letsencrypt notes

Stop - go to Iris's pages Let's Encrypt cPanel Hostname SSLs! and Let's Encrypt Domain SSLs! where good information lives. This was field notes from many months ago when the edges were much rougher. Fun read if you want to know how I spend my weekends.

What it is

https://letsencrypt.org/ is a new, automated way to acquire SSL certificates using the ACME protocal. The letsencrpt tool automates CSR generation, CA submission, domain validation, and receipt of the certificate. Quick requirements : this uses DV via http, so the domain must be assigned a DNS record, and point to the server where you're running the authentication. It's possible, using manual configuration, to do this the old fashioned way, running the command on one machine, and adding a proof-of-ownership page on the target server

Right now, it can issue certs. Automatic renewals are planned but not supported. All certificates expire after 90 days. There is no cpanel plugin. There is no installation support on centos, this will generate a cert, you will need to install it in apache the old fashioned way. See https://features.cpanel.net/topic/provide-support-for-lets-encrypt-automated-certificate-management-ssl for the cpanel discussion about this (letsencrypt, centos, and cpanel met in 2015 to discuss implementation details).

How to install (current)

clone the repo and run the tool:

```
git clone https://github.com/letsencrypt/letsencrypt
cd letsencypt/
./bootstrap/centos.sh # for centos 7
./letsencrypt-auto
```

Now, a dirty secret. Letsencrypt's tools will automatically configure apache on debian and ubuntu. They don't have support for centos right now (see https://letsencrypt.readthedocs.org/en/latest/using.html#letsencrypt-auto under plugins, apache is alpha, with a caveat of debian only). While you're there, read the rest of that page, since it's likely more up to date than this one. Running on centos as of November gives this:

```
No installers are available on your OS yet; try running "letsencrypt-auto certonly" to get a cert you can install manually
```

What this means is that you will need to stop the port 80 listener to use the tool, and ./letsencrypt-auto certonly is probably the right tool (this will negotiate a certiicate, and install it to /etc/letsencrypt/live/ for you). Right now from github this will still be signed by 'happy hacker fake CA' and will trigger warnings. So the big news is the tool got simpler to use in the past few months. One great feature is that authentication now defaults to port 80 instead of port 443, that's cool if you need to do file based authentication.

How to install (old version)

So step one will be to get the letsencrypt client. I'll be cloning this from https://github.com/letsencrypt/letsencrypt

clone the repo

```
cd /usr/local/src
 git clone https://github.com/letsencrypt/letsencrypt
 cd letsencrypt/
```

There's a nice big warning on their page, since they won't be live until September, this will give certs signed by "happy hacker fake CA".

> **Disclaimer**
>
> This is a **DEVELOPER PREVIEW** intended for developers and testers only.
>
> **DO NOT RUN THIS CODE ON A PRODUCTION SERVER. IT WILL INSTALL CERTIFICATES SIGNED BY A TEST CA, AND WILL CAUSE CERT WARNINGS FOR USERS.**

Also, behold instructions here https://letsencrypt.readthedocs.org/en/latest/using.html

let's try either the fedora or centos script and see. This should install augeas at a minimum (a configuration file library from RedHat http://augeas.net/)

