Hobo - Bug #8580

orig et secret dans KNOWN_SERVICES

09 octobre 2015 11:45 - Thomas Noël

Statut:	Fermé	**Début:**	09 octobre 2015
Priorité:	Haut	**Echéance:**	
Assigné à:	Benjamin Dauvergne	**% réalisé:**	100%
Catégorie:		**Temps estimé:**	0:00 heure
Version cible:			
Patch proposed:	Oui	**Planning:**	

Description

Actuellement le settings.KNOWN_SERVICES généré par hobo a la tête suivante :

```
#
# exemple pour un combo (this=true) qui connait deux wcs :
#
KNOWN_SERVICES = {
    'combo: {
        'default': {'title': 'test', 'url': 'http://127.0.0.1:8999/',
            'secret': '12345', 'orig': 'combo', 'this': true },
     },
    'wcs': {
        'default': {'title': 'test', 'url': 'http://127.0.0.1:8999/',
            'secret': 'abcdef', 'orig': 'combo'},
        'other': {'title': 'test2', 'url': 'http://127.0.0.2:8999/',
            'secret': 'x0x0x0', 'orig': 'combo'},
    }
}
```

Autrement dit, orig/secret est le couple à utiliser pour **émettre** une requête signée vers le site cible. Le orig est toujours le même, le nom du service local... Soit.

Maintenant, il faudrait l'autre partie, celle qui permet de vérifier la signature d'un requête qui arrive. Est-ce qu'on considère qu'on utilise le même secret dans les deux sens (secret partagé) ? Si oui, il manque encore le "orig" de provenance... Et donc, est-ce qu'on devrait pas plutôt utiliser le "orig" pour cela?

Donc avoir plutôt :

```
KNOWN_SERVICES = {
    'combo: {
        'default': {'title': 'test', 'url': 'http://127.0.0.1:8999/',
            'secret': '---', 'orig': 'combo', 'this': true },
     },
    'wcs': {
        'default': {'title': 'test', 'url': 'http://127.0.0.1:8999/',
            'secret': 'abcdef', 'orig': 'wcs1'},
        'other': {'title': 'test2', 'url': 'http://127.0.0.2:8999/',
            'secret': 'x0x0x0', 'orig': 'wcs2'},
    }
}
```

Et sur wcs1 par exemple :

```
KNOWN_SERVICES = {
    'combo: {
        'default': {'title': 'test', 'url': 'http://127.0.0.1:8999/',
            'secret': 'abcdef', 'orig': 'combo', },
     },
    'wcs': {
        'default': {'title': 'test', 'url': 'http://127.0.0.1:8999/',
            'secret': '---', 'orig': 'wcs1', 'this': true },
        'other': {'title': 'test2', 'url': 'http://127.0.0.2:8999/',
```

```
            'secret': '987123', 'orig': 'wcs2'},
    }
}
```

Pour émettre un appel :

- on prend le orig qui est dans le service "this"
- on signe avec le secret du service cible

Pour vérifier un appel reçu :

- on cherche le orig dans la liste
- on vérifie que c'est bien signé avec le secret partagé

Ok ?

Demandes liées:

Lié à w.c.s. - Development #8961: Dans check_hobo générer les secrets avec sh...	**Fermé**	**12 novembre 2015**
Lié à Combo - Bug #9351: Terminer la notification de paiement à wcs (cas local)	**Fermé**	**16 décembre 2015**
Bloque Hobo - Development #8896: Avoir une classe d'authentificatoin django-r...	**Fermé**	**06 novembre 2015**

Révisions associées

Révision 4359db0c - 26 janvier 2016 03:11 - Benjamin Dauvergne

tests_multitenant: restore default settings after modifying them (#8580)

Révision f36985e2 - 26 janvier 2016 03:11 - Serghei Mihai

settings_loaders: compute a symmetric shared secret for services (#8580)

Secret is computed by appying xor to the SHA-256 hash of each service
secret, making it insensible to the ordering of the two secrets.

Tests of the basic properties of the shared_secret() function are added.

