Configuring Gmail as a Sendmail email relay

Contents[Hide]

1.Introduction

In this configuration tutorial we will guide you through the process of configuring sendmail to be an email relay for your gmail or google apps account. This allows you to send email from your bash scripts, hosted website or from command line using mail command. Other examples where you can utilize this setting is for a notification purposes such or failed backups etc. Sendmail is just one of many utilities which can be configured to rely on gmail account where the others include postfix, exim , ssmpt etc. In this tutorial we will use Debian and sendmail for this task.

2. Install prerequisites

```
# apt-get install sendmail mailutils sendmail-bin
```

3. Create Gmail Authentication file

```
# mkdir -m 700 /etc/mail/authinfo/
# cd /etc/mail/authinfo/
```

next we need to create an auth file with a following content. File can have any name, in this example the name is gmail-auth:

```
AuthInfo: "U:root" "I:YOUR GMAIL EMAIL ADDRESS" "P:YOUR PASSWORD"
```

Replace the above email with your gmail or google apps email.

Please note that in the above password example you need to keep 'P:' as it is not a part of the actual password.

In the next step we will need to create a hash map for the above authentication file:

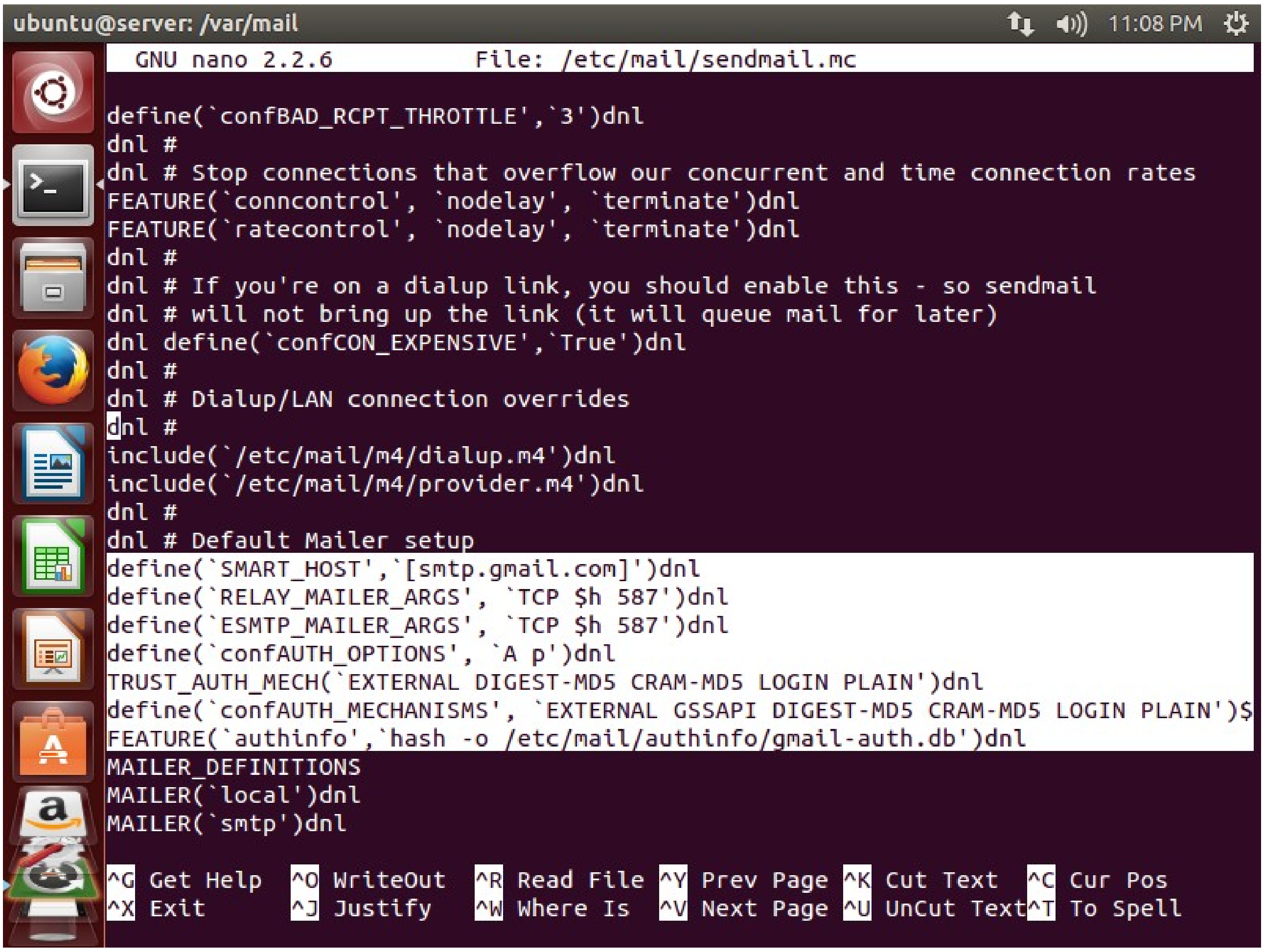
```
# makemap hash gmail-auth < gmail-auth
```

