Passerelle - Bug #20789

Ajouter un support général de ca_bundle et certif clients

18 décembre 2017 17:01 - Thomas Noël

Statut:	Fermé	**Début:**	18 décembre 2017
Priorité:	Normal	**Echéance:**	
Assigné à:	Thomas Noël	**% réalisé:**	0%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	

Description

ca_bundle nécessaire au moins sur le déploiement solis/cd80

Révisions associées

Révision f6a14c37 - 19 décembre 2017 10:22 - Thomas Noël

utils: rename utils.LoggedRequest as utils.Request (#20789)

... because it will not only used for logging.

Révision cb5000d5 - 19 décembre 2017 14:48 - Thomas Noël

utils.Request: use auth, certs and proxy from resource, if available (#20789)

Révision b5452c2d - 19 décembre 2017 16:17 - Thomas Noël

solis: add http auth, ssl and proxy attributes (#20789)

Historique

#1 - 18 décembre 2017 17:02 - Thomas Noël

- Fichier 0001-solis-add-ca-bundle-and-client-certificate-support-2.patch ajouté

- Statut changé de Nouveau à En cours

- Patch proposed changé de Non à Oui

#3 - 18 décembre 2017 19:09 - Frédéric Péters

Ça m'irait bien qu'on standardise cette méthode requests_options au niveau de la classe de base (et avec des getattr parce que quand même pas pour ajouter ca_bundle et keystore partout).

Et en fait la partie ca_bundle, ça m'irait bien que ça ne se retrouve pas partout, qu'on puisse gérer ça dans des options globales à Passerelle.

Bref, si on peut se donner ici ce temps, c'est cool, sinon je créerai des tickets pour évoluer vers ça.

#4 - 18 décembre 2017 21:13 - Benjamin Dauvergne

Mixin !! Enfin plutôt un modèle abstrait ResourceWithCertificates :)

#5 - 18 décembre 2017 23:22 - Thomas Noël

- Sujet changé de solis: ajouter le support de ca bundle et certif clients à Ajouter un support général de ca_bundle et certif clients

- Description mis à jour

#6 - 18 décembre 2017 23:26 - Thomas Noël

Frédéric Péters a écrit :

> Ça m'irait bien qu'on standardise cette méthode requests_options au niveau de la classe de base (et avec des getattr parce que quand même pas pour ajouter ca_bundle et keystore partout).

Yep. Je prépare ça.

> Et en fait la partie ca_bundle, ça m'irait bien que ça ne se retrouve pas partout, qu'on puisse gérer ça dans des options globales à Passerelle.

Nope, car il s'agit ici de dire "cette instance de connecteur accepte que le certif du serveur en face provienne de telle CA (et pas d'une autre)". L'idée est donc de permettre une authentification plus forte du serveur en face, et pas seulement d'ajouter des CA à un bundle système par défaut qu'on ne connait pas. C'est bien dans la config de l'instance du connecteur que ça a lieu, à côté de la définition d'un éventuel certif client.

> Mixin !! Enfin plutôt un modèle abstrait ResourceWithCertificates :)

Ouaip.

#7 - 18 décembre 2017 23:47 - Frédéric Péters

> Nope, car il s'agit ici de dire "cette instance de connecteur accepte que le certif du serveur en face provienne de telle CA (et pas d'une autre)".

J'ai l'impression que le nom n'est pas terrible; exprimé comme ici le souhait me semble plutôt relever du certifcate pinning. (...) et que la réalité au final serait un mix des deux parce qu'en face on se trouve avec un certicat signé par une autorité non reconnue / autosigné mais on veut s'assurer qu'il ne change pas. (je n'ai pas cherché l'adresse du serveur qu'on a en face ici).

(et question vocabulaire la documentation de python-requests qui propose juste "path to a CA bundle to use" n'aide pas).

