Retour d'expérience sur l'utilisation de Kerberos à l'INRIA

Guillaume Rousse
INRIA, Direction des systèmes d'information
Domaine de Voluceau
Rocquencourt - BP 105
78153 Le Chesnay

Résumé

Kerberos est un protocole d'authentification sécurisé, de type Single Sign On, standardisé, utilisable depuis n'importe quel système d'exploitation. Son utilisation en dehors du cadre d'un environnement Active Directory, où il est utilisé de manière implicite, reste encore relativement confidentielle, et notamment à cause des difficultés d'intégration.

Le centre de recherche INRIA de Saclay a mis en place Kerberos depuis maintenant plusieurs années. Cette décision a été motivée à l'origine par le souhait de sécuriser l'utilisation de NFS pour les répertoires personnels des utilisateurs. Cette première expérience, qui s'est révélée par la suite être en fait l'une des plus complexes, nous a permis de nous rendre rapidement compte des potentialités de l'outil. Très vite, nous avons choisi d'expérimenter sa généralisation à d'autres usages, partout où c'était possible: sites Web, accès local, accès via SSH, serveur d'impression, etc.

Cet exposé présente les questions que nous nous sommes posées, les choix d'architecture qui ont été fait, les difficultés que nous avons rencontrées, et les solutions qui ont été mises en place pour y remédier. L'expérience montre que la difficulté majeure, une fois de plus, reste de pouvoir gérer les différents scénarios d'utilisation liés à une population d'utilisateurs hétérogène dans ses usages, dans ses outils, et dans sa capacité technique. Vu l'investissement nécessaire à la mise en place d'une telle infrastructure, il nous semble important d'analyser ces différents scénarios pour juger de la pertinence de cet effort dans les différents cas possibles.

Mots clefs

Authentification forte, Kerberos, Single Sign On, interopérabilité

1 Introduction

Cet article présente un retour d'expérience sur la mise en place de Kerberos dans le centre de recherche INRIA de Saclay. Il commence par quelques explications sur le contexte qui nous a amené à nous intéresser à ce protocole, puis donne un aperçu général de celui-ci, ses objectifs, ses concepts et son fonctionnement. La discussion principale concerne la manière dont ce système a été déployé puis intégré à l'infrastructure informatique du centre, et quels types d'applications ont pu en bénéficier. Enfin, quelques mots de conclusion tirent le bilan de l'expérience.

2 Contexte

En 2000, l'INRIA crée trois nouveaux centres de recherche à Saclay, Lille et Bordeaux. Malgré la dispersion géographiques, ces trois nouveaux centres sont gérés comme un centre unique,

sous le nom collectif d'INRIA Futurs jusqu'en 1997, lorsqu'ils deviennent totalement autonomes. Pendant une longue phase d'incubation, l'infrastructure informatique va principalement être fournie par un tiers, à savoir l'université d'accueil pour Lille et Bordeaux, et le centre de recherche de Rocquencourt pour Saclay. Mais au bout de quelques années, il devient de plus en plus nécessaire d'accéder à l'autonomie et de monter une infrastructure propre. Ceci implique notamment de proposer des comptes informatiques, et des espaces de stockage associés. Dans le monde Unix, ces espaces de stockage sont généralement utilisés comme répertoires personnels (*home directory*) accessibles en NFS. Et la sécurisation de ces accès est un problème récurrent, bien connu dans les services informatiques des centres.

Fort heureusement, nous sommes au XXIe siècle, et une nouvelle version du vénérable protocole NFS, NFSv4, vient d'être normalisée par l'IETF (RFC3530). Parmi les nombreux changements, il y a la possibilité d'utiliser un modèle de sécurité basé sur l'identité de utilisateur, et non plus sur celle de sa machine, en s'appuyant sur un autre protocole d'authentification robuste et lui aussi normalisé, Kerberos (RFC 4120). Et parmi les artisans de cette nouvelle version, il y a NetApp, le fabricant des solutions de stockage utilisées à l'INRIA, et dont les commerciaux nous promettent qu'elle est supportée sans problème. Bingo, il n'y a plus qu'à essayer d'aller titiller le cerbère...