Révision c85d548f - 26 janvier 2016 03:11 - Benjamin Dauvergne

passerelle: use shared_secret for ApiUser.key (fixes #8580)

Historique

#1 - 09 octobre 2015 12:12 - Benjamin Dauvergne

De toute façon on fait de la signature HMAC donc secret partagé, il n'y a pas de question sur ce point. Idéalement il faudrait des secrets point à point, i.e le secret partagé entre w.c.s et passerelle n'as pas à être le même secret que entre combo et passerelle. Pour ça il faudrait que hobo soit un peu plus sioux et distribue dans le hobo_deploy destiné aux différentes applications des secret key différentes calculé via HASH(know_service.secret+deployed_service.secret). Comme quand on retire un service, il ne peut quand même pas connaître les clés des autres. Ce qui est bien c'est que c'est déjà directionnel on ne fait pas vraiment du broadcase parce qu'en plus du hobo.json on indique via le base_url quel tenant est visé par le message.

#2 - 09 octobre 2015 13:44 - Benjamin Dauvergne

Thomas Noël a écrit :

> Actuellement le settings.KNOWN_SERVICES généré par hobo a la tête suivante :

Actuellement le secret calculé dans KNOWN_SERVICES est calculé ainsi: SHA1(orig du service local + secret du service local), à partir du hobo.json tous les services peuvent le calculer et vont calculer la même chose, donc ça ne donne aucun avantage particulier par rapport a utiliser "secret du service local".

Le code est relativement affreux:

```
        base_url, secret = [(s.get('base_url'), s.get('secret_key'))¶
                            for s in services if s.get('this')][0]¶
```

Si tu choisis dans passerelle de faire ce que tu dis, i.e. de vérifier les signatures des appels avec le secret du service cible, alors tous les APIUser auront la même clé (même service cible == passerelle => même clé). Ça me parait plus logique d'utiliser la clé du service source (l'appelant). Et donc dans KNOWN_SERVICES on recopie juste la clé de chaque service dans le champ 'secret', en fait on peut simplement mettre le hobo_json['services'] dans KNOWN_SERVICES et tout deviendra plus simple. Pour connaître son propre orig, un connection.tenant.get_base_url() suffit pas besoin de passer par KNOWN_SERVICES.

#3 - 09 octobre 2015 13:48 - Benjamin Dauvergne

Le souci avec ça c'est qu'on leak les secrets dans le backoffice de Passerelle si il les affiche dans l'administration des APIUser. C'est pour cela que dans ce cas ça peut-être intéressant de calculer la clé par SHA1(clé locale+clé distante) our encore SHA1(base_url locale+base_url distante+clé locale+clé distante), ça rend le secret P2P, donc si je vole le secret du APIUser combo dans passerelle je ne peux pas appeller les web-services de w.c.s avec (ce n'est pas le même).

#4 - 09 octobre 2015 14:05 - Thomas Noël

Benjamin Dauvergne a écrit :

> Le souci avec ça c'est qu'on leak les secrets dans le backoffice de Passerelle si il les affiche dans l'administration des APIUser. C'est pour cela que dans ce cas ça peut-être intéressant de calculer la clé par SHA1(clé locale+clé distante) our encore SHA1(base_url locale+base_url distante+clé locale+clé distante), ça rend le secret P2P, donc si je vole le secret du APIUser combo dans passerelle je ne peux pas appeller les web-services de w.c.s avec (ce n'est pas le même).

Pour passerelle on peut voir autrement : https://dev.entrouvert.org/issues/8551

#5 - 09 octobre 2015 14:07 - Thomas Noël

Benjamin Dauvergne a écrit :

> Thomas Noël a écrit :
>
> > Actuellement le settings.KNOWN_SERVICES généré par hobo a la tête suivante :
>
> Actuellement le secret calculé dans KNOWN_SERVICES est calculé ainsi: SHA1(orig du service local + secret du service local), à partir du hobo.json tous les services peuvent le calculer et vont calculer la même chose, donc ça ne donne aucun avantage particulier par rapport a utiliser "secret du service local".
> Le code est relativement affreux:
> [...]
> Si tu choisis dans passerelle de faire ce que tu dis (...)

Mais ce que je propose c'est aussi de revoir le contenu de KNOWN_SERVICES, et donc le code, qui ne va pas.