Et à continuer à réfléchir sur une situation dont je n'ai pas le détail précis, je me mets à imaginer les possibilités suivantes, sans exposer cette option,

- si certificat pas reconnu
  - affichage d'un message proposant d'en voir les détails, de l'accepter pour l'avenir (comme un navigateur, comme un client ssh).
- si certificat reconnu
  - ajout dans un menu d'une action "pin certificate" qui limiterait à l'avenir les connexions au certificat présent.

Tout ça nous emmène sans doute trop loin.

> > Mixin...
>
> Ouaip.

Perso avoir une grosse classe de base me va plutôt mieux.

#8 - 19 décembre 2017 00:21 - Thomas Noël

Frédéric Péters a écrit :

> > Nope, car il s'agit ici de dire "cette instance de connecteur accepte que le certif du serveur en face provienne de telle CA (et pas d'une autre)".
>
> J'ai l'impression que le nom n'est pas terrible; exprimé comme ici le souhait me semble plutôt relever du certifcate pinning. (...) et que la réalité au final serait un mix des deux parce qu'en face on se trouve avec un certicat signé par une autorité non reconnue / autosigné mais on veut s'assurer qu'il ne change pas. (je n'ai pas cherché l'adresse du serveur qu'on a en face ici).

Pour le CD80 on a en face un serveur avec un certificat de leur PKI. On veut donc vérifier la requête se fait sur une machine signée par eux, il suffit pour ça de faire un «verify=resource.ca_bundle.path» avec le bundle qu'ils nous ont fourni. Que le certif ne change pas n'est pas un enjeu, au contraire (ça expire, un certif) ; on veut juste vérifier qu'il est certifié par la collectivité.

> documentation de python-requests qui propose juste "path to a CA bundle to use" n'aide pas

(perso, ca_bundle, je comprends immédiatement de quoi il s'agit)

> Tout ça nous emmène sans doute trop loin.

Yep.

> > > Mixin...
>
> > Ouaip.
>
> Perso avoir une grosse classe de base me va plutôt mieux.

Je voyaix les mixin pour ajouter automatiquement les champs username/password, keystore ou ca_bundle dans une resource, sans avoir besoin de déclarer autre chose que ces classes à côté du BaseResource.

#9 - 19 décembre 2017 01:36 - Thomas Noël

- Fichier 0001-utils-rename-utils.LoggedRequest-as-utils.Request-20.patch ajouté

- Fichier 0002-utils.Request-use-auth-certs-and-proxy-from-resource.patch ajouté

0001 : renomme utils.LoggedRequest en utils.Request parce qu'on va dépasser le simple logging.
0002 : gestion de auth, verify, cert et proxies s'ils sont disponibles dans la resource respectivement sous la forme de champs username/password, ca_bundle ou verify_cert, keystore et proxy.

Manque 0003 pour des mixin dans base.models, pour rendre ça facile à utiliser dans les futures ressources où on voudra ajouter ces possibilités.

#10 - 19 décembre 2017 03:34 - Thomas Noël

- Fichier 0003-add-SecureMixin-with-auth-certs-and-proxy-fields-207.patch ajouté

- Fichier 0004-solis-use-SecureMixin-system-20789.patch ajouté

- Fichier Screenshot-2017-12-19 Passerelle.png ajouté

0003 : voilà le mixin, que j'ai nommé "SecureMixin" parce que j'ai pas d'autre idée.
0004 : et utilisation dans Solis

Mais pépin que j'ai pas encore cherché à résoudre : au niveau du manage, les paramètres de sécu se posent avant les paramètres de la ressource, on se retrouve par exemple pour Solis avec l'URL à indiquer tout en bas... Zut. Copie d'écran jointe où on voit l'URL demandée tout en bas...

#11 - 19 décembre 2017 08:28 - Frédéric Péters

> Copie d'écran ...

Qui m'éclaire sur le nom (ou c'est juste le matin), l'attribut devrait s'appeler trusted_certificate_authorities.

```
# - use login/pass authentication if resource.username and resource.password exist
# - use client side certificate if resource.keystore exists
# - use server certificate CA verification if resource.ca_bundle exists
# - disable CA verification if resource.verify exists
```

... exists and is set; dirais-je.

```
    logger = logging.getLogger('logged_requests')
```

Virer la partie logged_, vu le 0001.

