/sbin/amavisd
```

Fix the amavisd script to use the correct version of perl. With your text editor edit:

`/kolab/local/sbin/amavisd`

```
#!/kolab/bin/perl -T

#------------------------------------------------------------
# This is amavisd-new.
....
```

Create a folder for the amavisd configuration

```
#mkdir /kolab/etc/amavisd
#chown kolab:kolab-r /kolab/etc/amavisd
#chmod 775 /kolab/etc/amavisd
```

Copy the amavisd.conf file to the configuration folder:

```
(change directory to where you extracted amavis-new)
#cp amavisd.conf /kolab/etc/amavisd/
#chown kolab:kolab-r /kolab/etc/amavisd/amavisd.conf
```

Optional. Use steps similar to the above to create a folder into which you want to quarantine virus mails. This did not fit our requirements, we just delete virus mails.

Configure the amavisd.conf file as per your requirements.

From the INSTALL quide, the important parts are:

```
edit file /etc/amavisd.conf and adjust variables $daemon_group
and $daemon_user to match the chosen group and user name,
adjust variables $MYHOME, $TEMPBASE and $QUARANTINEDIR to match
the directories just created, then check/adjust other
variables,
especially those in 'Section I', including $mydomain.
```

Important. For our Kolab server we set the following:

`/kolab/etc/amavisd/amavisd.conf`

```
$MYHOME = '/kolab/var/amavis';
$daemon_user  = 'kolab-r';
$daemon_group = 'kolab-r';
$QUARANTINEDIR = undef;         #delete virus/spam mails

(Uncomment the clam section, update the socket file)
### http://clamav.elektrapro.com/
 ['Clam Antivirus-clamd',
   \&ask_daemon, ["CONTSCAN {}\n", '/kolab/var/clam/clamd'],
```

To be clear, make sure you set the above mentioned variables if you want to use this install guide, also uncomment the clam section and set the ask_daemon variable. There are other variables you need to set, this I leave up to you own judgement.

Create the SpamAssassin user_prefs file:

```
#mkdir /kolab/var/amavis/.spamassassin
#chown kolab:kolab-r /kolab/var/amavis/.spamassassin
#chmod 770 /kolab/var/amavis/.spamassassin
#touch /kolab/var/amavis/.spamassassin/user_prefs
#chown kolab:kolab-r /kolab/var/amavis/.spamassassin/user_prefs
#chmod 664 /kolab/var/amavis/.spamassassin/user_prefs
```

Before we install clam, we will test that amavisd initializes:

```
#su - kolab-r -c '/kolab/local/sbin/amavisd -c \ /
kolab/etc/amavisd/amavisd.conf debug'
```

This will execute the daemon in debug/verbose mode using the kolab-r user. Check the output carefully for any errors. You can terminate the daemon with ctrl-c.

Install ClamAnti-Virus

Extract the clam distribution onto your local machine.

From the clam distribution folder issue the following to install Clam into Kolab:

```
(from clam folder)
#./configure --prefix=/kolab/local/ --with-user=kolab-r --with-
group=kolab-r –-with-dbdir=/kolab/var/clam
#make
#make check
#make install
```

We will assure that clam uses our kolab-r restricted user and group. We will save the virus database files into /kolab/var/clam

Configure clamd

Now we need to create a clamd configuration file.

There is a template file in the Appendix called clamav.conf. This HOWTO will use the template file as a basis.

```
#mkdir /kolab/etc/clamd
#chown kolab:kolab-r /kolab/etc/clamd
```

Copy the clamav.conf file into the configuration directory.

```
(obtain clamav.conf from the appendix or download files)
#cp clamav.conf /kolab/etc/clamd/
#chown kolab:kolab-r /kolab/etc/clamd/clamav.conf
```

Test the clam daemon. Monitor the syslog for any errors

```
#su - kolab-r -c '/kolab/local/sbin/clamd \
-c /kolab/etc/clamd/clamav.conf'
```

There should be no errors reported in syslog

You can stop clamd with:

```
#killall clamd
```

Configure amavisd-new

So far we have installed all the components that we will be using, now we will begin with the final configuration for production use.

Start clamd.

```
#su - kolab-r -c '/kolab/local/sbin/clamd \
-c /kolab/etc/clamd/clamav.conf'
```

Start the amvisd-new daemon.