#6 - 09 octobre 2015 14:34 - Benjamin Dauvergne

Thomas Noël a écrit :

> Benjamin Dauvergne a écrit :
>
> > Le souci avec ça c'est qu'on leak les secrets dans le backoffice de Passerelle si il les affiche dans l'administration des APIUser. C'est pour cela que dans ce cas ça peut-être intéressant de calculer la clé par SHA1(clé locale+clé distante) our encore SHA1(base_url locale+base_url distante+clé locale+clé distante), ça rend le secret P2P, donc si je vole le secret du APIUser combo dans passerelle je ne peux pas appeller les web-services de w.c.s avec (ce n'est pas le même).
>
> Pour passerelle on peut voir autrement : https://dev.entrouvert.org/issues/8551

Ok ça élimine ce souci là.

#7 - 09 octobre 2015 14:48 - Benjamin Dauvergne

Thomas Noël a écrit :

> Mais ce que je propose c'est aussi de revoir le contenu de KNOWN_SERVICES, et donc le code, qui ne va pas.

Merci d'expliquer ce que tu proposes pour le contenu de KNOWN_SERVICES parce que je ne comprends pas les exemples.

#8 - 09 octobre 2015 14:55 - Serghei Mihai

- Fichier 0001-compute-service-api-key-from-its-orig-and-destinatio.patch ajouté

- Patch proposed changé de Non à Oui

Un début de patch...

#9 - 09 octobre 2015 15:07 - Thomas Noël

Benjamin Dauvergne a écrit :

> Thomas Noël a écrit :
>
> > Mais ce que je propose c'est aussi de revoir le contenu de KNOWN_SERVICES, et donc le code, qui ne va pas.
>
> Merci d'expliquer ce que tu proposes pour le contenu de KNOWN_SERVICES parce que je ne comprends pas les exemples.

Il s'agit des deux derniers blocs KNOWN_SERVICES dans la description du ticket.

Le premier est celui qui serait vu par un tenant **combo**:

```
KNOWN_SERVICES = {
    'combo: {
        'default': {'title': 'test', 'url': 'http://127.0.0.1:8999/',      'secret': '---',    'orig': 'combo'
,  'this': true },
     },
    'wcs': {
        'default': {'title': 'test', 'url': 'http://127.0.0.1:8999/',      'secret': 'abcdef', 'orig': 'wcs1'
 },
        'other':  {'title': 'test2', 'url': 'http://127.0.0.2:8999/',      'secret': 'x0x0x0', 'orig': 'wcs2'
 },
    }
}
```

Et le second par un tenant **wcs1** :

```
KNOWN_SERVICES = {
    'combo: {
        'default': {'title': 'test',  'url': 'http://127.0.0.1:8999/',     'secret': 'abcdef', 'orig': 'combo'
, },
     },
    'wcs': {
        'default': {'title': 'test',  'url': 'http://127.0.0.1:8999/',     'secret': '---',    'orig': 'wcs1',
     'this': true },
        'other':   {'title': 'test2', 'url': 'http://127.0.0.2:8999/',     'secret': '987123', 'orig': 'wcs2'}
,
    }
}
```

Quand combo interroge wcs1 : ?orig=combo & signature=(secret = abcdef)

- combo trouve l'orig dans le this=true et la clé dans le service wcs1
- wcs trouve la clé dans le orig = combo

Voili voilà.

#10 - 09 octobre 2015 15:13 - Benjamin Dauvergne

- Fichier 0001-compute-service-api-key-from-its-orig-and-destinatio.patch ajouté

Problème de symétrie que j'oubliais avec les fonctions de hash, une autre proposition en utilisant XOR qui est commutatif.

#11 - 09 octobre 2015 15:13 - Benjamin Dauvergne

Thomas Noël a écrit :

> Benjamin Dauvergne a écrit :
>
> > Thomas Noël a écrit :
> >
> > > Mais ce que je propose c'est aussi de revoir le contenu de KNOWN_SERVICES, et donc le code, qui ne va pas.
> >
> > Merci d'expliquer ce que tu proposes pour le contenu de KNOWN_SERVICES parce que je ne comprends pas les exemples.
>
> Il s'agit des deux derniers blocs KNOWN_SERVICES dans la description du ticket.
>
> Le premier est celui qui serait vu par un tenant **combo**:
> [...]
>
> Et le second par un tenant **wcs1** :

> [...]
>
> Quand combo interroge wcs1 : ?orig=combo & signature=(secret = abcdef)
>
> - combo trouve l'orig dans le this=true et la clé dans le service wcs1
> - wcs trouve la clé dans le orig = combo
>
> Voili voilà.

Trop compliqué.