run the bootstrap script to install packages

```
bootstrap/fedora.sh
```

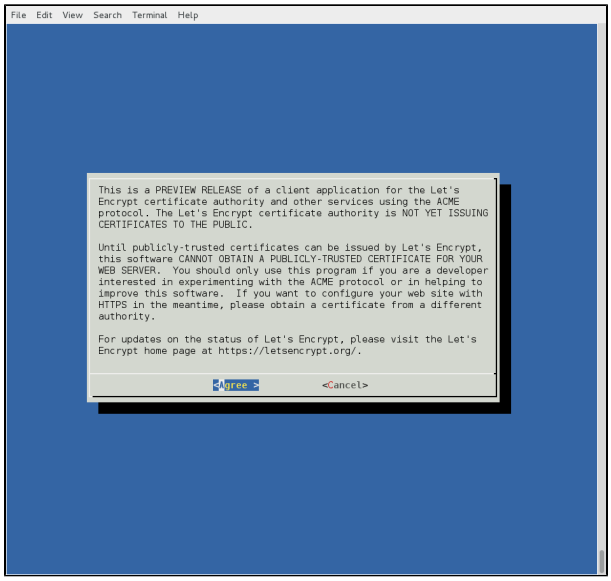
Create a virtual env and install within it, as suggested, using the python 2 installed (if you have python2.7 available, you might override with that, and nginx support is probably not interesting.):

```
virtualenv --no-site-packages -p python2 venv
./venv/bin/pip install -r requirements.txt acme/ . letsencrypt-apache/
letsencrypt-nginx/
```

Lets Encrypt!

Again, a lot has changed recently. If you have a beta invite, do see the instructions at https://community.letsencrypt.org/t/beta-program-announcements/1631 since they include the --server flag you will need. If you installed this today, the letsencrypt-auto tool has very different flags than it did in May/June when I first tested this.

```
service httpd stop           ## or do manual
./venv/bin/letsencrypt auth
```

