02 - Interfaçage Cyrus-imap avec CAS

Problématique

Le portail doit pouvoir proposer un canal permettant d'apporter des informations sur la messagerie de l'utilisateur (nombre de mails non lus, ...).
Il est probable que ce canal doive lancer un webmail externe (IMP ?) lorsque l'utilisateur désire utiliser sa messagerie d'une manière plus intensive.

Le portail comme le webmail doivent donc avoir un accès imap.

La difficulté : imap a besoin d'un nom d'utilisateur et d'un mot de passe, alors que dans un environnement CAS, l'authentification d'une personne se fait à l'aide d'un ticket ne transportant aucune information, et de plus, non re-jouable.

Comment faire ?

CAS en version 2, propose une fonctionnalité de proxy. Un proxy CAS (dans notre cas, le portail et le webmail) est capable de demander au serveur cas un ticket (Proxy Ticket, ou PT) permettant à un service tiers (dans notre cas, le serveur imap) d'authentifier une personne.
Pour que le serveur imap puisse authentifier une personne provenant d'un proxy cas, il faut donc qu'il soit capable de 'parler cas', donc d'implémenter le protocole CAS afin d'utiliser un PT comme un mot de passe.

Mais, dans le cas de la messagerie, il faut aussi que le serveur pop/imap sache authentifier par les moyens habituel (comptes LDAP, UNIX, ...) afin de permettre l'utilisation de clients de messagerie traditionnels.

Un autre soucis est d'optimiser le mécanisme. en effet, la plupart des webmails génèrent de très nombeuses connexions IMAP lors de leur utilisation. Il est difficilement concevable de 'dérouler' le mécanisme CAS pour chaque connexion (demande d'un PT au serveur CAS depuis le webmail, puis validation de ce PT par le service IMAP).

Ce document propose des solutions pour interfacer simplement CAS avec le serveur cyrus-imap, et les optimisations nécessaires.

Interfacer cyrus-imap avec CAS

Cyrus-imap s'appuie sur cyrus-sasl pour la partie authentification.
Cyrus-sasl est une librairie qui permet d'utiliser différents mécanismes d'authentification.
Parmi ceux-ci, il peut utiliser un 'démon' unix sachant parler différents protocoles ou méthodes, dont LDAP, pam, ... C'est salsauthd.

saslauthd présente beaucoup d'intérêts par rapport à notre problématique :

- Il est autonome. cyrus-sasl lui passe un identifiant et un mot de passe, saslauthd retour vrai ou faux. Il est donc possible d'intervenir sur ce démon sans avoir à toucher au code de cyrus-sasl, et donc de cyrus-imap.
- il sait utiliser pam (Pluggable Authentication Module).
- il sait mettre en cache les mots de passe utilisateurs (optionnel).
- Il permet différents modes d'authentification.

Les étapes sont les suivantes :

Coté pam

Il faut activer le pam_cas pour le service imap.

Voir : 01 - Installation et paramétrage de pam_cas.

Coté cyrus

- Compiler cyrus-sasl avec l'option *--with-saslauthd*.
- Dans le fichier /etc/imapd.conf, mettre *sasl_pwcheck_method: saslauthd*
- Pour que saslauthd utilise pam, il faut le lancer avecl'option *-a pam*
- pour optimiser le fonctionnement, saslauthd peut gérer un cache. C'est très utilise pour l'utilisation avec pam, engénéral.
  Pour activer le cache, il faut ajouter l'option *-c*
  D'autres options de gestion du cache (taille, ttl, ...) sont disponibles. Faire *man saslauthd* pour plus d'information.

A partir de ce moment, cyrus-imap sait utiliser l'interface pam pour authentifier les utilisateurs.

Cyrus-imap peut donc authentifier grâce au login de l'utilisateur et au PT. Mieux, comme pam permet d'enpiler plusieurs mécanismes d'authentification, le service imap peut continuer à fonctionner pour des clients de messagerie traditionnels.