```
#su - kolab-r -c '/kolab/local/sbin/amavisd -c \
/kolab/etc/amavisd/amavisd.conf'
```

This time we did not specify debug, so amavisd will load and fork into the background.

To test that the daemon is running as expected issue the following:

```
#telnet 127.0.0.1 10024
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
220 [127.0.0.1] ESMTP amavisd-new service ready
#quit
221 2.0.0 [127.0.0.1] (amavisd) closing transmission channel
Connection closed by foreign host.
```

Kolab Postfix master.cf Configuration

Kolab uses template files to manage changes that occur in LDAP. It is critical that you update the correct template files as any change in the LDAP directory will override the main configuration files with their templates.

Important: To be clear about it: Dont change the /kolab/etc/postfix/master.cf file. It will be overwritten by /kolab/etc/kolab/master.cf.template.

We add the following lines to master.cf.template.

`/kolab/etc/kolab/master.cf.template`

```
smtp-amavis unix -      -       n       -       2       smtp
    -o smtp_data_done_timeout=1200
    -o disable_dns_lookups=yes
127.0.0.1:10025 inet n  -       n     -       -  smtpd
    -o content_filter=
    -o local_recipient_maps=
    -o relay_recipient_maps=
    -o smtpd_restriction_classes=
    -o smtpd_client_restrictions=
    -o smtpd_helo_restrictions=
    -o smtpd_sender_restrictions=
    -o smtpd_recipient_restrictions=permit_mynetworks,reject
    -o mynetworks=127.0.0.0/8
    -o strict_rfc821_envelopes=yes
```

To propogate the changes to our postfix configuration files:

```
#/kolab/etc/kolab/kolab -v -o
```

This runs the kolab backend in verbose (-v) mode once (-o) – in other words it does not go into daemon mode. We most probably already have a kolab daemon running.

To verify that your changes were propagated, issue the following.

```
#tail -14 /kolab/etc/postfix/master.cf
```

It should show the lines above.

Reload the postfix daemon:

```
#postfix reload
```

This is one of those commands that depend on you being in the Kolab environment. Check the opa section if you dont know what I mean.

Verify that the daemon reloaded successfully:

```
#tail -20 /kolab/var/postfix/log/postfix.log
```

To check that postfix added a listener for us on the correct port we telnet to the new listener on the local host.

```
#telnet 127.0.0.1 10025
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
220 mail-router1.internal.adtza.co.za ESMTP Postfix
#quit
221 Bye
Connection closed by foreign host.
```

Testing amavis-new, clamd and postfix

First we will send a normal, untainted message via the amavis listener and check that it gets delivered to the end-user.

```
#telnet 127.0.0.1 10024
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
220 [127.0.0.1] ESMTP amavisd-new service ready
#MAIL FROM:<test@example.com>
250 2.1.0 Sender test@example.com OK
#RCPT TO:<postmaster>
250 2.1.5 Recipient postmaster OK
#DATA
354 End data with <CR><LF>.<CR><LF>
#Subject: test1
#
#test1
#.
250 2.6.0 Ok, id=26807-01, from MTA: 250 Ok: ...
```

This will send a message with the subject test1 to postmaster. If you inspect the contents of the file there should be an entry:

```
X-Virus-Scanned: by amavisd-new at mydomain.com
```

While still connected to the port, we will check clamd with the Eicar.org test pattern.

```
#MAIL FROM:<test@example.com>
250 2.1.0 Sender test@example.com OK
```

```
#RCPT TO:<postmaster>
250 2.1.5 Recipient postmaster OK
#DATA
354 End data with <CR><LF>.<CR><LF>
#Subject: test2
#
#X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-
FILE!$H+H*
#.
250 2.5.0 Ok, but 1 BOUNCE
```

As you can see I set amavisd-new to reject mail and bounce it. You might want to ensure that the MAIL FROM: command is from a valid user, this will assure that you can verify the amavisd-new message.

Kolab Postfix main.cf Configuration

Now that we have verified that the scanner is working according to expectations we will set postfix to always forward messages to the amavisd-new scanner.

Edit the main.cf.template file and add the following line:

`/kolab/etc/kolab/main.cf.template`

```
content_filter = smtp-amavis:[127.0.0.1]:10024
```

Ensure that the template file change is propogated to the postfix configuration files:

```
#/kolab/etc/kolab/kolab -v -o
```

Verify that the change was properly propogated:

```
#grep content_filter /kolab/etc/postfix/main.cf
content_filter = smtp-amavis:[127.0.0.1]:10024
```

Reload the postfix daemon's configuration with:

```
#postfix reload
```

Mail coming through your server should now be content scanned by amavisd-new and clam anti-virus (through) clamd. You can verify this by sending a message through the server and verifying that it contains the X-Virus-Scanned tag mentioned above.

Startup and shutdown

We need to assure that we can properly start and stop these services as well as reload them when needed. To this extent I have created the necesarry files for startup and shutdown of the services.

Aquire the files rc.amavisd, rc.clamd and rc.kolab.diff from either Apendix A or from a better download location.

Set up the rc.clamd script:

