Using certbot

acme.sh

A lot of installations use acme.sh now. The simplest way to figure out things on these installations is to run

```
acme.sh --list
```

or

```
acme.sh --help
```

works.

ISPConfig uses this as the default. If certbot is not installed on installation or upgrade, acme.sh is automagically downloaded and installed.

certbot

First, certbot can also be called letsencrypt on some really, really old machines. And, it is stored in various places, depending on how you set it up.

On a Devuan servers, the certificates are stored in /etc/letsencrypt. A lot of this is taken from https://certbot.eff.org/docs/using.html, which is confusing and hard to use, but does give an exhaustive list of the commands at the bottom.

See what certificates are on system

```
certbot certificates
```

Sample output would be as follows. Note that you may have multiple entries, and not all entries will have multiple domains in them.

```
Saving debug log to /var/log/letsencrypt/letsencrypt.log

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- -
Found the following certs:
  Certificate Name: mail.example.com
    Domains: mail.example.com imap.example.com smtp.example.com
    Expiry Date: 2020-10-26 12:09:56+00:00 (VALID: 37 days)
    Certificate Path: /etc/letsencrypt/live/mail.example.com/fullchain.pem
    Private Key Path: /etc/letsencrypt/live/mail.example.com/privkey.pem
```

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
- -
```

Delete an existing domain

```
certbot delete --cert-name mail.example.com
```

will delete the entire certificate (including all aliases)

Add a new domain/alias

Ok, this one is tricky as you need a way to authenticate that you own the domain. The simplest way to do this is if you have a web server running. I have a couple of mail servers that I wanted certs for, so I installed apache2, then set it up for the server name plus a couple of aliases.

```
certbot certonly --authenticator webroot -w /var/www/html --cert-name
myserver.example.com -d
smtp.example.com,mail.example.com,myserver.example.com
```

This creates a certificate named myserver.example.com, with the names smtp, mail and myserver (yes, you have to add that if you want it on the cert). That cert will be located in /etc/letsencrypt/live/myserver.example.com/, a the certificate is valid for all three names.

Editing an alias from a cert

The above syntax allows you to add/delete aliases from an existing certificate. Simply rerun with the way you want the certificate to look. So, for example, if I wanted to remove the mail alias and add a dovecot alias, I'd do the following.

```
<code bash>certbot certonly --authenticator webroot -w /var/www/html --cert-
name myserver.example.com -d
smtp.example.com,dovecot.example.com,myserver.example.com
```

certbot will ask you if you really want to do this, and then it will do it.

Setting up smtp and imap

My favorite server is postfix for smtp and dovecot for imap/imaps, so this will cover that. Once you have your certificates downloaded, do the following:

postfix

postfix has a great cli tool named postconf which allows you to modify the configuration from the cli,

so I'll use that here. This assumes you have the certs installed.

Basically, we'll create a symbolic link from the postfix directory (/etc/postfix on Devuan) and point to the live certs in the letsencrypt live directory.

Why not simply point to letsencrypt directly? Because, we'll use the same link with dovecot later, so we'll just point both of them to the same symbolic link.

```
# link the cert
ln -s /etc/letsencrypt/live/myserver.example.com/fullchain.pem
/etc/postfix/smtpd.cert
# and the private key
ln -s /etc/letsencrypt/live/myserver.example.com/privkey.pem
/etc/postfix/smtpd.key
# add them to postfix
postconf -e smtpd_tls_cert_file=/etc/postfix/smtpd.cert
postconf -e smtpd_tls_key_file=/etc/postfix/smtpd.key
# restart postfix
service postfix reload
```

dovecot

I'm sure there is an easier way to do this, but I just edit /etc/dovecot/dovecot.conf. I understand that Debian goes crazy with the conf.d, but I use a lot of ISPConfig stuff (which doesn't use that), and honestly I think it makes life more difficult.

Basically, just add the following two lines in the base config. This uses the same certs as postfix

```
ssl_cert = </etc/postfix/smtpd.cert
ssl_key = </etc/postfix/smtpd.key
```

Now, restart dovecot

```
service dovecot restart
# watch for any errors, ^c to break out of tail -f
tail -f /var/log/mail.log
```

From:
https://wiki.linuxservertech.com/ - **Unix Server Tech Knowledge Base**

Permanent link:
https://wiki.linuxservertech.com/unix/letsencrypt/certbot

Last update: **2023/07/15 07:16**