En plus, un cache peut être activé, ce qui présente un grand intérêt. Le PT, qui est en principe un jeton non rejouable, peut le devenir pour cyrus-imap. Il devient en quelque sorte un identifiant de session spécifique, ce qui évite de recourir à tout le mécanisme d'acquisition puis de validation d'un PT à chaque requête.

Tests IMAP CAS-ifié

Un programme php utilisant phpCAS vous est fourni afin de tester la CAS-ification d'un serveur IMAP : casimap.php.

Il vous permet de décomposer les différentes étapes d'obtention de PGT, de PT, et de générer une connexion IMAP soit d'une manière traditionnelle, soit avec un PT non encore joué, soit avec le dernier PT joué qui seréait en cache IMAP.

Cet utilitaire vous permet donc de tester un IMAP CAS-ifié, et son comportement vis à vis du cache de mot de passe. Il fonctionne en mode proxy CAS.

Personnalisation du script

```
$cas_hostname="auth.univ.fr";         // c'est l'url du serveur CAS
   $cas_port=443;                        // le port d'accès https au serveur CA
   $cas_uri='';                          // l'uri de base du serveur CAS
   $imap_server="mail.univ.fr";          // le serveur imap à tester
   $imap_port=143;                       // le port imap
   $imap_box="INBOX";                    // le préfixe des boites IMAP
   $imap_service="imap://$imap_server";  // le service passé pour la récupération du PT. Doit être identique
                                         // au service paramétré dans pam-cas
```

Le 'service' passé au serveur CAS sera ici implicitement *imap://mail.univ.fr*. Si cela ne convient pas , écraser la variable *$imap_service* avec l'URL de service désiré.

Installation du script

Il faut bien sûr en préalable l'installation de phpCAS, avec ses pré-requis.

Ce script peut être accessible en http ; il **doit** impérativement pouvoir l'être également en https, puisque la connexion de callback du serveur CAS vers le script en vue du passage du PGT est en https.

On suppose que le script est accessible à l'URL http(s)://test/univ.fr/casimap.php

Paramétrage pam_cas

Le fichier pam_cas sera paramétré comme indiqué dans la doc pam_cas.

- paramètre de service (option -s) : doit contenir l'URL du service indiqué ci-dessus. Dans notre exemple :
  -simap://mail.univ.fr
- paramètres de proxy (option -p) : doit contenir l'url du (ou des) proxies CAS qui vont utiliser le service imap CAS-ifié. A priori, il y aura au moins le socle ENT, et le webmail. Pour le test, il faut rajouter notre script. Donc, pour suivre l'exemple :
  -phttps://test.univ-nancy2.fr/casimap.php

Essais

Ce script est composé de 2 parties : une dédié au mécanisme CAS, une autre à un login imap 'classique'.

Partie 'CAS'

Différentes options sont possibles :

- Obtenir un PGT : génère une redirection vers le serveur CAS afin d'obtenir un (nouveau) ST et PGT. C'est la première opération à faire pour tester le mécanisme CAS.
  Le PGT obtenu s'affiche à l'issue de cette opération.
  Il est possible de redemander successivement d'autres PGT.
- Obtenir un PT : Ce bouton n'est visible que si l'on dispose au préalable d'un PGT.
  Le PT obtenu s'affiche à l'issue de cette opération.
  Il est possible de redemander successivement d'autres PT.
- Ouvrir une connexion IMAP avec le PT Courant : test de la connexion IMAP avec le PT courant.
  Le résultat de la connexion IMAP s'affiche en tête du document W3 retourné.
  Il est possible de tester le cache IMAP en cliquant de nouvelles fois sur ce bouton sans réclamer de nouveaux PT.
- Logout applicatif sans CAS. Permet de supprimer la session applicative, sans se déloguer de CAS.
- Logout applicatif et CAS. Même chose que précédemment, avec un logout CAS.

