

abuledu-alacarte - Feature - Fonctionnalité #2160

Serveur relais pour l'authentification SSO

10/05/2012 18:15 - Eric Seigne

Statut:	Fixed - Corrigé - Implémenté	Début:	10/05/2012
Priorité:	Normale	Echéance:	
Assigné à:	Eric Seigne	% réalisé:	100%
Catégorie:		Temps estimé:	10.00 heures
Version cible:	Version 1.0		

Description

Il faudrait que alacarte se comporte comme une sorte de serveur relais pour propager l'authentification SSO à toutes les autres applications.

Le principe serait que les utilisateurs n'aient pas à s'identifier à chaque fois dans chaque logiciel ... c'est même un prérequis pour la partie logs et parcours pédagogiques.

Historique

#1 - 10/05/2012 18:17 - Eric Seigne

- % réalisé changé de 0 à 90

C'est fait pour la 1ere partie, mais il restera un chantier assez énorme pour tout ce qui est des clés d'authentification des différentes applications.

Actuellement les messages alacarte -> application sont cryptées (faible mais c'est déjà ça); à terme il faudra que chaque logiciel ait une clé spécifique pour que alacarte n'aille pas diffuser les informations sensibles à des utilisateurs qui auraient re-compilés un logiciel du terrier (et modifié un peu les logs) par exemple.

Pour la version 1.0 je pense que c'est satisfaisant, mais pour la suite il faudra finir cette partie !

#2 - 10/05/2012 18:22 - Eric Seigne

J'ai oublié d'indiquer ce qu'il faudra recopier dans la doc une fois validé:

- c'est intégré dans AbulEduNetworkAccessManagerV1
- AbulEduLocalSocketIpcServerV1 et AbulEduLocalSocketIpcClientV1 marchent de concert: la communication est à sens unique mais ça marche alors j'ai pas creusé du côté des QTCP + SSL
- La crypto est basique via SimpleCrypt

J'ai testé et validé avec imagéo. Il me reste du travail pour finir l'intégration propre dans alacarte ou j'ai encore pas mal de code dupliqué en ce moment.

#3 - 10/05/2012 18:27 - Eric Seigne

Voilà le code en question à intégrer dans la lib

```
//TODO a deplacer dans la lib en prenant compte que c'est le cas particulier de alacarte
//void MainWindow::localSocketMessageReceived(QString message)
//{
//    qDebug() << "MainWindow::localSocketMessageReceived : " << message;
//    //Contenu du message
```

```
// //HELLO:leterrier-calculment:ssoRequest
// QStringList params = message.split(":");
// qDebug() << "MainWindow::localSocketMessageReceived " << params;
// //On se connecte maintenant sur le client
// if(params.count() > 1)
//     m_localSocketClient->abeSlotChangeRemoteServer(params.at(1));
// QByteArray login("eric.seigne@tests.ryxeo");
// QByteArray pass("azaz");
// m_localSocketClient->abeSlotSendMessageToServer("HELLO:ssoAnswer:" + login.toBase64() + ":" + pass.toBase64());
// //Reponse
// //HELLO:ssoAnswer:jgkeljgieajgiaejgera6432654
//}
```

#4 - 16/05/2012 23:48 - Eric Seigne

- Statut changé de *Assigned* - En cours à *Fixed* - Corrigé - Implémenté

- % réalisé changé de 90 à 100

C'est fait, la lib est maintenant autonome.

À vérifier que ça marche bien, et il faudra aussi ajouter la propagation des nouvelles identités si jamais des logiciels sont encore ouverts au moment où on change d'utilisateur ...