3 Présentation du protocole

3.1 Objectifs

Kerberos est un protocole d'authentification robuste, de type SSO (Single Sign On). Ces trois termes sont important ici.

C'est un protocole d’authentification : il garantit qu'un utilisateur est bien celui qu'il prétend être, et uniquement ceci. Les attributs liés à cette identité, et notamment ceux correspondant à des attributions, ne sont pas gérés.

Il est robuste, c'est à dire qu'il offre un certain nombre de garanties que cette identification est fiable et difficile à usurper.

Enfin, il s'agit d'un protocole de type Single Sign On, dans la mesure où du point de vue de l'utilisateur, celui-ci s'authentifie une fois et une seule, par exemple en saisissant un mot de passe, et que cette authentification est ensuite automatiquement propagée à l'ensemble des applications auxquelles il s'adresse.

Ces différentes caractéristiques montrent que Kerberos permet à la fois d'améliorer sécurité et ergonomie pour l'utilisateur final, alors que ces deux notions sont habituellement antagonistes.

3.2 Concepts

Une installation de Kerberos constitue un « royaume », c'est-à-dire une entité distincte de toute autre installation. Ce royaume va constituer un espace de nommage unique pour ses différents participants. Par convention, et afin d'assurer cette unicité, c'est le nom de domaine DNS mis en majuscule qui détermine le nom de ce royaume. Au domaine saclay.inria.fr correspond donc le royaume SACLAY.INRIA.FR.

À l'intérieur de ce royaume, chaque participant est identifié par un « principal », de la forme *nom@ROYAUME.* Certains acteurs peuvent éventuellement posséder plusieurs principaux, différenciés alors par leur instance, ce qui donne des principaux de la forme *nom/instance@ROYAUME*. Typiquement, les principaux correspondants à des services hébergés sur une machine sont de la forme *service/machine@ROYAUME*, et les administrateurs système possèdent des principaux *nom/admin@ROYAUME* dédiés à l'administration en supplément de leurs principaux d'utilisateur *nom@ROYAUME*.

L'élément central de chaque royaume est le Key Distribution Center, ou KDC. Celui-ci est constitué d'une base de données de tous les principaux du royaume, chacun associé à un secret partagé avec l'entité correspondant à ce principal. Pour un utilisateur, ce secret est son mot de passe. Pour un service, ce secret est en général généré aléatoirement à la création du principal. Ce secret est stocké sous la forme d'une clé pour chaque algorithme de chiffrement supporté. D'autres informations relatives au principal peuvent également être stockées, comme par exemple la durée de validité du principal, la date de dernier changement de mot de passe, etc.

Le KDC joue le rôle d'un tiers de confiance, en distribuant aux utilisateurs des tickets prouvant leur identité vis-à-vis d'un service cible. Ces tickets sont en fait des jetons d'authentification, à durée de vie limitée afin d'éviter une attaque par rejeu. Ils sont totalement opaques pour l'utilisateur, et déchiffrables uniquement par l'interlocuteur qu'il a choisi, ce qui assure une authentification réciproque.

3.3 Fonctionnement

Dans le cas du fonctionnement normal de Kerberos, l'authentification se fait en trois étapes, représentées sur la Figure 1.

