Notes on configuring Postfix to work with an SSL and SASL enabled relay host

Jack-Benny Persson

31 oktober 2015

1 Background

For as long as I can remember I've been running my own mail servers, sometimes just locally, sometimes live on the internet. My current mail server has been in operation for about 4 years now. I use my ISP's SMTP as a relay host since port 25 is blocked by my ISP. This has been working really well for the last four years. But all of the sudden it stopped working with an error saying something like "host access denied". After some Googling around I found that this message is common if the relay host is using SSL/TLS and SASL authentication. So I've check my ISP's website and found that had in face switched to forcing SSL/TLS and authentication. What supprised me a bit was that I had to use an stunnel between my mail server and the relay host. But things weren't that difficult get going after all. Here's my note on how I setup Postfix to work with a relay host that uses SSL/TLS and authentiation. Along the way are also some tips & trix.

2 Get to work

2.1 The tunnel

We'll start by setting up our SSL tunnel from our mail server to our ISP's SMTP server. From now on I'm going to assume you're on a Debian system. Start by installing it with `apt-get install stunnel` as root. After it's installed succesfully we'll need to enable it in `/etc/default/stunnel`. Set `ENABLE=1` in this file. That's the only change you need to do in this file. Next we need to create a configuration file for stunnel. Create a new file in `/etc/stunnel/` that ends with `.conf`, for example `isp-smtp.conf`. Here's what my file looks like, change what's needed to fit your ISP.

```
[bahnhof-smtp-tls]
accept = 127.0.0.1:10025
client = yes
connect = mail2.bahnhof.se:465
protocol = smtp
```

Next we need to restart the stunnel with `/etc/init.d/stunnel restart`. If everything is working you should see the stunnel service with `ps -aux | grep stunnel` and `netstat -tua`.

2.2 Testing the tunnel

Now that our tunnel is up and running, we like to test it manually to see that everything is working as expected. First we need to do some preparation, mainly create base64 strings of our username and password for our ISP's SMTP. This is done like shown below.

```
jake@elektra:~$ echo -n 'myuser' | base64
bXl1c2Vy
jake@elektra:~$ echo -n 'mysecretpass' | base64
bXlzZWNyZXRwYXNz
jake@elektra:~$
```

Now save those base64 encoded string in your clipboard and connect to your ISP SMTP through the tunnel we just built. See the example session below.

```
jake@elektra:~$ telnet localhost 10025
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 smtp-auth.bahhof.se
ehlo test
250-smtp-auth2.bahnhof.se
250-PIPELINING
250-SIZE 52428800
250-ETRN
250-STARTTLS
250-AUTH PLAIN LOGIN
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
```

```
auth login bXl1cVy
334 UGFzc3dvcmQ6
bXlzZWNyZXRwYXNz
235 2.0.0 OK Authenticated
mail from: jack-benny@cyberinfo.se
250 2.1.0 jack-benny@cyberinfo.se... Sender ok
rcpt to: jack-benny@farhult.net
250 2.1.5 jack-benny@farhult.net... Recipient ok
data
354 Enter mail, end with "." on a line by itself
Subject: Test
Testing some stuff
.
250 2.0.0 n713Y5rh055422 Message accepted for delivery
quit
221 2.0.0 smtp-auth.bahnhof.se closing connection
Connection closed by foreign host.
```

2.3 Configuring Postfix

Now that we're certain the tunnel is working we move on to setting up our Postifx. First thing we need to do is to create a file which contains our username and password for our ISP's SMTP. For example create the file `/etc/postfix/isp_smtp_pass` with the following content (adjust to your username/password).

```
[127.0.0.1]:10025 myuser:mysecretpassword
```

Next we need to make it a db with the command `cd /etc/postfix && postmap isp_smtp_pass`. You should now have a file callded `isp_smtp_pass.db` in `/etc/postfix/`. If so, you can now remove the plain file `isp_smtp_pass` so you don't have your username and password lying around on the server in plain text.

Now it's time to configure Postfix to use the tunnel we've setup and the SASL authentication. Insert the following three rows in your `main.cf` file, usually it's in `/etc/postfix`.

```
smtp_sasl_auth_enable = yes
smtp_sasl_security_options =
smtp_sasl_password_maps = hash:/etc/postfix/isp_smtp_pass
```

Now all we need to do is restart Postfix with `/etc/init.d/postfix restart` as root and we're ready to begin sending mail again. If something isn't working make sure to check Postfix logfiles and also the stunnel logs.