

Hobo - Development #16863

poser SECURE_PROXY_SSL_HEADER

13 juin 2017 15:23 - Frédéric Péters

Statut:	Fermé	Début:	13 juin 2017
Priorité:	Normal	Echéance:	
Assigné à:		% réalisé:	0%
Catégorie:		Temps estimé:	0:00 heure
Version cible:		Planning:	
Patch proposed:	Oui		

Description

Je rate sans doute un truc, dans nos exemples de configuration nginx, et dans nos configurations en vrai, on fait :

```
proxy_set_header X-Forwarded-SSL on;
proxy_set_header X-Forwarded-Protocol ssl;
proxy_set_header X-Forwarded-Proto https;
```

Festival de la redondance, mais en l'absence de settings.SECURE_PROXY_SSL_HEADER, qui est vide par défaut, ça tombe sur :

```
def _get_scheme(self):
    return 'https' if os.environ.get("HTTPS") == "on" else 'http'
```

Et le HTTPS dans l'environnement, j'imagine que c'est nginx qui le pose (j'ai regardé les sources sans identifier vraiment, mais je n'ai pas trouvé de mention d'HTTPS dans gunicorn).

Anyway, pour faciliter l'offloading HTTPS, hobo ne pourrait pas taper SECURE_PROXY_SSL_HEADER = ('X-Forwarded-Proto', 'https') dans debian_config_common.py ? (ou une des autres valeurs qu'on pose)

Révisions associées

Révision 62e597e1 - 04 décembre 2017 11:49 - Frédéric Péters

debian: set SECURE_PROXY_SSL_HEADER for nginx/runserver integration (#16863)

Historique

#1 - 13 juin 2017 15:33 - Benjamin Dauvergne

Il me semble que c'est gunicorn qui le gère à la place de Django et met le os.environ['HTTPS'] = 'on'.

#2 - 13 juin 2017 15:46 - Frédéric Péters

ok, creusé davantage, gunicorn définit :

```
class SecureSchemeHeader(Setting):
    name = "secure_scheme_headers"
    section = "Server Mechanics"
    validator = validate_dict
    default = {
        "X-FORWARDED-PROTOCOL": "ssl",
        "X-FORWARDED-PROTO": "https",
        "X-FORWARDED-SSL": "on"
    }
```

et face à l'entête, termine en mettant :

```
environ['wsgi.url_scheme'] = url_scheme
```

et dans django c'est WSGIRequest qui définit :

```
def _get_scheme(self):
    return self.environ.get('wsgi.url_scheme')
```

(plus tard je regarderai ce que fait uwsgi).

#3 - 13 juin 2017 21:28 - Frédéric Péters

uwsgi regarde HTTP_X_FORWARDED_SSL et s'il y a "on" dedans il passe l'info à wsgi dans wsgi.url_scheme.

#4 - 03 décembre 2017 13:20 - Frédéric Péters

- Fichier 0001-debian-set-SECURE_PROXY_SSL_HEADER-for-nginx-runserv.patch ajouté
- Statut changé de Nouveau à En cours
- Patch proposed changé de Non à Oui

À l'occasion de la préparation d'une image de développement pour la formation à Strasbourg, je retombe ici; dans une configuration qui a nginx qui fait reverse proxy http vers un service lancé en runserver local, il n'y a rien pour indiquer l'https.

Avec django 1.8 il y a encore moyen d'arranger les choses en posant HTTPS=on dans l'environnement du runserver (comme noté dans la description), mais dans django master, le `_get_scheme` est passé de `return 'https' if os.environ.get("HTTPS") == "on" else 'http'` à `return 'http'`. (ccff08c1, Fixed [#24463](#) -- Removed mod_python functionality from HttpRequest._get_scheme()).

#5 - 04 décembre 2017 11:44 - Benjamin Dauvergne

Ack.

#6 - 04 décembre 2017 11:49 - Frédéric Péters

- Statut changé de En cours à Résolu (à déployer)

```
commit 62e597e15761553841b34b604410764201dc6a96
Author: Frédéric Péters <fpeters@entrouvert.com>
Date: Sun Dec 3 13:11:30 2017 +0100
```

```
debian: set SECURE_PROXY_SSL_HEADER for nginx/runserver integration (#16863)
```

#7 - 23 décembre 2018 16:00 - Frédéric Péters

- Statut changé de Résolu (à déployer) à Solution déployée

Fichiers

0001-debian-set-SECURE_PROXY_SSL_HEADER-for-nginx-runserv.patch 03 décembre 2017

Frédéric Péters