> 0003_...

Et sur l'idée d'un mixin, paradoxalement trop de choses réunies; le travail dans 0002 qui fait un gros "Requests" me va bien, mais pour moi il pouvait s'accompagner de simples modifications par convention dans les classes des connecteurs. Ou alors diviser, HttpAuthMixin, ProxyMixin, SslMixin, etc. Et sur l'ordre des champs, mettre la classe Mixin après la classe BaseRessource ?

#12 - 19 décembre 2017 11:15 - Thomas Noël

- Fichier 0004-solis-use-SecureMixin-system-20789.patch ajouté

- Fichier 0003-add-SecureMixin-with-auth-certs-and-proxy-fields-207.patch ajouté

- Fichier 0002-utils.Request-use-auth-certs-and-proxy-from-resource.patch ajouté

- Fichier 0001-utils-rename-utils.LoggedRequest-as-utils.Request-20.patch ajouté

Frédéric Péters a écrit :

> > Copie d'écran ...
>
> Qui m'éclaire sur le nom (ou c'est juste le matin), l'attribut devrait s'appeler trusted_certificate_authorities.

Tout à fait.

> ... exists and is set; dirais-je.

Yep.

> Virer la partie logged_, vu le 0001.

Yep.

> > 0003_...

> Et sur l'idée d'un mixin, paradoxalement trop de choses réunies; le travail dans 0002 qui fait un gros "Requests" me va bien, mais pour moi il pouvait s'accompagner de simples modifications par convention dans les classes des connecteurs. Ou alors diviser, HttpAuthMixin, ProxyMixin, SslMixin, etc. Et sur l'ordre des champs, mettre la classe Mixin après la classe BaseRessource ?

J'ai divisé en plusieurs Mixin.

Et poser ces mixin après ou avant BaseResource ne change rien... cet ordre des champs, je ne sais pas d'où il vient.

#13 - 19 décembre 2017 11:33 - Benjamin Dauvergne

Est-ce qu'on déporterait pas le changement de comportement de request dans les mixin ? Genre avoir une méthode get_requests_kwargs(**kwargs) comme get_form_kwargs(**kwargs) dans les vues génériques Django ?

Bon après ce qui me gêne ça va être les cas ou username/password ne doit pas être utilisé en HTTP Basic, mais bon on verra le jour où ça arrive pour rendre le comportement débrayable (ça casse rien d'avoir un entête HTTP Basic qui ne sert pas, ce n'est juste pas bien propre).

#14 - 19 décembre 2017 15:59 - Thomas Noël

- Fichier 0003-solis-add-http-auth-ssl-and-proxy-attributes-20789.patch ajouté

- Fichier 0002-utils.Request-use-auth-certs-and-proxy-from-resource.patch ajouté

- Fichier 0001-utils-rename-utils.LoggedRequest-as-utils.Request-20.patch ajouté

> ça va être les cas ou username/password ne doit pas être utilisé en HTTP Basic

Yep, j'ai revu l'affaire dans la série jointe. J'ai nommé les attributs d'une BaseResource de façon plus explicite :

```
    basic_auth_username = models.CharField(max_length=128, blank=True,
                            verbose_name=_('HTTP Basic Auth username'))
    basic_auth_password = models.CharField(max_length=128, blank=True,
                            verbose_name=_('HTTP Basic Auth password'))

    client_certificate = models.FileField(upload_to=keystore_upload_to,
                            null=True, blank=True,
                            verbose_name=_('Client certificate'),
                            help_text=_('Client certificate and private key (PEM format)'))
    verify_cert = models.BooleanField(default=True,
                            verbose_name=_('Check HTTPS Certificate validity'))
    trusted_certificate_authorities = models.FileField(upload_to=trusted_cas_upload_to,
                            null=True, blank=True,
                            verbose_name=_('Trusted CAs'),
                            help_text=_('Trusted CAs (PEM format)'))

    http_proxy = models.CharField(max_length=128, blank=True,
                            verbose_name=_('Proxy URL'))
```

J'ai pour l'instant retiré l'affaire des Mixin, inutilisables à cause du formulaire désordonné que ça donne pour ajouter un connecteur. Pas de drame, ces mixin pourront revenir quand ce détail sera réglé, ça changera rien, même pas de migrations.

