Pull mails with getmail and dovecot

purzel wrote in our forum a post but this was still a Firmware which has not mailstation included then. I put all the necessary steps out of his post. In the following the steps which have to be performed with actual firmwares as well

All lines beginning with $ mean that this is a command on the CLI directly

Trouble Shooting

If troubles with getmail occure which show the following msg in the logs:

Filter error (filter Filter_external spamc (allow_root_commands="True", arguments="('-s 250000', '-p 783', '-u USER')", command="spamc",exitcodes_drop="('99', '100')", exitcodes_keep="('0',)", group="None", ignore_stderr="False", path="/usr/syno/mailstation/bin/spamc", unixfrom="False", user="None") returned 64 (Error in argument 1, char 2: argument required for option s)

Then you should change the arguments within your rc files tot he follwing style

arguments = ("-s","250000","-p","783","-u","USER",)

Prequisites

The following packages have to be installed

- ~~Dovecot~~ (included in the firmware)
- ~~dovecot-doc~~ (only if you need the docs)
- py25-getmail
- py-getmail-common
- python25
- ~~cron~~ (not really needed as the call can be made with Syno cron as well)

If you cannot find a particular ipkg package then a

$ ipkg list | grep PART_NAME

could possibly help

Prepare dovecot

Every user which want to use getmail need the following directory be present in its homedir. The directory is called .getmail (don't forget the leading dot to hide the dir). The dir must be owned by the user that want to use getmail.

#change to users home
$ cd /volume1/homes/USERNAME
$ mkdir ./.getmail
$ chown USERNAME ./.getmail && chmod 0700 ./.getmail

Within the .getmail-directory a so called rc files is expected for every external account to be pulled. The name of the file is not important as the file will be specified on the CLI.
The file could look like this

```
[options]
delete = true
message_log = /volume1/homes/USERNAME/.getmail/log

[retriever]
type = SimplePOP3Retriever
server = pop.gmx.net
port = 110
username = externalUserName
password = PasswordForExternalAccount
use_apop = false
timeout = 180
delete_dup_msgids = false

[destination]
type = Maildir
path = /volume1/homes/USERNAME/.Maildir/
user = externalUserName
filemode = 0600
```

At this point I want to mention the official manual of getmail, which describes all the parameters possible for this program

It's important that you are aware that POP3 will send your pasword un-ecrypted to the server. But luckily the mailstation allows the user of SSL connections to the servers. So if your provider does allow a secure login, then you should use it as described below (example with gmail)

```
type = SimplePOP3SSLRetriever
server = pop.googlemail.com
port = 995
username = myaccount@gmail.com
password = MegaGigaGeheim
```

Now you need to „tell" getmail which files to use for fetching the external account. To do this just create a file called getmail.sh directly in USERNAMES home.
This file should be executable (chmod +x)

```
#!/bin/sh
/opt/bin/getmail -q -d --rcfile /volume1/homes/Hans/.getmail/gmx.rc
```

Only set option **–d** (delete) if getmail runs without error. This means that after loading the mail, it will be deleted from external account. For testing (until getmail runs fine) better use option **–l** (leave)

After that you can give the script a first try. Just start the script as the user who the script belongs to and see what happen.

$ su USERNAME –c /volume1/homes/USERNAME/getmail.sh

Getmail does log into a log file per rc file. You can find the logs at **/volume1/homes/USERNAME/.getmail/log**

Automatic calls via cron

After the manual call works properly it's time to implement a regular call via cron. There are different approaches depending on which version of cron you're using. Eighter syno cron or ipkg cron.

Syno-cron

Syno-cron does not support the "who" for cronjobs. So per default all jobs are running with root, which makes getmail crash. Getmail ALWAYS need to be run as user the mails are for. So if you want to use syno-cron the following would to the job

***/15 * * * * root su USERNAME –c "/volume1/homes/Hans/getmail.sh &>/dev/null"**

Note that for syno-cron only root is accepted in "who" and therefore you would have to run the command via su and USERNAME. **Furthermore syno-cron expects TABS as whitespaces (except for the command to run)**

Cron ipkg

For ipkg cron jobs on per user base can be defined. The job files are stored in /opt/var/cron/crontabs which contains one file per user. You can create a new file in there for your user

$ touch /opt/var/cron/crontabs/USERNAME
$ chown USERNAME /opt/var/cron/crontabs/USERNAME
$ chmod 0600 /opt/var/cron/crontabs/USERNAME

The cron file could look like that:

***/15 * * * * /volume1/homes/Hans/getmail.sh &>/dev/null**

This calls the script all 15 minutes. Note that "who" must not be specified as this is a user specific cron file. **Note as well that normale single whitespaces are used!**

Install Spamassassin

Although Spamassassin is coming with the firmware, I suggest to install the version from ipkg. The following packages are needed for SA:

* perl

* perl-io-socket-ssl

* spamassassin

After the installation the config file of SA ha sto be adjusted like the following **(/opt/etc/spamassassin/local.cf**)

```
rewrite_header_subject *****SPAM*****
required_score 5.0
use_bayes 1
bayes_auto_learn 1
bayes_ignore_header X-Bogosity
bayes_ignore_header X-Spam-Flag
bayes_ignore_header X-Spam-Status
bayes_ignore_header X-getmail-filter-classifier
```

The last line is important, otherwise SA will classify all new mails from getmail as spam!

Next step would be to check/amend the startscript of SA **(/opt/etc/init.d/S62spamd**) which could look as follows

```
#!/bin/sh
echo "Starting spamd"
/opt/bin/spamd -d -c -m 1 --max-conn-per-child=100 --pidfile=/var/run/spamd.pid -p 783
```

Important that you specify the port (-p 783) in order to tell SA which port to listen on.
Finally ensure that the script is executable:

```
$ chmod +x /opt/etc/init.d/S62spamd
```

Now we have to tell getmail to use SA when fetching mails. Create a hidden dir within users home

```
$ mkdir /volume1/homes/USERNAME/.spamassassin
$ chown USERNAME /volume1/homes/USERNAME/.spamassassin
```

Finally add in every rc file you want to use with SA the following config

```
[filter-spamassassin]
type = Filter_external
path = /opt/bin/spamc
allow_root_commands = true
arguments = ("-s 250000", "-p 783", "-u USERNAME", )
```

So that's it. SA should run together with getmail to get your external accounts. Call the getmail script and have a look into the logfiles. When SA is called for the first time then a line should appear in log informing that **/volume1/homes/USERNAME/.spamassassin** has been created. If any problem occurs it should be noted in the logfile **(/var/log/messages**)

If everything run well then every mail that getmail received should contain a header line like:

X-Spam-Checker-Version: SpamAssassin 3.1.8 (2007-02-13) on DiskStation