

SMTP AUTH Using Cyrus and Postfix

By Saul Cisneros

Installing the SMTP AUTH service can be quite a challenge. The challenge doesn't come from a lack of understanding the SASL protocol, but the lack of straight forward and up to date information. Therefore, this document sets out to briefly explain the bare requirements for a Cyrus SASL implementation.

Postfix and must already be installed from source and working. This guide should work on any Posix compliant system. Solaris 10 was used while writing this guide. However, like all systems, if you have a non-standard file system layout or directory scheme, you will have to make sure you specify the correct binary and library paths in order to make sure everything works correctly. Also, since Cyrus will be installed from source, and Postfix will need to be re-compiled with the Cyrus SASL options, Gcc must be installed and working correctly as well.

The versions of software used in this guide are:

Postfix 2.3.8
Cyrus 2.1.22

After you have downloaded and unpacked the Cyrus package, CD into the Cyrus directory. Run the following command as root:

```
# ./configure --enable-anon --enable-plain --enable-login --disable-krb4 --with-saslauthd=/var/run/saslauthd --with-pam --with-openssl=/usr/local/ssl --with-plugin-dir=/usr/local/lib/sasl2 --enable-otp
```

This checks your system for any needed dependencies and also sets up the configuration for the Makefile that will be used to compile and install the Cyrus source. You can setup a prefix directory here, but I don't recommend doing that the first time since that has a way of complicating things very quickly. You can leave out the `--with-openssl` flag if you don't plan on using SSL. Next run:

```
# make
```

This compiles the options you specified into the program. This may take a while depending on the speed of your system.

```
# make install
```

This installs all the files to their proper locations. If you didn't get any errors, you're doing well. If you did, work through them one at a time, and try not to get frustrated. If you're tired take a nap before continuing.

Now you will need to re-compile Postfix to support Cyrus. Run the following:

```
# make makefiles CCARGS="-DUSE_SASL_AUTH -DUSE_CYRUS_SASL  
-I/usr/local/include/sasl" AUXLIBS="-L/usr/local/lib -lsasl2"
```

Be aware that if you are using Solaris 10 as well then you will need the Solaris 10 specific options in order to compile the software correctly. Next, just make and install.

```
# make
```

Compiles the source

```
# make install
```

This installs everything to their correct places. Keep in mind that after installing the files the installer will ask you a few questions about how you prefer to setup your installation. Choose the default options and you should be good.

Now it's time to edit the Postfix configuration file `/etc/postfix/main.cf`. Open the file with your favorite text editor, preferably VI because it's the best. Add the following anywhere in the file; I added this to the bottom of mine.

```
smtp_sasl_path = smtpd  
smtp_tls_auth_only = no  
smtp_sasl_auth_enable = yes  
smtp_sasl_password_maps = dbm:/etc/sasl2  
smtp_sasl_mechanism_filter = !gssapi, !external, static:all  
smtp_sasl_type = cyrus  
smtp_sasl_path = /var/run/saslauthd/mux/mux  
smtp_sasl_security_options = noanonymous  
smtp_sasl_local_domain =  
broken_sasl_auth_clients = yes
```

For the password maps I'm using the DBM method because my OS doesn't support the Hash method. This option will depend on your operating system. The SASL path is where Postfix will connect to the SASL socket. The broken SASL clients option is for compatibility with older versions of Microsoft Outlook. Also note that we are using smtp options here, not smtpd. Only specify the smtpd options if you are using SSL, otherwise the implementation will fail. Also you will need to include this in your Postfix configuration file:

```
smtpd_recipient_restrictions = permit_sasl_authenticated,permit_mynetworks,check_relay_domains
```

The highlighted part should be added to the `smtpd_recipient_restrictions` line. You may or may not have the other options.

Now you will need to create the `/etc/postfix/sasl/smtpd.conf` file for Postfix. It should contain the following:

```
pwcheck_method: saslauthd
saslauthd_version: 2
mech_list: PLAIN LOGIN
saslauthd_path: /var/run/saslauthd/mux/mux
```

This instructs Postfix to use the Saslauthd method and authenticate users against users that exist on the server. If you have already created users on your server to check mail via POP3, then you should just be able to use one of the usernames and passwords to authenticate.

Now its time to start SASL.

```
# saslauthd -m /var/run/saslauthd/mux -a pam -d &
```

This starts the SASL authentication daemon, with the SASL socket in the stated location. The Ampersand tells the server to run it in the background. Now you will need to test your connection to the server.

```
# /usr/local/sasl/sbin/testsasauthd -u username -p password -f
/var/run/saslauthd/mux/mux
```

Where username is a user on your system and the password is of course the password. If you get an “Ok Success” response, you are good to go further. If not, don’t despair. Just take your time and troubleshoot. Next run:

```
# saslfinger -s
```

This will verify the Postfix SASL configuration. The “s” flag specifies the server configuration for relay users. There is a “c” flag as well but that is for the server to server authentication option which is not discussed here nor needed by average SMTP AUTH users. The output you will need to be most concerned about is this:

```
-- mechanisms on localhost --
250-AUTH LOGIN PLAIN
250-AUTH=LOGIN PLAIN
```

If you see at least this output, then you are ok. If not check your configuration. All you should have to do now is configure your email client to connect via authenticated SMTP. If all is working, congratulations!