4. Configure your sendmail

Put bellow lines into your sendmail.mc configuration file **right above first "MAILER" definition line**:

```
define(`SMART_HOST',`[smtp.gmail.com]')dnl
define(`RELAY_MAILER_ARGS', `TCP $h 587')dnl
define(`ESMTP_MAILER_ARGS', `TCP $h 587')dnl
define(`confAUTH_OPTIONS', `A p')dnl
TRUST_AUTH_MECH(`EXTERNAL DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
define(`confAUTH_MECHANISMS', `EXTERNAL GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN
PLAIN')dnl
FEATURE(`authinfo',`hash -o /etc/mail/authinfo/gmail-auth.db')dnl
```

Do not put the above lines on the top of your sendmail.mc configuration file !



```
  GNU nano 2.2.6          File: /etc/mail/sendmail.mc

define(`confBAD_RCPT_THROTTLE',`3')dnl
dnl #
dnl # Stop connections that overflow our concurrent and time connection rates
FEATURE(`conncontrol', `nodelay', `terminate')dnl
FEATURE(`ratecontrol', `nodelay', `terminate')dnl
dnl #
dnl # If you're on a dialup link, you should enable this - so sendmail
dnl # will not bring up the link (it will queue mail for later)
dnl define(`confCON_EXPENSIVE',`True')dnl
dnl #
dnl # Dialup/LAN connection overrides
dnl #
include(`/etc/mail/m4/dialup.m4')dnl
include(`/etc/mail/m4/provider.m4')dnl
dnl #
dnl # Default Mailer setup
define(`SMART_HOST',`[smtp.gmail.com]')dnl
define(`RELAY_MAILER_ARGS', `TCP $h 587')dnl
define(`ESMTP_MAILER_ARGS', `TCP $h 587')dnl
define(`confAUTH_OPTIONS', `A p')dnl
TRUST_AUTH_MECH(`EXTERNAL DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')dnl
define(`confAUTH_MECHANISMS', `EXTERNAL GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN PLAIN')$
FEATURE(`authinfo',`hash -o /etc/mail/authinfo/gmail-auth.db')dnl
MAILER_DEFINITIONS
MAILER(`local')dnl
MAILER(`smtp')dnl

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

In the next step we will need to re-build sendmail's configuration. To do that execute:

```
# make -C /etc/mail
```

Reload sendmail service:

```
/etc/init.d/sendmail reload
```

and you are done.

5. Configuration test

Now you can send an email from your command line using mail command:

```
$ echo "Just testing my sendmail gmail relay" | mail -s "Sendmail gmail Relay" my-email@my-domain.com
```