#12 - 09 octobre 2015 15:23 - Serghei Mihai

Benjamin Dauvergne a écrit :

> Problème de symétrie que j'oubliais avec les fonctions de hash, une autre proposition en utilisant XOR qui est commutatif.

...int(secret, 16)...
pour des secret de type:

```
  "secret_key": "4_*kksd8=nl*rk-v22%mohzhr9!9t^f#*#(*=*rhh5*-sfn88@",
  ...
  "secret_key": "&l$wjhw@%feis^(#(5jbss#s%w#b)mug(hw#=(tc7^re7v1l(_",
```

ne fonctionnera pas

#13 - 09 octobre 2015 15:47 - Benjamin Dauvergne

- Fichier 0001-compute-service-api-key-from-its-orig-and-destinatio.patch ajouté

Une version qui fonctionne quelque soit le format des secrets.

#14 - 10 octobre 2015 23:43 - Serghei Mihai

Ça me semble bon

#15 - 11 octobre 2015 16:31 - Thomas Noël

Ok pour signer quand on envoie... mais quid de la vérification à la reception avec ces infos ? ("orig" reste invariant tout le long).

#16 - 11 octobre 2015 23:16 - Benjamin Dauvergne

Thomas Noël a écrit :

> Ok pour signer quand on envoie... mais quid de la vérification à la reception avec ces infos ? ("orig" reste invariant tout le long).

C'est pour ça que je ne hash pas orig dans ma clé. Ou bien je n'ai pas compris ?

#17 - 06 novembre 2015 11:31 - Benjamin Dauvergne

Bon on va reprendre du début parce que le code qui génère KNOWN_SERVICES c'est n'importe quoi, dans orig on a le domaine de l'application locale, c'est bien pour créer une requête mais pour vérifier celles qu'on reçoit ça ne sert à rien.

On va déjà énumérer les bouts de code qui utilisent KNOWN_SERVICES ou qui copie son fonctionnement pour émettre ou recevoir des requêtes:

- wcs qui génère ses api-secrets en suivant le schéma défini dans KNOWN_SERVICES sha1(orig de l'émetteur + secret de l'émetteur)
- welco:

```
welco/utils.py:124:    url += 'orig=%s' % wcs_site.get('orig')
welco/utils.py:153:    params['orig'] = wcs_site.get('orig')
```

Ceux qui n'utilisent pas KNOWN_SERVICES et font n'importe quoi:

- hobo/agent/passerelle/management/commands/hobo_deploy.py: service['secret_key'] est directement copié dans ApiUser.key c'est incompatible avec le schéma prévu par KNOWN_SERVICES

Ceux qui n'utilisent pas KNOWN_SERVICES et font ce qu'ils peuvent:

- wcs/wf/wscall.py: c'est configuré à la main à chaque fois

Plan d'action:

- changer la génération de secret pour ce que je propose, i.e. pour deux services donnés on obtien toujours la même clé (via XOR entre les clés ou n'importe quoi de commutatif ou en utilisant une même clé pour toute la plateforme)
- ajouter une clé verif_orig qui sert à trouver la clé pour vérifier une requête reçue (je laisse orig et sa sémantique actuelle pour rétrocompatibilité)
- corriger hobo_deploy dans passerelle en conséquence et penser à refaire un hobo-deploy --ignore-timestamp sur tous les passerelle
- modifier wcs/check_hobos en conséquence et refaire un check_hobos --ignore-timestamp
- pour wscall on ne fait rien pour l'instant on verra quand on voudra proposer des web-services pré-configurés depuis ce que passerelle publie

#18 - 06 novembre 2015 13:58 - Benjamin Dauvergne

Coté authentic ça passe par le #8896 qu'on ne pourra résoudre qu'après celui ci.

#19 - 06 novembre 2015 13:58 - Benjamin Dauvergne

- Bloque Development #8896: Avoir une classe d'authentificatoin django-rest-framework pour authentic2 compatible avec les signatures d'URL dans Publik ajouté

#20 - 12 novembre 2015 10:59 - Benjamin Dauvergne

- Assigné à mis à Benjamin Dauvergne

- Priorité changé de Normal à Haut

Je m'en occupe, en suivant mon plan.