Partie imap 'classique'

Cette seconde partie permet d'une part, de s'assurer que le script fonctionne correctement en imap 'classique', donc avec le login et mot de passe utilisateur, et d'autre part, d'apréhender l'écrasement du PT par le mot de passe utilisateur dans le cache IMAP.

Le résultat de la connexion IMAP s'affiche comme précédemment.

En cas de problème

- De récupération de PGT ou de PT : voir les logs d'accès et d'erreur du serveur CAS et du serveur apache supportant le script de test.
- De connexion imap : voir les logs d'accès et d'erreur du serveur CAS, et le fichier messages de la machine UNIX supportant le serveur IMAP CAS-ifié. Le code erreur des messages pam_cas donnent des indications sur le dysfonctionnement.

Si un PT est obtenu, mais que la connexion imap de fonctionne pas, redemander un nouveau PT.
En dessous du bouton "Ouvrir une connexion imap avec le PT courant", s'affiche une url du genre :

```
https://auth.univ.fr:443/proxyValidate?ticket=PT-168270-aG...&service=imap://mail.univ.fr
```

Tenter de valider 'à la main' ce PT :
Depuis un serveur ayant un acces https vers le serveur CAS (le serveur imap est un bon candidat), lancer :

```
wget --ca-certificate=/Cert/ac-racine.pem -O /tmp/cas.log "https://auth.univ.fr:443/proxyValidate?ticket=PT-
168270-aG...&service=imap://mail.univ.fr"
```

(mettre le contenu de l'URL entre 2 cotes).

Vous devez obtenir quelque chose comme celà dans le fichier /tmp/cas.log :

```
<cas:serviceResponse xmlns:cas='http://www.yale.edu/tp/cas'>
  <cas:authenticationSuccess>
    <cas:user>vmathieu</cas:user>
    <cas:proxies>
      <cas:proxy>https://infocri.univ-nancy2.fr/~vmathieu/CAS/casimap.php</cas:proxy>
    </cas:proxies>
  </cas:authenticationSuccess>
</cas:serviceResponse>
```

Rem : il est possible de remplacer l'option *--ca-certificate=/Cert/ac-racine.pem* par *--no-check-certificate* lors du wget ; dans ce cas, un warning sera affiché.

Difficultés coté client imap

Après cette mise en oeuvre, on dispose d'un service imap sachant parler CAS (en plus des autres méthodes), avec une possibilité de cache du PT.

Il subsiste une difficulté coté client (dans notre cas, le portail et le webmail).

En effet, ceux-ci sont porteurs d'un PT qui peut être considéré, dans ce contexte, comme un mot de passe volatile à durée réduite.

C'est d'autant plus vrai que le PT transmis par le portail est nécessairement différent du PT transmis par le webmail, tous deux étant différent du mot de passe transmis éventuellement par un client de messagerie traditionnel. saslauthd ne mémorise que le dernier 'mot de passe' valide.

Il faut donc que le portail comme le webmail sache réagir correctement lorsque l'authentification imap échoue : il doit pouvoir **une seule fois apès échec** redemander un PT auprès du serveur CAS afin de retenter la connexion.

Ca semble relativement aisé au niveau du portail. Il est probable que ce ne soit pas aussi simple au niveau IMP à la fois pour cet aspect, et pour le promouvoir en proxy CAS.

Optimisation poussée

La mise en oeuvre précédente s'est faite sans aucune modification de code à tous niveaux.

Un patch modifiant le fonctionnement du cache du démon saslauthd est proposé, permettant d'associer plusieurs 'mots de passe' à un utilisateur.

