

dehydrated – setup for Slackware 15.0

Edward McGuire

February 12, 2024

Copyright © 2023, 2024 Edward McGuire.

Contents

1	Introduction	3
2	Download and install	3
3	Add a logfile to the daily cronjob	3
4	Add automatic restart of Apache	4
5	Configuration variables	4
5.1	Run as a non-root user as a safety measure	4
5.2	Set the challenge method	4
5.3	Enable the hook script.	5
5.4	Disable private key renewal	5
5.5	Set a contact email address	5
5.6	Name the lockfile	5
6	Domain table	5
7	Directories for certificates	5
8	Apache	6
9	Makefile	8

1 Introduction

The `dehydrated` utility creates and renews SSL certificates for websites. These are not self-signed certificates. They are trusted by all the major web browsers. The certificates are provided for free by Let's Encrypt. Certificate production is fully automatic. Each certificate has a 3-month lifetime.

The `dehydrated` process requests certificate production on a daily schedule. Existing certificates are renewed when they are within 30 days of expiration. When a certificate is produced, Apache is restarted to put it into effect.

The setup for this follows the excellent article by `alienbob`, “[Using Let's Encrypt to Secure your Slackware webserver with HTTPS](#)” at *Alien Pastures*.

His instructions are for Slackware 14.2. For Slackware 15.0, the procedure is a bit simpler. This is mainly because `dehydrated` now ships with Slackware.

2 Download and install

The `dehydrated` package ships with Slackware as of 15.0. It is a member of the N (network) software set.

Simply check that it is installed:

```
$ cd /var/lib/pkgtools/packages
$ ls dehydrated
dehydrated-0.7.0-noarch-3
```

(or)

```
dehydrated-0.7.1-noarch-1_slack15.0
```

3 Add a logfile to the daily cronjob

The stock `dehydrated` daily cronjob throws away its `stdout`. It hands its `stderr` back to `cron`. That too is thrown away unless you have configured your email. This change to `/etc/cron.daily/dehydrated` saves the job output in a file:

```
-/usr/bin/dehydrated -c > /dev/null
+/usr/bin/dehydrated -c >>/var/log/dehydrated 2>&1
```

Review the file `/var/log/dehydrated` for errors.

The update from 0.7.0 to 0.7.1 preserves this change. But remember this customization could be lost when `dehydrated` gets updated in the future.

4 Add automatic restart of Apache

`dehydrated` can reload Apache automatically when certificates change. The `deploy_cert` hook runs each time a certificate is produced. It is designed for reloading a service to recognize a new certificate. That is exactly what we want to do.

Copy `hook.sh` from `/usr/doc/dehydrated-(version)/docs/examples` to `/etc/dehydrated`. Add a line at the end of `deploy_cert`:

```
sudo /usr/sbin/apachectl -k graceful
```

The user needs `sudo` permission to run this command without a password. Create a file `/etc/sudoers.d/httpd_reload` containing the line:

```
metaed ALL=NOPASSWD: /usr/sbin/apachectl -k graceful
```

In this example, `metaed` is the name of the user running `dehydrated`.

In the `alienbob` article, a `sudo` change is also made for `root`. That change is no longer needed in Slackware 15.0.

5 Configuration variables

Edit `/etc/dehydrated/config`. Set the following variables.

5.1 Run as a non-root user as a safety measure

In this example the non-root user is `metaed`.

```
DEHYDRATED_USER=metaed
DEHYDRATED_GROUP=metaed
```

5.2 Set the challenge method

Let's Encrypt verifies we own a domain before handing us a certificate. This is called a "challenge". Available challenge methods are:

http-01 - test for a specific file returned by the domain HTTP service

dns-01 - test for a specific key returned by the domain DNS server

tls-alpn-01 - test for a specific response returned by the domain TLS server

The `http-01` method is the most popular because it is easy.

```
CHALLENGETYPE="http-01"
```

5.3 Enable the hook script.

```
HOOK=/etc/dehydrated/hook.sh
```

5.4 Disable private key renewal

Here we are following alienbob's lead.

```
PRIVATE_KEY_RENEW="no"
```

5.5 Set a contact email address

Warning emails are sent here when a certificate is about to expire.

```
CONTACT_EMAIL=security@newjersey.metaed.com
```