#21 - 12 novembre 2015 13:05 - Benjamin Dauvergne

- Lié à Development #8961: Dans check_hobo générer les secrets avec shared_secret() ajouté

#22 - 12 novembre 2015 13:12 - Benjamin Dauvergne

- Fichier 0001-tests_multitenant-restore-default-settings-after-mod.patch ajouté

- Fichier 0002-settings_loaders-compute-a-symetric-shared-secret-fo.patch ajouté

- Fichier 0003-passerelle-use-shared_secret-for-ApiUser.key-fixes-8.patch ajouté

#23 - 12 novembre 2015 13:36 - Benjamin Dauvergne

- Fichier 0002-settings_loaders-compute-a-symetric-shared-secret-fo.patch supprimé

#24 - 12 novembre 2015 13:36 - Benjamin Dauvergne

- Fichier 0002-settings_loaders-compute-a-symetric-shared-secret-fo.patch ajouté

Correction au patch 2.

#25 - 13 novembre 2015 11:22 - Benjamin Dauvergne

- Statut changé de Nouveau à En cours

#26 - 16 novembre 2015 13:37 - Frédéric Péters

Il y aura donc besoin de coordination pour les mises en recette/prod (wcs et hobo se trouvant à nouveau liés); avant d'avancer ici il faudrait du coup à mon sens voir pour avancer hobo et wcs en prod sur les dernières versions.

#27 - 30 novembre 2015 23:36 - Frédéric Péters

Je n'ai pas testé mais le commentaire de #8961 s'applique peut-être au code ajouté ici pour passerelle.

#28 - 01 décembre 2015 11:13 - Benjamin Dauvergne

Non ici c'est bon:

```
        for service in services:
            # Why refer to ourself ?
            if service.get('this'):
                continue
```

#29 - 16 décembre 2015 10:32 - Frédéric Péters

- Lié à Bug #9351: Terminer la notification de paiement à wcs (cas local) ajouté

#30 - 18 décembre 2015 10:15 - Frédéric Péters

- Fichier 0001-passerelle-use-shared_secret-for-ApiUser.key-fixes-8.patch ajouté

(actualisation du patch passerelle après rebase au-dessus de tox)

#31 - 18 décembre 2015 12:00 - Benjamin Dauvergne

Désole je l'avais fait aussi de mon coté hier je n'ai pas mis à jour le ticket, je remet la série de patch.

#32 - 18 décembre 2015 12:04 - Benjamin Dauvergne

- Fichier 0001-tests_multitenant-restore-default-settings-after-mod.patch ajouté

- Fichier 0002-settings_loaders-compute-a-symetric-shared-secret-fo.patch ajouté

- Fichier 0003-passerelle-use-shared_secret-for-ApiUser.key-fixes-8.patch ajouté

#33 - 29 décembre 2015 14:32 - Frédéric Péters

À tester davantage encore, ce bout de code pour zapper le service courant pose problème :

```
+            # Why refer to ourself ?
+            if service.get('this'):
+                continue
```

En effet, le portail agent il est utilisé pour connaître la liste de l'ensemble des services (__services.js, pour construire le menu publik) et ici, le service appelé est zappé et donc le portail agent ne se trouve pas présent dans le menu publik.

(et tant qu'à avoir à toucher à nouveau ces patchs, je viens de vérifier et il faut deux "m" à symmetric).

#34 - 30 décembre 2015 02:21 - Benjamin Dauvergne

- Fichier 0002-settings_loaders-compute-a-symmetric-shared-secret-f.patch ajouté

Diff

```
diff --git a/hobo/multitenant/settings_loaders.py b/hobo/multitenant/settings_loaders.py
index b6c2679..7c9ae01 100644
--- a/hobo/multitenant/settings_loaders.py
+++ b/hobo/multitenant/settings_loaders.py
@@ -52,14 +52,7 @@ class KnownServices(FileBaseSettingsLoader):
         secret = this['secret_key']

         for service in services:
-            # Why refer to ourself ?
-            if service.get('this'):
-                continue
             service_id = service.get('service-id')
-            # compute a symetric shared secret using XOR
-            # secrets MUST be hexadecimal numbers of the same even length
-            shared_secret = (self.shared_secret(secret, service['secret_key']) if 'secret_key' in
-                             service else None)
             url = service.get('base_url')
             verif_orig = urlparse.urlparse(url).netloc.split(':')[0]
             service_data = {
@@ -68,9 +61,13 @@ class KnownServices(FileBaseSettingsLoader):
                 'title': service.get('title'),
                 'orig': orig,
                 'verif_orig': verif_orig,
-                'secret': shared_secret,
                 'variables': service.get('variables')
             }
+            # compute a symmetric shared secret using XOR
+            # secrets MUST be hexadecimal numbers of the same even length
+            if not service.get('this'):
+                service_data['secret_key'] = (self.shared_secret(secret, service['secret_key']) if
+                                              'secret_key' in service else None)
             if service_id in known_services:
                 known_services[service_id][service.get('slug')] = service_data
             else:
```

Et nouveau patch 2.