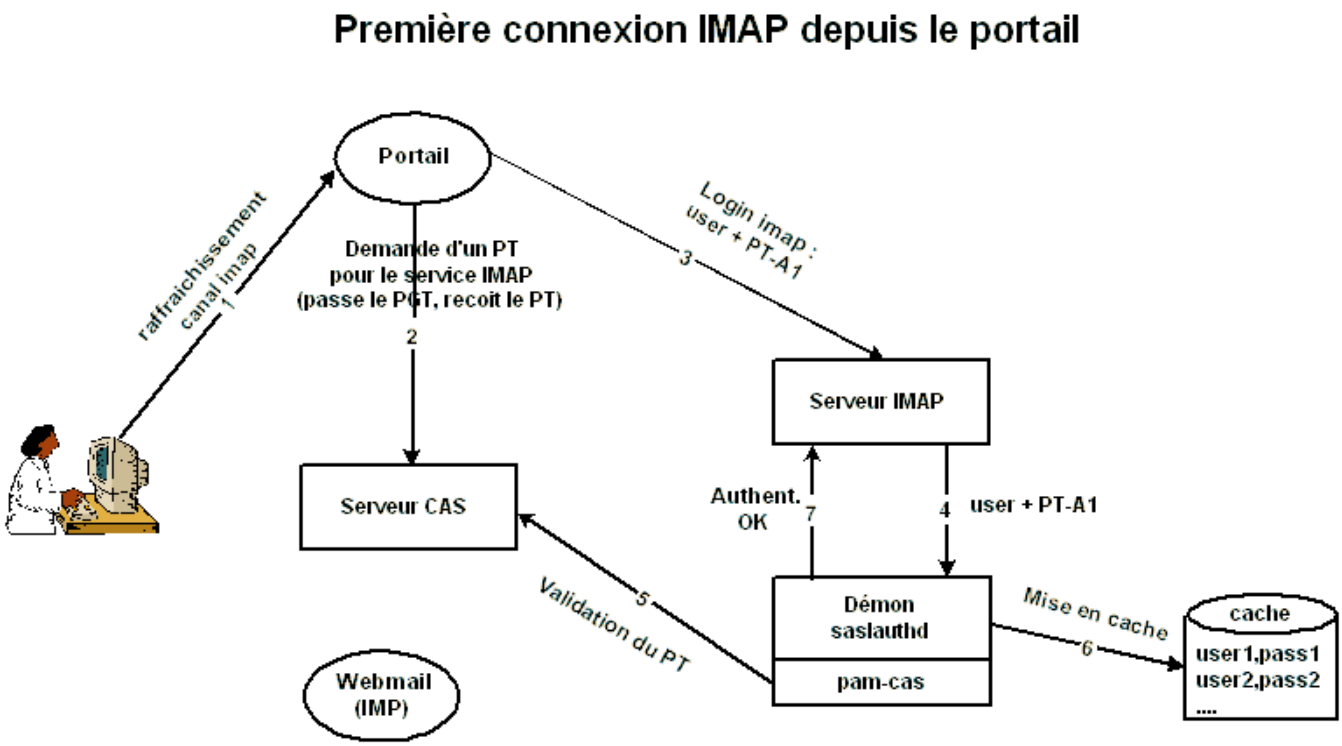
> Plutôt que de patcher saslauthd, il est maintenant plus simple d'utiliser la fonctionnalité "cacheDirectory" de la dernière version de esup-pam-cas.

Annexe : cinématique

Les schémas joints montrent la cinématique de l'authentification imap, dans différents cas.