5.6 Name the lockfile

dehydrated uses this lockfile to avoid race conditions. This file must be writable by DEHYDRATED_USER.

```
LOCKFILE="${BASEDIR}/var/lock"
```

6 Domain table

The table of domains to certify is the file `/etc/dehydrated/domains.txt`. See `/usr/doc/dehydrated*` for a complete guide. Each line represents a domain to certify. The line can include the names of subdomains that should share the certificate. The file will look something like this:

```
example.com www.example.com  
example.org www.example.org
```

7 Directories for certificates

The DEHYDRATED_USER needs to write files to these directories. The example below uses metaed.

```
# mkdir -p /etc/dehydrated/accounts
# chown metaed:metaed /etc/dehydrated/accounts
# mkdir -p /etc/dehydrated/certs
# chown metaed:metaed /etc/dehydrated/certs
# mkdir -p /etc/dehydrated/chains
# chown metaed:metaed /etc/dehydrated/chains
# mkdir -p /etc/dehydrated/var
# chown metaed:metaed /etc/dehydrated/var
```

The `alienbob` article also mentions:

```
# mkdir -p /srv/www/dehydrated
# chown metaed:metaed /srv/www/dehydrated
```

But now that `dehydrated` is preinstalled with Slackware 15.0, creating this directory is done by the package installer. It is still necessary to change the user:group if you are running `dehydrated` as non-root.

When `dehydrated` is upgraded from 0.7.0 to 0.7.1, the Slackware package removes and creates the directory, so it is necessary to restore the non-root user:group after the upgrade.

8 Apache

The Let's Encrypt `http-01` challenge looks for a file on the domain website. The file has a specific filename, which Apache must be made aware of. In `/etc/httpd/httpd.conf` this can be configured once for all domains. Add outside any `VirtualHost` block:

```
Alias /.well-known/acme-challenge /srv/www/dehydrated
<Directory /srv/www/dehydrated>
    Options none
    AllowOverride None
    Require all granted
</Directory>
```

It is also necessary to enable `ssl_module` and set module parameters. Again this can be done in `/etc/httpd/httpd.conf`. This is mostly taken from `/etc/httpd/httpd/extra/httpd-ssl.conf`. You can use that instead if you prefer to configure Apache using multiple files.

```
Listen 443
LoadModule ssl_module lib64/httpd/modules/mod_ssl.so
<IfModule !ssl_module>
    Error "ASSERT: This configuration depends on ssl_module. Load
it with LoadModule."
</IfModule>
SSLCipherSuite          HIGH:MEDIUM:!MD5:!RC4:!3DES
```

```
SSLProxyCipherSuite    HIGH:MEDIUM:!MD5:!RC4:!3DES
SSLHonorCipherOrder    on
SSLProtocol             all -SSLv3
SSLProxyProtocol        all -SSLv3
SSLPassPhraseDialog     builtin
SSLSessionCache         "shmcb:/var/run/ssl_scache(512000)"
SSLSessionCacheTimeout 300
SSLRandomSeed           startup builtin
SSLRandomSeed           connect builtin
```

Within a domain's `VirtualHost` block, add domain specific SSL parameters. The file where you keep your domain blocks is up to you. Here is a simple example.

```
<VirtualHost *:443>
  ServerName EXAMPLE.COM (or whatever)
  DocumentRoot /srv/www/EXAMPLE.COM (or whatever)
  (other settings relevant to the domain)
  SSLEngine on
  SSLCertificateFile "/etc/dehydrated/certs/EXAMPLE.COM/cert.pem"
  SSLCertificateKeyFile "/etc/dehydrated/certs/EXAMPLE.COM/privkey.pem"
  SSLCertificateChainFile "/etc/dehydrated/certs/EXAMPLE.COM/chain.pem"
  SSLCACertificatePath "/etc/ssl/certs"
  SSLCACertificateFile "/etc/ssl/certs/ca-certificates/crt"
</VirtualHost>
```

9 Makefile

```
8 <makefile 8>≡
  include ../makefile
  makefile :: dehydrated.sentinel ;
  all :: dehydrated.pdf makefile
  clean :: ; rm -f dehydrated.pdf makefile
```

This code is written to file `makefile`.