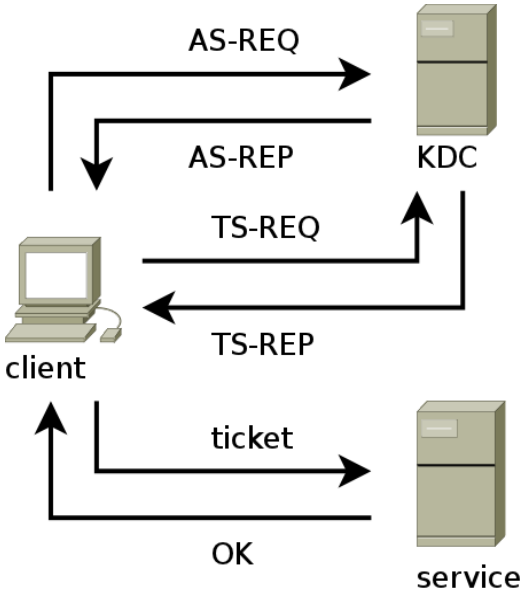

Figure 1: authentification Kerberos

Le client commence par s'adresser au KDC, par le biais d'un message AS-REQ (*Authentication Service Request*), contenant uniquement son identité. Le serveur lui répond par un message AS-REP (*Authentication Service Reply*), chiffré par le secret partagé avec l'utilisateur, contenant un premier élément d'authentification, le TGT (*Ticket Granting Ticket*) ou ticket maître, et une clé de session. Seule la connaissance du secret partagé permet donc d'accéder à ce ticket.

Dans un second temps, le client désireux de s'adresser à un service particulier va de nouveau s'adresser au KDC par le biais d'un message TS-REQ (*Ticket Service Request*), contenant l'identité du service en question, et le fameux TGT obtenu à l'étape précédente. Le serveur répond alors par un message TS-REP (*Ticket Service Reply*) contenant un nouvel élément d'authentification, le ticket pour le service demandé, chiffré avec la clé de session.

Enfin, dans un troisième temps, le client s'adresse au service voulu, par le biais du protocole dédié (HTTP, FTP, SSH, etc.) et lui présente le ticket obtenu à l'étape précédente, et obtient l'accès désiré ou pas, si un mécanisme d'autorisation est mis en place.

Seule la première étape nécessite une interaction avec l'utilisateur, à savoir la saisie du mot de passe pour déchiffrer la réponse du KDC. De plus, durant toute la durée de vie du TGT (généralement de l'ordre de la journée), elle n'est plus nécessaire, et tout accès à un nouveau service est transparent. C'est ce qui constitue le principe du SSO, l'authentification unique.

Ce fonctionnement nécessite bien sûr que l'application rendant le service visé, le client utilisé et le protocole utilisé soient spécifiquement conçus pour gérer ce mode d'authentification, ce qui constitue une condition relativement restrictive.

Il existe également un autre mode de fonctionnement, qui utilise également l'infrastructure Kerberos, mais qui n'a pas grand chose à voire avec le précédent, dans la mesure où le client ne gère plus de ticket d'authentification. Ce mécanisme, que l'on peut qualifier de validation de mot de passe par Kerberos, est représenté sur la Figure 2.

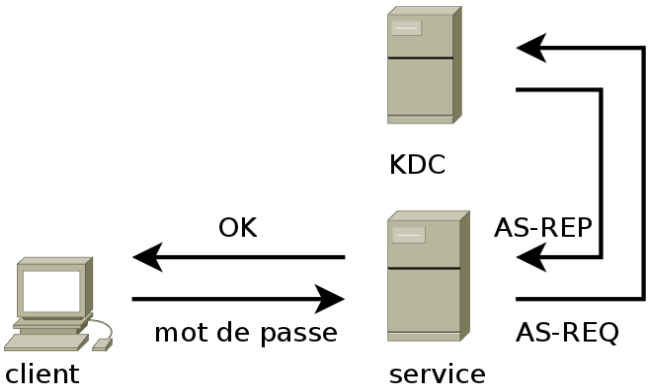

Figure 2: validation de mot de passe

Dans ce mode de fonctionnement, le client se contente de s'adresser au service visé en utilisant un classique couple « nom d'utilisateur/mot de passe ». Le service va alors utiliser ces informations pour rejouer l'authentification auprès du KDC, par le biais du dialogue AS-REQ/AS-REP détaillé précédemment. Si le message AS-REP peut être déchiffré grâce au mot de passe, alors le mot de passe est validé.

