Complete guide to install SSL certificate on your OS X server hosted website

Server

JeffTheRocker #1 April 28, 2016, 7:20am

Introduction

If like me you are, among other many tasks, a system administrator of a website hosted on OS X server, chances are your are not fully comfortable with what to do precisely in order to get your website running with a valid SSL certificate. After some struggle, I would like to share my experience on the installation of let's encrypt certificate.

The target audience of this article is people with a minimum IT skills (I assume you know how to open the terminal and some basic knowledge of shell commands), already having a configured and running website hosted on OS X and managed with the Server App. Also this article assumes you will generate the certificate on the machine hosting the server itself.

Note: The following procedure was successfully installed on two similar server, both are Mac mini running on OS X Mavericks (10.9.5) with the Server app installed, configured and running. Also I have done several installation on those servers (they are not freshly packed out of the box), which means that you may need to install other piece of software in the first step in order to make it work.

This article extend the following articles: **Mac OSX (Server): import LE certificate?** and **Installing and Configuring LetsEncrypt on a Mac OS X Client Server**

Overview

So, here is the plan:

- install let's encrypt client ;
- (test and) generate the certificates ;
- import the certificate into OS X's KeyChain ;
- configure your website in order to make it work with https ;
- (test and) automate the renewal.
- enjoy…

Step 1 Install let's Encrypt client

The first step is to instal Homebrew. Here is the official documentations on how to install homebrew `http://brew.sh/`. Basically you just have to launch

```
ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

then

```
brew doctor
```

and

```
brew update
```

Once Homebrew is installed you can install Let's Encrypt Client. The "client" is a terminal command tool that allow you to retrieve the certificates. Do not expect a nice and sleek UI, it will be command line. But it is quite simple to use and is perfect for automation. So here is the command line in order to retrieve the source code for the client:

```
git clone https://github.com/letsencrypt/letsencrypt
```

I had issue with some folder access. I don't know what was blocking but I had to change rights for python packages folder and change it back after the installation of the client. I am not sure if it is the best way to solve this problem but at least it works.

So BEFORE launching let's encrypt change the access of the folder by launching the command

```
sudo chmod -R 777 /Library/Python/2.7/site-packages/
```

Let's encrypt does not officially support OS X, you will need to launch for the first time with the `--debug` flag. Don't worry, everything will be fine.
To install the client you have to run the commad:

```
~/letsencrypt/letsencrypt-auto --help --debug
```

If everything went fine, you should see the following folder:

- `~/letsencrypt`
- `/etc/letsencrypt`

Now you can change back the rights of the python folder by doing

```
sudo chmod -R 755 /Library/Python/2.7/site-packages/
```

> ####Terminal beginner tips

- ~ means the current user home folder which is located under `/Users/your_user_name/`, where you have the folder Desktop, Documents, Downloads, Music etc…)
- in order to navigate in command line you can use the command `cd` (for Change Directory). For example `cd ~/letsencrypt`
- in order to list the content of a folder you can use the `ls` command. However `ls` won't display much information, `ls -la` is much better. You can also specify a folder for example `ls -la ~/letsencrpyt` will display the content of the folder letsencrypt wherever your current location is.
- in order to know your current location, you can use the `pwd` command line. if you do `cd ~/letsencrypt`, `pwd` will display `/Users/your_user_name/letsencrypt`

Step 2 (test and) generate the certificate

First locate your website directory (the web root). If you didn't change anything it should be under `/Library/Server/Web/Data/Default`. There, you must create the folders `.well-known` and `acme-challenge` inside `.well-known`. In order to do so launch :

```
mkdir PATH_TO_WEB_FOLDER/.well-known
```

and

```
mkdir PATH_TO_WEB_FOLDER/.well-known/acme-challenge
```

You should test that with a test file (`echo "Success" > PATH_TO_WEB_FOLDER/.well-known/acme-challenge/test.html`) and check if you can access it with your browser "`http://example.com/.well-known/acme-challenge/test.html`".

When you are sure this folder is accessible, you can start with the actual certificate generation.

It is not mandatory but I suggest you create a folder containing two scripts, one config file and logs (I tried to use the standard logs but for access issue I gave up). To do so you have to do :

```
mkdir ~/letsencrypt/my_script
mkdir ~/letsencrypt/my_script/logs
```

####Terminal Beginner tips
In order to create a file you can use vi (or nano) as command line or a texteditor. I am used to vi but it can be quite unsettling at first. Here are some few commands for vi:

- to open the vi editor launch the command `vi FILENAME` (ex: `vi cert.ini` will create the file once you saved it)
- use `i` to enter edit mode, copy the file content below, paste it with *cmd+V*, use the arrows to move

> to the placeholder (like YOUR_EMAIL), and edit it, use *esc* in order to exit edit mode, then type `:wq` in order to save and quite. If you are stuck simply press *esc* then type `:q!` and it will quit without saving.
> - in order to delete a line type esc and then `dd`
> - in order to delete a single caracter (not in edit mode) use `x`
> - to undo something type *esc* and `u`

Then create the following files:

cert.ini