```
#cp rc.clamd /kolab/etc/rc.d/
#chown kolab:kolab /kolab/etc/rc.d/rc.clamd
#chmod 755 /kolab/etc/rc.d/rc.clamd
```

Set up the rc.amavisd script:

```
#cp rc.amavisd /kolab/etc/rc.d/
#chown kolab:kolab /kolab/etc/rc.d/rc.amavisd
#chmod 755 /kolab/etc/rc.d/rc.amavisd
```

Patch the rc.kolab script to call the new scripts

```
#cd /kolab/etc/rc.d/
```

```
#patch < /path/to/rc.kolab.diff
```

You can test the scripts by restaring Kolab:

```
#/kolab/etc/rc.d/rc.kolab stop
stopping kolab backend ...
stopping postfix ...
postfix/postfix-script: stopping the Postfix mail system
stopping amavisd...
stopping clamd...
stopping apache ...
/kolab/sbin/apachectl stop: httpd stopped
stopping cyrus imapd ...
stopping saslauthd ...
stopping openldap ...
stopping proftpd (if running) ...

#/kolab/etc/rc.d/rc.kolab start
starting openldap ...
starting sasl ...
starting cyrus imapd ...
starting apache ...
/kolab/sbin/apachectl start: httpd started
starting clamd
starting amavisd
starting postfix ...
postfix/postfix-script: starting the Postfix mail system
starting kolab backend (please wait)
```

Updating Clam Anti-Virus

There are two aspects to keeping clam up to date. Firstly you need to keep the clam engine up to date. For this we will be using freshclam utility that is provided with clamav.

Clam pattern (db) files

Test the freshclam utility. If you need to access the internet through Proxy servers.

Check freshclam -h if you need to specify proxies, etc.

```
#freshclam
Current working dir is /kolab/var/clam
Checking for a new database - started at Wed Jul 23 14:07:20
2003
Connected to clamav.elektrapro.com.
Reading md5 sum (viruses.md5): OK
Reading md5 sum (viruses2.md5): OK
Downloading viruses.db ... done
Downloading viruses.db2 ... done
Database updated (containing in total 8870 signatures).
Database updated from clamav.elektrapro.com.
```

Install the clam_pattern_update.sh script as follows:

```
#cp /path/to/clam_pattern_update.sh /kolab/local/bin/
#chown kolab:kolab /kolab/local/bin/clam_pattern_update.sh
#chmod 755 /kolab/local/bin/clam_pattern_update.sh
```

To test if the script is performing as expected we execute:

```
#/kolab/local/bin/clam_pattern_update.sh
#cat /var/log/clam_pattern_update.log
```

After execution your syslog file should also report that clamd has restarted with the updated database.

`/var/log/syslog`

```
clamd[17069]: SelfCheck: Database status OK
```

Scheduling the updates

It is critical that the Clam pattern files keep up to date and that the engine stays current. To this extent it is highly recommended that you schedule a task to run the update script regularly.

To edit cron:

```
#crontab -e
```

Add the following entry for a update to run twice a day.

`crontab`

```
0 */12 * * *    /kolab/local/bin/clam_pattern_update.sh
```

Conclusion

At this point you should have a proper anti-virus and anti-spam solution integrated into your Kolab server.

Please report any comments, errors or improvements to kolab@codefusion.co.za

Enjoy!

Apendix A – Supporting files

/kolab/etc/rc.d/rc.amavisd

```
#!/kolab/lib/openpkg/bash /kolab/etc/rc
##
##  rc.amavisd
##
##  Copyright (c) 2003 Stephan Buys <s.buys@codefusion.co.za>
##  adopted from: rc.kolab
##  rc.kolab:
##  Copyright (c) 2002 Martin Konold
<martin.konold@erfrakon.de>
##  Copyright (c) 2002 Tassilo Erlewein
<tassilo.erlewein@erfrakon.de>


%config
    amavisd_enable="yes"

%start -p 200 -u root
    opServiceEnabled amavisd || exit 0

    echo "starting amavisd"
    if [ -f /kolab/var/amavis/amavisd.pid ]; then
       PID=`cat /kolab/var/amavis/amavisd.pid | awk '{print
$1}'`
       AMAVISS=`ps -p $PID 2>/dev/null | grep -c amavisd | awk
'{print $1}'`
       if [ $AMAVISS -gt 0 ]; then
          echo "Warning: amavisd is already running under pid
$PID!"
       else
          /kolab/local/sbin/amavisd -c /
kolab/etc/amavisd/amavisd.conf
       fi
    else
       /kolab/local/sbin/amavisd -c /
kolab/etc/amavisd/amavisd.conf
    fi