Ce fonctionnement présente l'énorme avantage de ne plus nécessiter d'adaptation du client ou du protocole applicatif. Par contre, il n'offre aucune des garanties de robustesse de Kerberos, puisque le mot de passe circule entre le client et le service, et ne constitue pas non plus une solution SSO. On peut donc le voir comme une solution de repli, lorsque le mécanisme normal n'est pas utilisable.

Pour plus de détails concernant le fonctionnement du protocole, voir [1] et [2].

4 Mise en place d'un royaume Kerberos

4.1 Intégration au référentiel de comptes utilisateurs

La mise en place de Kerberos amène à créer une liste de couples « principal/mot de passe », donc en fait une nouvelle base de comptes utilisateurs. Ce qui amène immédiatement au problème de la synchronisation de cette base avec le référentiel principal. Il s'agit en effet d'assurer la double contrainte que tous les utilisateurs possèdent également un principal Kerberos, et que le mot de passe associé à ce principal soit identique à celui utilisé ailleurs.

Afin de simplifier au maximum ce problème, nous avons choisi d'intégrer cette base Kerberos à notre référentiel, qui se trouve être un annuaire LDAP de comptes POSIX. Et nous avons choisi pour cela d'utiliser Heimdal, une implémentation de Kerberos alternative à celle du MIT, qui offrait à l'époque de meilleures capacités à cet égard.

Cette intégration comporte deux aspects :

- stockage de la base Kerberos dans l'annuaire LDAP
- mise en place d'un greffon de synchronisation des mots de passe

Le stockage de la base dans l'annuaire est une option offerte par Heimdal (MIT peut également le faire maintenant, mais avec un schéma différent) pour utiliser un annuaire plutôt qu'une base BDB classique pour stocker ses informations. De plus, nous avons choisi de fusionner ces informations supplémentaires avec celles des comptes utilisateurs, en ajoutant les attributs nécessaires aux enregistrements existants, plutôt que d'en faire des enregistrements distincts dans une autre branche. La Figure 3 présente l'arborescence résultante : les comptes utilisateurs sont dans la branche *ou=users,dc=saclay,dc=inria,dc=fr*, les principaux correspondant à des services sont dans la branche *ou=kerberos,dc=saclay,dc=inria,dc=fr*.