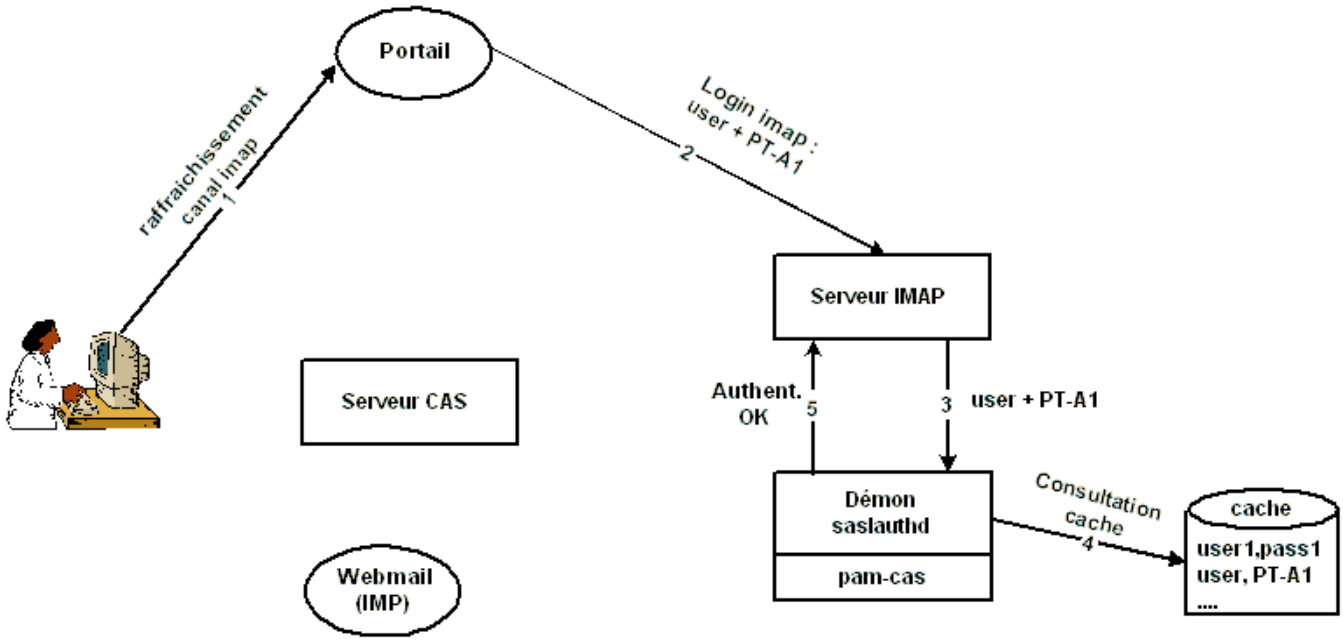
Il suppose que le portail et le webmail ont déja acquis auprès du serveur CAS un PGT (Proxy Granting Ticket) pour l'utilisateur.

Première connexion imap depuis portail



1. L'utilisateur provoque un évènement qui demande au 'canal' dédié à la messagerie d'accéder au serveur imap.
2. Le portail dispose d'un PGT pour l'utilisateur. Il demande donc au serveur CAS un PT pour le service imap (appelons-le PT-A1).
3. Il tente une connexion imap. Pour la phase de login imap, il va transmettre le login de l'utilisateur et le PT-A1.
4. cyrus-imap fait appel au démon saslauthd pour authentifier ; il passe donc le login et le PT-A1.
5. saslauthd n'a pas encore l'utilisateur dans son cache. Il passe donc ces 2 informations à l'interface pam. Celle-ci utilise pam_cas, et génère une requête https vers le serveur CAS avec ce PT. Le serveur CAS valide le PT, pam valide donc l'authentification pour saslauthd.
6. saslauthd met en cache l'information user + password (ici, PT-A1).
7. saslauthd valide l'authentification à cyrus-imap.
   La phase de login s'est donc bien déroulée.