%stop -p 200 -u root
    opServiceEnabled amavisd || exit 0

    echo "stopping amavisd..."
    if [ -f /kolab/var/amavis/amavisd.pid ]; then
       kill `cat /kolab/var/amavis/amavisd.pid` 2>/dev/null
    fi
    sleep 1
    killall -9 amavisd 2>/dev/null
    exit 0

%reload -u root
    opServiceEnabled amavisd || exit 0

    echo "reload amavisd (if running) ..."
```

```
    if [ -f /kolab/var/amavis/amavisd.pid ]; then
      kill -HUP `cat /kolab/var/amavis/amavisd.pid` 2>/dev/null
    fi


%restart -u root
    opServiceEnabled amavisd || exit 0
    /kolab/etc/rc.d/rc.amavisd stop
    sleep 1
    /kolab/etc/rc.d/rc.amavisd start
```

/kolab/etc/rc.d/rc.clamd

```
#!/kolab/lib/openpkg/bash /kolab/etc/rc
##
##  rc.clamd
##
##  Copyright (c) 2003 Stephan Buys <s.buys@codefusion.co.za>
##  adopted from: rc.kolab
##  rc.kolab:
##  Copyright (c) 2002 Martin Konold
<martin.konold@erfrakon.de>
##  Copyright (c) 2002 Tassilo Erlewein
<tassilo.erlewein@erfrakon.de>


%config
    clamd_enable="yes"

%start -p 200 -u root
    opServiceEnabled clamd || exit 0

    echo "starting clamd"
    if [ -f /kolab/var/clam/clamd.pid ]; then
       PID=`cat /kolab/var/clam/clamd.pid | awk '{print $1}'`
       CLAMDS=`ps -p $PID 2>/dev/null | grep -c clamd | awk
'{print $1}'`
       if [ $CLAMDS -gt 0 ]; then
          echo "Warning: clamd is already running under pid
$PID!"
       else
          rm -f /kolab/var/clam/clamd
          /kolab/local/sbin/clamd -c /
kolab/etc/clamd/clamav.conf
       fi
    else
       rm -f /kolab/var/clam/clamd
       /kolab/local/sbin/clamd -c /kolab/etc/clamd/clamav.conf
    fi

%stop -p 200 -u root
    opServiceEnabled clamd || exit 0

    echo "stopping clamd..."
    if [ -f /kolab/var/clam/clamd.pid ]; then
```

```
      kill `cat /kolab/var/clam/clamd.pid` 2>/dev/null
    fi
    sleep 1
    killall -9 clamd 2>/dev/null
    exit 0

%reload -u root
    opServiceEnabled clamd || exit 0
    echo "reload clamd (if running) ..."
    /kolab/etc/rc.d/rc.clamd stop
    sleep 1
    /kolab/etc/rc.d/rc.clamd start

%restart -u root
    opServiceEnabled clamd || exit 0
    /kolab/etc/rc.d/rc.clamd stop
    sleep 1
    /kolab/etc/rc.d/rc.clamd start
```

rc.kolab.diff

```
--- rc.kolab    2003-07-04 14:31:26.000000000 +0200
+++ rc.kolab-new        2003-07-15 15:23:45.000000000 +0200
@@ -57,6 +57,10 @@
     echo "starting apache ..."
     /kolab/sbin/apachectl start

+    /kolab/etc/rc.d/rc.clamd start
+
+    /kolab/etc/rc.d/rc.amavisd start
+
     echo "starting postfix ..."
     /kolab/sbin/postfix start

@@ -87,6 +91,10 @@
     echo "stopping postfix ..."
     /kolab/sbin/postfix stop

+    /kolab/etc/rc.d/rc.amavisd stop
+
+    /kolab/etc/rc.d/rc.clamd stop
+
     echo "stopping apache ..."
     /kolab/sbin/apachectl stop

@@ -153,6 +161,10 @@
       kill -HUP `cat /kolab/var/imapd/imapd.pid` 2>/dev/null
     fi

+    /kolab/etc/rc.d/rc.clamd reload
+
+    /kolab/etc/rc.d/rc.amavisd reload
+
     echo "reload postfix ..."
```

```
    /kolab/sbin/postfix reload
```

clam_pattern_update.sh

```
#!/bin/bash
eval `/kolab/etc/rc --eval all env`
/kolab/local/bin/freshclam > /var/log/clam_pattern_update.log
2>&1
/kolab/etc/rc.d/rc.clamd restart
```

/kolab/etc/clamd/clamav.conf

```
LogSyslog
PidFile /kolab/var/clam/clamd.pid
DataDirectory /kolab/var/clam
LocalSocket /kolab/var/clam/clamd
ScanArchive
```