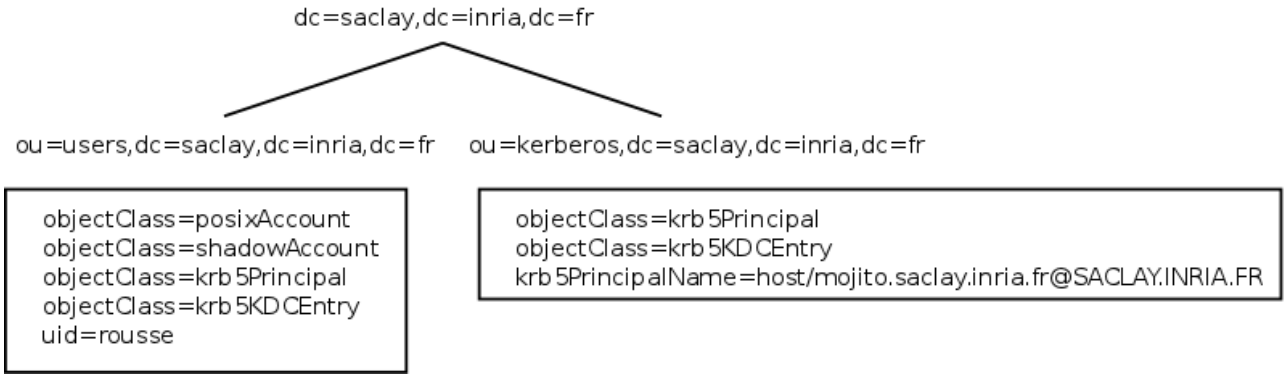

Figure 3: Arborescence de l'annuaire LDAP

La synchronisation des mots de passe se fait par le biais d'un greffon pour OpenLDAP (*overlay*, dans la terminologie du projet), *smbk5pwd*, qui intercepte toute action de changement du mot de passe et synchronise automatiquement les clés à Kerberos (une par algorithme de chiffrement supporté). Ce greffon est par contre totalement spécifique au couple Heimdal/OpenLDAP. Cela suppose évidement que le mot de passe soit transmis en clair au serveur, par une connexion chiffrée, et qu'il en assure le chiffrement lui-même par une opération étendue dédiée. L'utilisation d'un autre greffon, *ppolicy*, assure que les manipulations directes de l'empreinte du mot de passe ne sont pas possibles.

Au final, il n'y a plus deux bases de comptes qu'il faut synchroniser, mais une seule, étendue pour supporter l'utilisation de Kerberos de façon totalement transparente.

4.2 Fiabilisation de l'infrastructure

La base Kerberos étant intégrée à l'annuaire LDAP, il n'y a pas besoin d'utiliser les mécanismes de réplication spécifiques à Heimdal, ceux d'OpenLDAP suffisent. Dans notre cas, nous avons mis en place une réplication multi-maîtres synchrone, avec deux serveurs, pour une base d'environ 1000 comptes utilisateurs.

Par ailleurs, ces serveurs ne sont pas accessibles directement, mais via un mécanisme de type SLB (S*erver Load Balancing*), géré par un équipement réseau, assurant répartition de charge et haute disponibilité du service, comme illustré par la Figure 4.

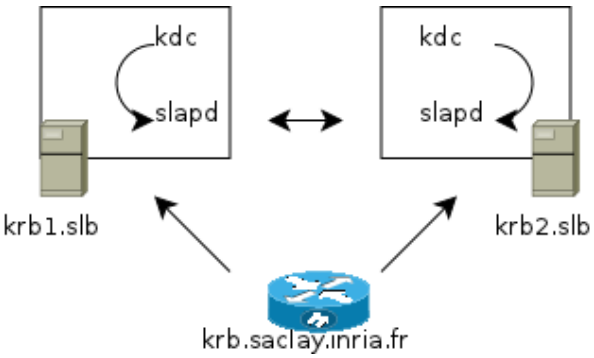

Figure 4: répartition de charge et haute disponibilité

5 Mise en œuvre applicative

5.1 NFS

Notre motivation initiale concernant NFSv4, il était normal de commencer par là. Malheureusement, le chemin était truffé d'embuches.

D'abord, NFSv4 ne fonctionnait tout simplement pas correctement sur nos équipements NetApp, Kerberos ou pas Kerberos. Les commerciaux nous l'avaient vendu, les ingénieurs du support nous l'ont immédiatement fait désactiver dès les premiers problèmes. Et l'arrivée de la version majeure 7.3 de DataOnTap, censée supporter le protocole de manière stable, n'a rien changé, il a fallu attendre une obscure révision 7.3.3P4 pour que le problème soit résolu de manière totalement anecdotique. Malgré une analyse technique détaillée du problème, et l'aide des ingénieurs de StoreData, il aura été totalement impossible d'obtenir l'escalade et le suivi du dossier chez NetApp. Pendant longtemps, nous avons donc été réduits à utiliser Kerberos avec NFSv3, le support ayant été rétro-porté.

Ensuite, et jusqu'à la version 7.3.1 de DataOnTap, il était impossible d'utiliser un royaume Kerberos pour NFS distinct du domaine ActiveDirectory. Il fallait donc passer par une relation d'approbation, et donc subir de multiples problèmes additionnels liés à l'ajout d'une couche de complexité supplémentaire. L'utilisation de royaumes séparés (mode *dual head* dans la documentation) apporte un progrès notable.