Nouvelles connexions imap depuis le portail

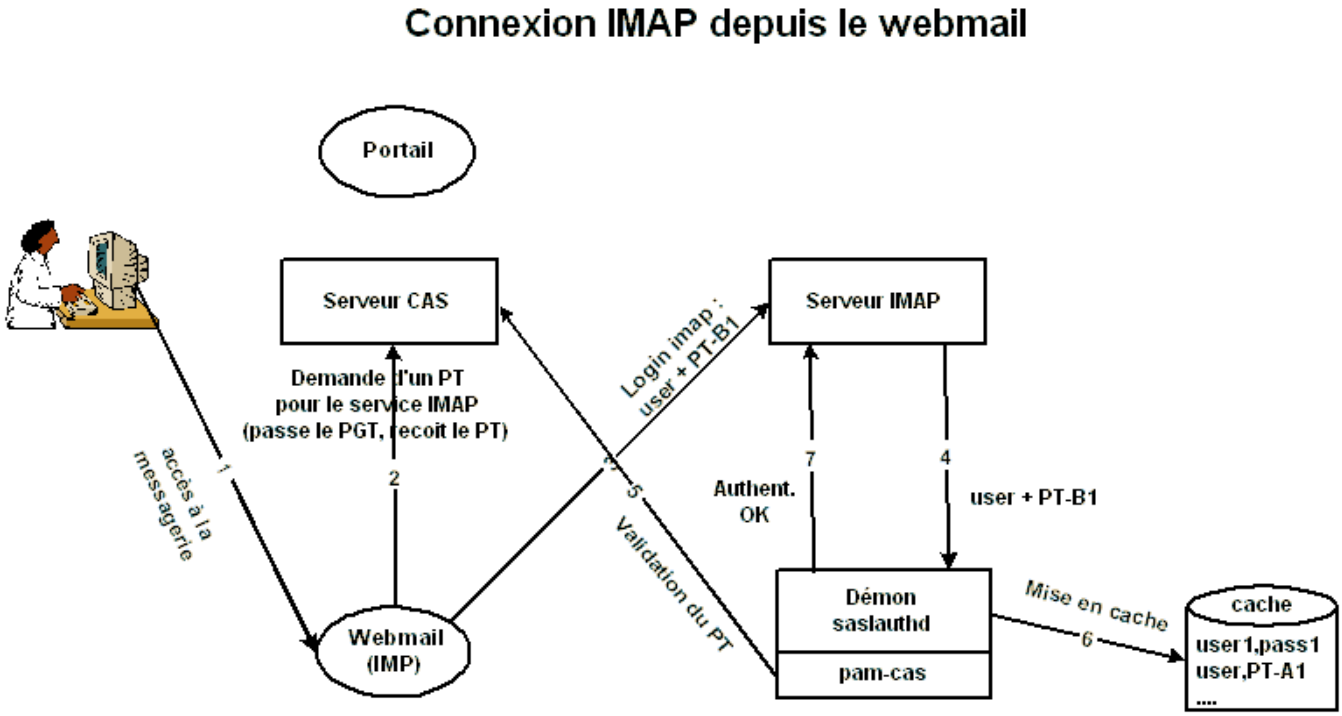


1. rafraichissement du 'canal imap'
2. login imap avec le login, et le même PT qu'auparavant (PT-A1)
3. cyrus-imap fait appel à saslauthd pour authentifier
4. saslauthd consulte son cache. il trouve une entrée pour l'utilisateur. Le mot de passe (PT-A1) est valide. Il n'a pas besoin de faire appel à pam-cas
5. L'authentification imap est effectuée

On voit bien que la procédure est beaucoup plus légère que précédemment.

Connexion IMAP depuis le webmail

On suppose maintenant que l'utilisateur vient de lancer le webmail IMP (depuis le portail, ou directement). IMP doit donc générer une connexion IMAP.



Le fonctionnement est identique au premier cas, sauf que saslauthd trouve une entrée pour l'utilisateur dans le cache ave PT-A1 comme mot de passe, alors que le mot de passe qui est proposé est un nouveau PT (appelé PT-B1).
saslauthd fait donc à nouveau appel à pam, qui valide. saslauthd met à jour son cache avec PT-B1, et authentifie la personne.