#35 - 05 janvier 2016 11:16 - Frédéric Péters

La clé doit être "secret", pas "secret_key".

#36 - 05 janvier 2016 11:25 - Benjamin Dauvergne

- Fichier 0002-settings_loaders-compute-a-symmetric-shared-secret-f.patch ajouté

Diff

```
diff --git a/hobo/multitenant/settings_loaders.py b/hobo/multitenant/settings_loaders.py
index 7c9ae01..8cee023 100644
--- a/hobo/multitenant/settings_loaders.py
+++ b/hobo/multitenant/settings_loaders.py
@@ -66,8 +66,8 @@ class KnownServices(FileBaseSettingsLoader):
             # compute a symmetric shared secret using XOR
             # secrets MUST be hexadecimal numbers of the same even length
             if not service.get('this'):
-                service_data['secret_key'] = (self.shared_secret(secret, service['secret_key']) if
-                                              'secret_key' in service else None)
+                service_data['secret'] = (self.shared_secret(secret, service['secret_key']) if
+                                          'secret_key' in service else None)
             if service_id in known_services:
                 known_services[service_id][service.get('slug')] = service_data
             else:
```

#37 - 18 janvier 2016 12:15 - Serghei Mihai

Ok

#38 - 18 janvier 2016 19:58 - Frédéric Péters

Attention, ça doit être coordonné avec le même code côté wcs, si ça se trouve poussé dans le dépôt de manière désordonnée ça rend les choses compliquées.

Selon moi, il faudrait une fenêtre de quelques jours calmes dans w.c.s., qu'en 0) on ai le trigger hobo redeploy en prod pour tout es les applis, en 1) le code des dépôts (hobo et wcs) mis en prod, en 2) ces modifications sur les secrets poussées, 3) en recette, 4) en prod. Qu'on reprenne ensuite le rythme normal des patchs.

#40 - 26 janvier 2016 03:15 - Benjamin Dauvergne

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

- % réalisé changé de 0 à 100

Appliqué par commit c85d548f443bfada84a17273e94387d591414940.

#41 - 04 février 2016 15:11 - Frédéric Péters

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

Fichiers

0001-compute-service-api-key-from-its-orig-and-destinatio.patch	1,37 ko	09 octobre 2015	Serghei Mihai
0001-compute-service-api-key-from-its-orig-and-destinatio.patch	1,94 ko	09 octobre 2015	Benjamin Dauvergne
0001-compute-service-api-key-from-its-orig-and-destinatio.patch	2,31 ko	09 octobre 2015	Benjamin Dauvergne
0001-tests_multitenant-restore-default-settings-after-mod.patch	9,44 ko	12 novembre 2015	Benjamin Dauvergne
0003-passerelle-use-shared_secret-for-ApiUser.key-fixes-8.patch	8,32 ko	12 novembre 2015	Benjamin Dauvergne
0002-settings_loaders-compute-a-symetric-shared-secret-fo.patch	6,04 ko	12 novembre 2015	Benjamin Dauvergne
0001-passerelle-use-shared_secret-for-ApiUser.key-fixes-8.patch	7,4 ko	18 décembre 2015	Frédéric Péters
0001-tests_multitenant-restore-default-settings-after-mod.patch	9,44 ko	18 décembre 2015	Benjamin Dauvergne
0003-passerelle-use-shared_secret-for-ApiUser.key-fixes-8.patch	7,77 ko	18 décembre 2015	Benjamin Dauvergne
0002-settings_loaders-compute-a-symetric-shared-secret-fo.patch	6,04 ko	18 décembre 2015	Benjamin Dauvergne
0002-settings_loaders-compute-a-symmetric-shared-secret-f.patch	6,06 ko	30 décembre 2015	Benjamin Dauvergne
0002-settings_loaders-compute-a-symmetric-shared-secret-f.patch	6,06 ko	05 janvier 2016	Benjamin Dauvergne