Enfin, NFS a un fonctionnement largement plus complexe que les autres services : le client effectue le montage du système de fichiers avec ses propres authentifiants, avant qu'un ou plusieurs utilisateurs distincts utilisent les leurs pour accéder aux fichiers. Il faut donc créer des principaux et déployer des fichiers « keytab » sur les machines des utilisateurs, ce qui est très difficile à faire sur des machines qui ne sont pas gérées par les services informatiques. Nous avons mis en place des interfaces CGI de création et distribution sur demande, mais leur utilisation par le public ciblé (les utilisateurs autonomes) est anecdotique.

Au final, kerberiser NFS, c'est possible, et cela fonctionne parfaitement, mais n'est réaliste que sur un parc géré de bout en bout (serveur et clients).

5.2 OpenSSH

Comparé au cas précédent, OpenSSH est trivial à configurer. En effet, l'authentification Kerberos est juste un mode d'authentification supplémentaire, comme peuvent l'être l'utilisation du mot de passe ou de clés, à activer dans la configuration du serveur en premier, dans celles des clients ensuite. Si la configuration du client n'est pas adaptée, il n'y a aucun effet secondaire.

OpenSSH permet également d'utiliser Kerberos pour la validation de mot de passe, comme présenté plus haut. Ce mode est intéressant dans certaines situations, comme par exemple une machine configurée pour monter des répertoires utilisateur par NFS kerberisé : en effet, on peut alors garantir que l'utilisateur, une fois authentifié, se trouve en possession d'un ticket Kerberos permettant d'accéder au système de fichiers, sans aucune configuration spécifique de son côté. Bien évidemment, il faut alors justifier auprès de ces utilisateurs l'exception à la politique générale que constitue l'utilisation d'une authentification par mot de passe, alors qu'elle est strictement interdite ailleurs au profit de l'authentification par clé...

5.3 Applications Web

Le module apache *mod_kerb* permet d'utiliser Kerberos pour des authentifications HTTP. Ceci permet de protéger des sites web statiques, ainsi que certaines applications déléguant la gestion de l'authentification au serveur Web. Parmi celles-ci, on peut citer notamment Cacti, Munin, Nagios.

La gestion de Kerberos par les navigateurs, par contre, est très hétérogène. Pour certains, comme Firefox et consort, il faut activer des options de configuration spécifiques, pour d'autres non, comme Safari. Le fait que même sur un parc géré de bout en bout, la gestion centralisée des applications de la fondation Mozilla soit une plaie complique encore la chose. Le délicat problème de la gestion de l'identité du service par le client montre également des comportement très variables, qu'il faut contourner du côté du service.

Heureusement, *mod_kerb* propose également un repli sur une authentification HTTP Basic classique si la négociation initiale échoue, avec validation du mot de passe via Kerberos. On peut donc ici également déployer Kerberos en complément d'une authentification classique, sans pénalité pour les utilisateurs non compatibles.

5.4 LDAP

La kerberisation de l'annuaire LDAP (utilisation de Kerberos pour authentifier les accès à celui-ci) est également relativement simple à mettre en place. Comme pour OpenSSH, il s'agit d'autoriser un mode d'authentification supplémentaire, en complément de l'authentification simple par mot de passe. Contrairement aux autres applications, le public visé concerne essentiellement les administrateurs systèmes, les utilisateurs accédant assez rarement directement à l'annuaire, il est donc plus à même d'effectuer les opérations de configuration nécessaire (mise en place de SASL, notamment).

Couplé à l'utilisation d'un outil comme ldapvi, permettant de lire et de modifier les données de l'annuaire au format LDIF dans l'éditeur de texte de son choix, ce mode d'authentification permet une facilité d'accès peu usuelle avec ce type d'infrastructure. Pour une population scientifique habituée à ce genre d'outil, il suffit de lancer `'EDITOR=emacs ldapvi (uid=login)'` sur une machine convenablement configurée pour modifier ses propres informations administratives, par exemple.