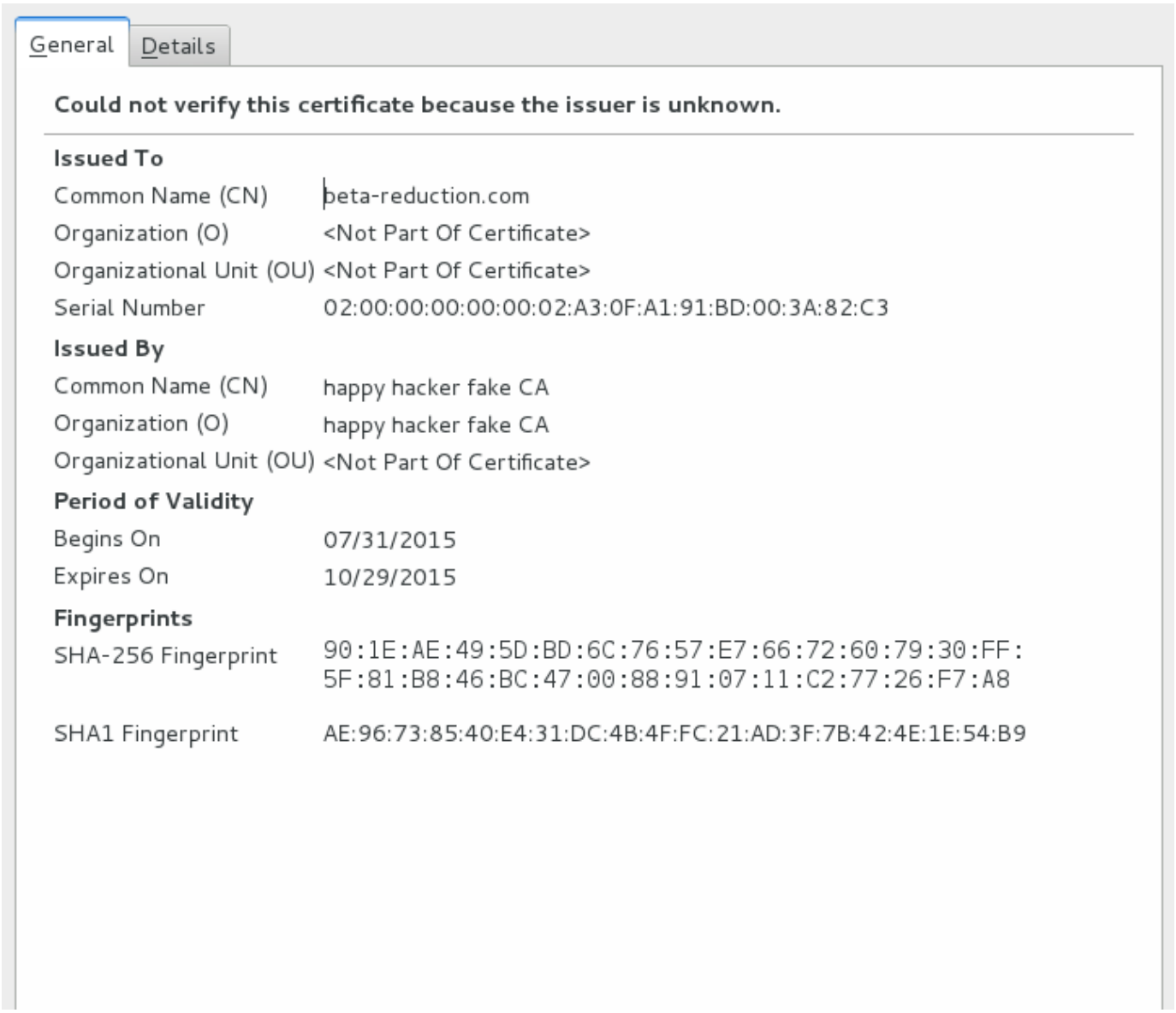


Once that finishes asking questions, then reporting great success, restart apache, and observe that these files are in /etc/letsencrypt/live/domain.com/ where there are links to the privkey, fullchain, chain, and cert. These are the certificates. We can install via cpanel, or just manually add to the apache conf... We'll just get our certs from the text files:

get the cert information

```
cat /etc/letsencrypt/live/domain.com/cert.pem
cat /etc/letsencrypt/live/domain.com/privkey.pem
cat /etc/letsencrypt/live/domain.com/chain.pem
```

Now we can install in WHM per normal, and after a restart, have a certificate warning for our signed snakeoil:


General
Details
Could not verify this certificate because the issuer is unknown.
Issued To
Common Name (CN) beta-reduction.com
Organization (O) <Not Part Of Certificate>
Organizational Unit (OU) <Not Part Of Certificate>
Serial Number 02:00:00:00:00:00:02:A3:0F:A1:91:BD:00:3A:82:C3
Issued By
Common Name (CN) happy hacker fake CA
Organization (O) happy hacker fake CA
Organizational Unit (OU) <Not Part Of Certificate>
Period of Validity
Begins On 07/31/2015
Expires On 10/29/2015
Fingerprints
SHA-256 Fingerprint 90:1E:AE:49:5D:BD:6C:76:57:E7:66:72:60:79:30:FF:
5F:81:B8:46:BC:47:00:88:91:07:11:C2:77:26:F7:A8
SHA1 Fingerprint AE:96:73:85:40:E4:31:DC:4B:4F:FC:21:AD:3F:7B:42:4E:1E:54:B9

Congratulations. Now have a cracker with cheese, it pairs well with ssl.

Manual Authentication

Manually doing this (without stopping apache) is a little trickier. It's going to look for a resource at https://domain.com/.well-known/acme-challenge/UUID, for me this was https://beta-reduction.com/.well-known/acme-challenge/5rtS1m6rnE7-9ucZL9pS2bU- So the first hurdle is you will need to install a temporary certificate (self signed from csr is fine) to get past this step, since the request is on https. One alternate, assuming the default website is assigned to this IP (you get the cpanel 404 page) is that you can put the .well-known directory in /usr/local/apache/htdocs/.

There's a note that this needs the following

- the right string in the response, this is important. Do not add any whitespace. A newline will break this.
- a 200 response code
- content-type text/plain

The trick I found with apache 2.4 was that the apache 2.2 config option DefaultType can't be used here (it just logs warnings in the error log, not helpful). Apache usually sets content type by extension, but gives text/html as the default for files without extensions. You can set this manually with ForceType, my .htaccess in acme-challenge reads the following, and checking the headers confirms this is working as needed:

htaccess set content-type

```
<Files "5rtS1m6rnE7-9ucZL9pS2bU-">
    ForceType text/plain
</Files>
```