#15 - 19 décembre 2017 16:02 - Frédéric Péters

> ... de façon plus explicite

Bien mais manque alors la migration username → basic_auth_username / password → basic_auth_password.

#16 - 19 décembre 2017 16:17 - Thomas Noël

- Fichier 0003-solis-add-http-auth-ssl-and-proxy-attributes-20789.patch ajouté

Ah oui, git add oublié, voici.

#17 - 20 décembre 2017 08:09 - Frédéric Péters

Ok.

#18 - 20 décembre 2017 09:50 - Thomas Noël

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

```
commit b5452c2d536883ac29d052d0702c475911af5ac8
Author: Thomas NOEL <tnoel@entrouvert.com>
Date:   Tue Dec 19 03:25:30 2017 +0100

    solis: add http auth, ssl and proxy attributes (#20789)

commit cb5000d54dd23eb744c05282b5c8116f0b3bb9c7
Author: Thomas NOEL <tnoel@entrouvert.com>
Date:   Tue Dec 19 01:32:24 2017 +0100

    utils.Request: use auth, certs and proxy from resource, if available (#20789)

commit f6a14c375b0dbc560538889b183d098a760828da
Author: Thomas NOEL <tnoel@entrouvert.com>
Date:   Mon Dec 18 23:53:35 2017 +0100

    utils: rename utils.LoggedRequest as utils.Request (#20789)

    ... because it will not only used for logging.
```

#19 - 04 août 2018 12:30 - Benjamin Dauvergne

- Statut changé de Résolu (à déployer) à Fermé

Fichiers

0001-solis-add-ca-bundle-and-client-certificate-support-2.patch	4,19 ko	18 décembre 2017	Thomas Noël
0002-utils.Request-use-auth-certs-and-proxy-from-resource.patch	8,15 ko	19 décembre 2017	Thomas Noël
0001-utils-rename-utils.LoggedRequest-as-utils.Request-20.patch	34,2 ko	19 décembre 2017	Thomas Noël
0003-add-SecureMixin-with-auth-certs-and-proxy-fields-207.patch	1,82 ko	19 décembre 2017	Thomas Noël
0004-solis-use-SecureMixin-system-20789.patch	6,89 ko	19 décembre 2017	Thomas Noël
Screenshot-2017-12-19 Passerelle.png	22,5 ko	19 décembre 2017	Thomas Noël
0004-solis-use-SecureMixin-system-20789.patch	7,1 ko	19 décembre 2017	Thomas Noël
0003-add-SecureMixin-with-auth-certs-and-proxy-fields-207.patch	2,01 ko	19 décembre 2017	Thomas Noël
0002-utils.Request-use-auth-certs-and-proxy-from-resource.patch	8,22 ko	19 décembre 2017	Thomas Noël
0001-utils-rename-utils.LoggedRequest-as-utils.Request-20.patch	36 ko	19 décembre 2017	Thomas Noël
0003-solis-add-http-auth-ssl-and-proxy-attributes-20789.patch	7,74 ko	19 décembre 2017	Thomas Noël
0002-utils.Request-use-auth-certs-and-proxy-from-resource.patch	8,43 ko	19 décembre 2017	Thomas Noël
0001-utils-rename-utils.LoggedRequest-as-utils.Request-20.patch	36 ko	19 décembre 2017	Thomas Noël
0003-solis-add-http-auth-ssl-and-proxy-attributes-20789.patch	10,2 ko	19 décembre 2017	Thomas Noël