Là encore, la mise en place de Kerberos est simple, et apporte une forte plus-value, même si c'est vis-à-vis d'une population restreinte.

5.5 Cups

La mise en place de l'authentification sur un serveur CUPS peut se faire à deux niveaux : l'impression elle-même, pour gérer par exemple des autorisations ou faire du suivi de quota, et l'accès à l'interface web, très pratique pour permettre aux utilisateurs d'annuler leurs taches bloquantes dans les queues d'impression.

La première possibilité, ne correspondant pas à nos besoins, n'a pas été testée. La deuxième, en revanche l'a été. Cependant, elle présente un gros souci : elle n'offre pas un complément à une authentification classique par mot de passe, mais un remplacement. Ceci exclut d'office tout navigateur mal configuré sans solution de repli. Cette utilisation n'a donc pas été retenue.

5.6 Interopérabilité avec Windows

Active Directory utilise nativement Kerberos pour gérer l'authentification, et chaque domaine Active Directory constitue également un royaume Kerberos. Lorsqu'une relation de confiance est mise en place avec un autre domaine Kerberos, il devient possible de déléguer les opérations d'authentification à cet autre royaume. Autrement dit, fini le cauchemar récurent de devoir synchroniser les mots de passe entre les deux univers Unix et Windows, puisque le second peut utiliser le mécanisme d'authentification du premier de façon transparente...

Néanmoins, la mise en place est complexe. En plus de la relation d'approbation (et des subtilités de configuration qui vont avec), il faut de toute façon que les comptes utilisateurs existent de part et d'autre (seule la synchronisation des mots de passe n'est plus nécessaire), et qu'il y ait une correspondance explicite entre les deux. De plus, Windows étant incapable d'utiliser les enregistrements DNS pour identifier la configuration d'un royaume Kerberos tiers, il faut déployer au préalable sur les machines du parc la configuration nécessaire...

Une fois les difficultés techniques résolues, il faut également former les utilisateurs à s'authentifier sur le domaine externe, plutôt que sur le domaine natif, ce qui peut être éventuellement rendu transparent par configuration. Mais le véritable problème vient surtout du fait que Kerberos n'est utilisable qu'au sein du domaine Active Directory. Autrement dit, pour des utilisateurs possédant des comptes informatiques, mais dont les machines ne sont pas intégrées au domaine, pour une raison ou une autre (typiquement, des visiteurs), ceci ne fonctionne tout simplement pas. Pour pouvoir utiliser le mécanisme d'authentification de repli (NTLM), un mot de passe doit obligatoirement être associé au compte natif. D'où la nécessité de garder un mécanisme de synchronisation de mots de passes, et donc une diminution forte de l'intérêt de mettre en place un système alternatif.

6 Conclusion

Kerberos tient effectivement ses promesses et apporte sécurité et confort d'utilisation. Il présente cependant un coût technique important pour sa mise en place, surtout dans un contexte hétérogène. Et également un coût psychologique important dans la mesure où des utilisateurs ne connaissant que le sésame universel du couple « login/mot de passe » sont généralement peu enclins à changer leurs habitudes, même pour se débarrasser d'interfaces fastidieuses.

Ce coût psychologique disparaît si ces changements sont menés de manière transparente pour les utilisateurs. Sur un parc géré de bout en bout, c'est tout à fait envisageable. Mais dans nos environnement habituels, avec des chercheurs gérant eux-même leurs machines, des invités qui vont et viennent, c'est largement plus complexe. Dans ce cas de figure, Kerberos reste très intéressant comme possibilité complémentaire aux autres mécanismes d'authentification, mais le tout Kerberos relève de la douce utopie...

7 Bibliographie

[1] Jason Garman. *Kerberos: The Definitive Guide*. O'Reilly Media, 2003.

[2] Guillaume Rousse, Kerberos, le SSO universel. *GNU/Linux Magazine*, 143, novembre 2011.