```
# Use a 4096 bit RSA key instead of 2048
rsa-key-size = 4096
# Register with the specified e-mail address
email = YOUR_EMAIL
# Generate certificates for the specified domains.
domains = LIST_OF_DOMAINS_AND_SUBDOMAINS
# Uncomment to use a text interface instead of ncurses
# text = True
# To use the webroot authenticator.
authenticator = webroot
webroot-path = WEB_ROOT_FOLDER
```

Of course you have to change `YOUR_EMAIL` with your actual email address, `LIST_OF_DOMAINS_AND_SUBDOMAINS` with the list of domains ex:(`example.com, www.example.com, example.org, www.example.org`) and `WEB_ROOT_FOLDER` with the folder of your web page, where you created the `.well-known` folder

get_cert.sh

```
#!/bin/sh

DOMAIN_DEFAUT= YOUR_DOMAIN
PEM_FOLDER="/etc/letsencrypt/live/${DOMAIN_DEFAUT}/"
LOG_FOLDER="/Users/YOUR_USER/letsencrypt/my_script/logs"
DATE=$(date +"%d-%m-%y")
LOG_FILE="${LOG_FOLDER}/${DATE}.log"

# Retrieve certificate - DELETE --dry-run AFTER THE TEST RUN WORKED
sudo /Users/YOUR_USER/letsencrypt/letsencrypt-auto certonly -c cert.ini --dry
-run

# Check that everything went fine
LE_STATUS=$?

if [ "$LE_STATUS" != 0 ]; then
    echo Automated Get certificate failed:
    cat $LOG_FILE
    exit 1
fi

# Generate a passphrase - UNCOMMENT THE NEXT LINE AFTER THE TEST RUN WORKED
#PASS=$(openssl rand -base64 45 | tr -d /=+ | cut -c -30)
```

Again, here you have to change `YOUR_DOMAIN` with the first domain name you entered (in our previous example it would be ***example.com***). You also have to change `YOUR_USER` twice with your current user. You may have noted that we are using the webroot method, which will automatically place a temporary file in the `.well-knwon/acme-challenge` folder to check that the website you declare is yours indeed. You could do that manually by omitting the -webroot flag, but it is problematic later for the automatic renewal.

Now, is time to try it. Don't worry we will be using a test environement, so you won't reach the 5-7 certificate a day authorized by Let's Encrypt. Simply launch:

```
~/letsencrypt/my_script/get_cert.sh
```

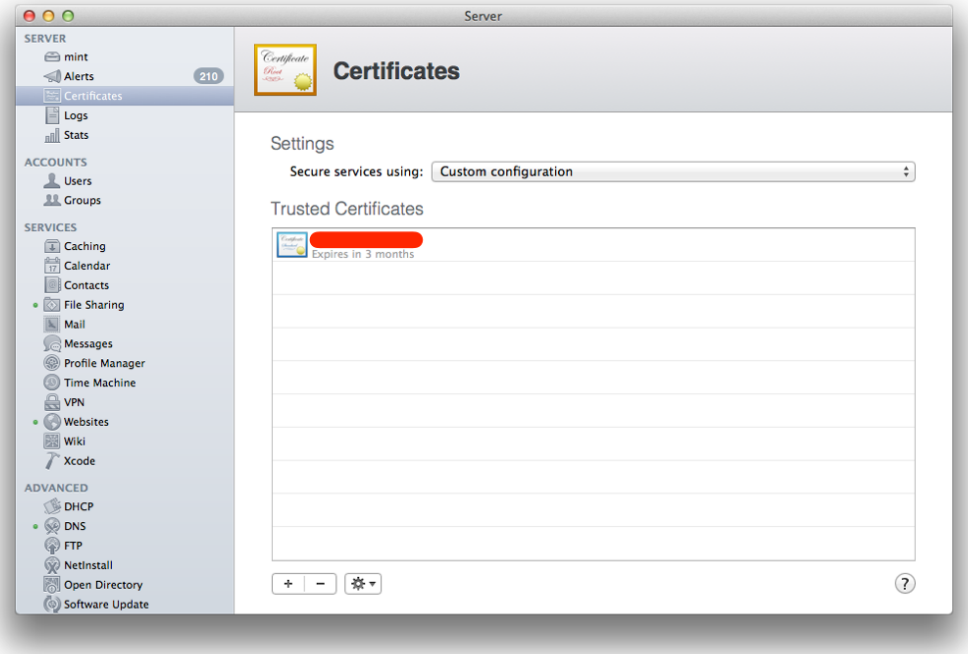
If everything went well (or after fixing the issue the script arose) you can remove, in the `get_cert.sh` script, the `--dry-run` flag and the commented line after "`UNCOMMENT THE NEXT LINE AFTER THE TEST RUN WORKED`"

Launch again the script:

```
~/letsencrypt/my_script/get_cert.sh
```

This time it should have done several things: created under `/etc/lets-encrypt/live/YOURSITE/` 3 `.pem` files, generate a .p12 file and add in the Keychain and Server app your certificate.

To check that everything went fine open the Server app and under “Certificates” you should see your shiny brand new certificate.



If it is not the case, check in Keychain that you have two new entries under “System” and “Certificates”, one for your domain and one for “Let’s Encrypt Authority X3”.

If that is still not the case check that the `.pem` and `.p12` files were generated. do the following commands:

```
sudo ls /etc/letsencrypt/live/
```

You should see your domain. Then do

```
sudo ls /etc/letsencrypt/live/YOUR_DOMAIN
```

You should see 5 files:

- `cert.pem`
- `fullchain.pem`
- `privkey.pem`
- `chain.pem`
- `letsencrypt_sslcert.p12`

if `letsencrypt_sslcert.p12` is not there then you probably didn’t provide the right domain in the `~/letsencrypt/my_script/get_cert.sh` file. If the `.pem` files are not there then you didn’t receive from let’s encrypt the certificate. Now is a good time to have a look at the log file created under

`~/letsencrypt/my_script/logs`.

Step 3 configure your website

Now you should have a working certificate in the server app. The next thing to do is to bind this certificate to sour web site. Go to Server app under Sites. If you use the default configuration (“Server Website”), you simply have to report your setting from “Server Website” to “Server Webstie (SSL)”. If you have another site, I suggest you to duplicate it (I mean create a new one and report the setting of the original), set the port to **443** and choose your certificate.

Now you should be able to access your website with `https://example.com` and see a lock next to the URL 😄
But it is by far not finished. You have to consider the following:

- Are all your subdomains included?
- Are all the internal link in your website made without the explicit `http://`
- Are all images referred without the explicit `http://` (could be the reason why you don’t have the green lock next to the URL)
- Do you want to make a full redirection from `http://example.com` to `https://example.com` or are simply some parts?

Unfortunately I cannot help you there…

Step 4 (test and) automate the renewal

The renewal script is almost the same as the `get_cert.sh` script. Therefore I propose you to duplicate `get_cert.sh`:

```
cp ~/letsencrypt/my_script/get_cert.sh ~/letsencrypt/my_script/renew.sh
```

simply change in `renew.sh` the line

```
sudo /Users/YOUR_USER/letsencrypt/letsencrypt-auto certonly -c cert.ini
```

by

```
sudo ~/letsencrypt/letsencrypt-auto renew --manual-public-ip-logging-ok --agr
ee-tos --force-renew > $LOG_FILE 2>&1
```

To test the renewal launch

```
~/letsencrypt/my_script/renew.sh
```

The expiration date & time of your certificates should have changed.

> **For production remove the `--force-renew` flag!**

In order to automate it you can add this script in the cron. To do so launch:

```
crontab -e
```

add the line

```
0 0 * * * /Users/YOUR_USER/letsencrypt/my_script/renew.sh
```

This would make the script run every midnight and check if the renewal date is close, then renew it.

Step 5 Enjoy…

Congratulation! You now deserve a beer or any drink you fancy!

6 Likes

Certbot OS X Server Support?

Multiple domains with OS X

Certbot OS X Server Support?

OSX / unauthorized & invalid response from

Certbot OS X Server Support?

Using Let's Encrypt to secure company connections

macOS Sierra 10.12.6 LE cert for Apple Wiki

SSL cert setup for Wiki on OS X Server

SSL cert setup for Wiki on OS X Server

Automatically renewing certs with macOS Server?

Instructions for macOS Server don't work

Clarification About Renewals of Certificates and Script Behaviour on MAC OS X Server

How to use Let's Encrypt cert with Mac OS X Server services (CardDAV etc)

Problem updating cert-bot OSX

"The certificate could not be exported"

macOS Sierra 10.12.6 LE cert for Apple Wiki

Let's Encrypt for os x mail

macOS: duplicate certificates after renewal

Wilcard Guide for Mac OS

---

system closed #2 May 27, 2016, 2:26pm

This topic was automatically closed 30 days after the last reply. New replies are no longer allowed.