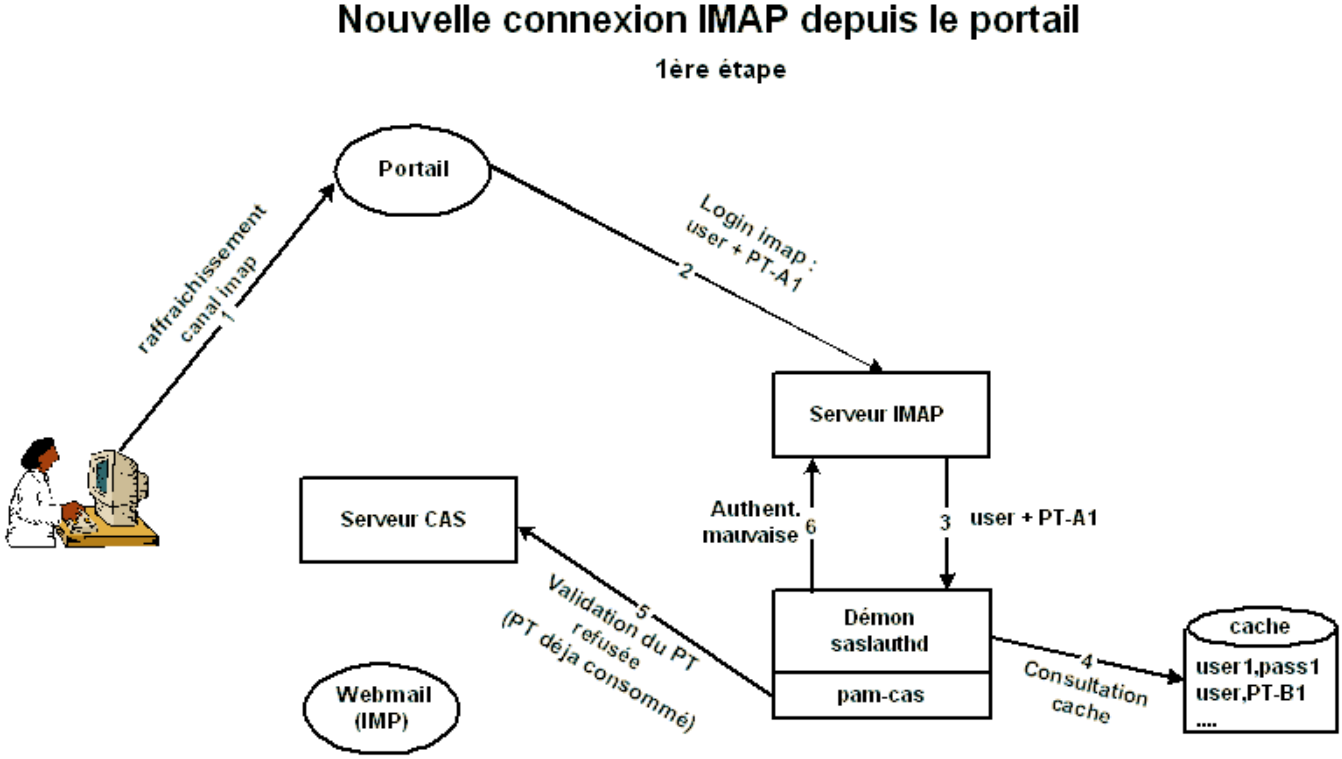
Nouvelles connexions imap depuis le webmail

identique à nouvelles connexions imap depuis le portail (second schéma)

Nouvelle connexion imap depuis le portail

On suppose que le portail doit à nouveau requêter IMAP.
Le 'canal' imap ne sait pas que le cache du serveur imap a changé. Le login IMAP va donc échouer :

Il faut donc que le canal sache analyser cette erreur, et puisse redemander à nouveau un PT pour l'utilisateur et redérouler tout le mécanisme.



Il faut aussi qu'il ne recommence pas cette procédure indéfiniment en cas de problème d'authentification.

Enfin, il va falloir gérer le même mécanisme côté webmail....