certbot-django Documentation

Release 0.2.0.post6

Craig Weber

May 06, 2019

Contents

This is a combined plugin for the certbot ACME client and also a Django-app for proving ACME challenges.

Full Documentation: https://certbot-django.readthedocs.io

CHAPTER 1

Contents

1.1 Installation

1.1.1 Server Installation

To use certbot-django, you must first install it in the Django application for which you'd like to obtain an SSL token.

1. Follow the instructions for installing asymmetric_jwt_auth. This is necessary for certbot to authentication with your application.
2. Install the `certbot-django` package.

```
$ pip install certbot-django
```

3. Add `certbot_django.server` to your application's `INSTALLED_APPS` list.

```
# myproject/settings.py
...
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.sites',
    'django.contrib.postgres',
    ...
    'asymmetric_jwt_auth',
    'certbot_django.server',
    ...
]
...
```

4. Include certbot-django in your applications URLs file. **In order to work, it must be placed at my.domain.com/.well-known/**.

```
# myproject/urls.py
from django.conf.urls import include, url
from django.contrib import admin
import certbot_django.server.urls

urlpatterns = [
    url(r'^admin/', include(admin.site.urls)),
    url(r'^\.well-known/', include(certbot_django.server.urls)),
    ...
]
...
```

5. Deploy your application so that it's accessible to the Internet over HTTP on port 80.
6. Run migrations to create the necessary models.

```
$ python manage.py migrate
```

1.1.2 Client Installation

Next, you must install certbot-django on the system where you plan run the certbot client.

Note: Unfortunately, you must install certbot-django globally on the system. Certbot will not be able to find the package if you install it in a virtual environment, even if certbot is installed and run from the same virtual environment.

```
$ pip install certbot
$ pip install certbot-django
```

1.2 Usage

1.2.1 First Run

The first time you run certbot for an application, you need to take a few extra steps to tell certbot how to authenticate with your application.

1. Make a directory to store the RSA private keys certbot_django will generate. This directory can be anywhere, but it should be owned by the user that certbot will run as and should have permissions set to restrict access from other users. In this example, we'll user a directory inside the user's `.ssh` directory.

```
mkdir -p ~/.ssh/certbot/
chmod 700 ~/.ssh/certbot/
```

2. Using Django Admin on your application, create a user (maybe called `certbot`). Give it a strong (or unusable) password and then throw the password away. You'll never need to use it. Make `certbot` a staff user and give it permission to add and delete ACMEChallenge objects.
3. Run certbot and tell it to use the `certbot_django:auth` authenticator plugin.

```
certbot certonly -d example.com \
                 -a certbot-django:auth \
                 --certbot-django-auth-key-directory=~/.ssh/certbot/ \
                 --certbot-django:auth-username=certbot \
                 --certbot-django:auth-public-ip-logging-ok
```

Theres a few things here to pay attention to here.

Certbot Options:

-d domain Get a certificate for this domain. Your application must already be available over HTTP at this domain on port 80.

-a authenticator Tell certbot to use certbot-django for domain authentication by specifying `certbot-django:auth`.

Certbot-Django Options:

--key-directory=directory Tell certbot-django where to generate / look for a private key when authenticating with your application. For example, `~/.ssh/certbot/`.

--username=username Tell certbot-django which Django user to login as when authenticating with your application. For example, username `certbot`.

--public-ip-logging-ok LetsEncrypt will log the Public IP of you machine as having requested a certificate for this domain. Pass this flag to allow this automatically. Otherwise, certbot will prompt you for permission to continue.

4. Certbot will print a public key in PEM format and ask you to add it to the `certbot` user (which you created a moment ago) in Django. You can do that using the Django Admin at `http://my.domain/admin/asymmetric_jwt_auth/publickey/add/`. You can think of this step as being equivalent to adding a public key to a user's `~/.ssh/authorized_keys` file on a *nix system.
5. After saving the public key, press enter in your terminal to continue.
6. Certbot will now authenticate with the application using the public / private keypair, add ACME challenge objects to your Django installation, tell LetsEncrypt to verify them, and finally cleanup by removing the challenge objects.

When complete, certbot will tell you where your shiny new SSL certificate is saved. If using an automated installer, certbot will take care of everything for you. Otherwise, you'll need to manually install the certificate on your server.

1.2.2 Subsequent Runs

Since LetsEncypt issues short-lived certificates (3-months), it's important to automate the renewal process. Fortunately, certbot makes this easy.

1. Leave the certbot user and it's public key on your application. It will need to be there every time the certificate gets renewed.
2. Leave the private key storage directory (in the example above, `~/.ssh/certbot/`) intact. The private key stored there will be used every time the certificate gets renewed.
3. Renew the certificate using certbot, again, telling it where the keys art and what user to authenticate as. This can be run manually or as a cronjob. *Note: If you run this as a cronjob, don't run it too frequently, since LetsEncypt aggressively throttles it's API. Currently it is limited to 5 requests per 7 days. This is fine since you only need to renew once per 3 months.*

```
certbot renew -d example.com \
              -a certbot-django:auth \
              --certbot-django-auth-key-directory=~/.ssh/certbot/ \
              --certbot-django-auth-username=certbot \
              --certbot-django-public-ip-logging-ok
```

If using an automated installer, certbot will take care of everything for you. Otherwise, you'll need to manually install the certificate on your server.