Kerberos : Linux, Windows et le SSO

Emmanuel Blindauer
IUT Robert Schuman, Strasbourg
Emmanuel.Blindauer@urs.u-strasbg.fr

Résumé

Avec la multiplication des postes informatiques, l'authentification des utilisateurs doit être uniforme pour tous. L'unification de l'authentification entre windows et linux est le problème le plus courant qui se pose. Il est possible de faire cette réunification via une authentification kerberos. Les différentes étapes de configuration sont décrites, ainsi que les services additionnels nécessaires pour faire fontionner les systèmes linux. L'utilisation de kerberos apporte ensuite des avantages en terme de sécurité, en ne transportant plus les mots de passe sur les réseaux, mais également en fournissant des solutions de Single Sign On sur des services au niveau du poste de travail, et des services web. Enfin une intégration de cet environnement au sein d'un Espace Numérique de Travail est effectuée (EPPUN), en utilisant le système d'authentification requis par ce projet.

Mots clefs

Active Directory, Authentification Unique, Linux, Single Sign On, Windows

1 Introduction

Les services que nous devons rendre à nos utilisateurs sont très variés. Un point primordial est l'authentification. C'est le premier service rendu à l'utilisateur, le plus utilisé, mais aussi un de nos plus gros problèmes : La sécurité de l'authentification est primordiale et l'hétérogénéité des parcs nous impose énormément de contraintes.

Plus le parc grandit, plus une multitude de systèmes doivent cohabiter. Les utilisateurs, eux, ne veulent pas connaître les différences entre systèmes. C'est à nous de faire un travail d'intégration. De plus, lorsqu'ils voient un fonctionnement sur un lieu, par exemple un Single Sign On pour accéder à un service de mail ou à un intranet, ils veulent pouvoir disposer de solutions équivalentes dans leur composante, même si techniquement ce n'est pas toujours possible...

Mais donnons à nos utilisateurs le maximum de ce qu'on puisse faire techniquement. L'utilisation d'un couple d'identification unique *(login/pass)* doit être la première étape à mettre en place. Ensuite il faudra exploiter l'existant pour faire une intégration la plus complète, tout en simplifiant les services aux utilisateurs.

2 Les solutions existantes

L'utilisation d'un grand nombre de plateforme Unix a nécessité très tôt une authentification centralisée. C'est le service NIS, puis NIS+ qui a été utilisé à ces fins pendant de nombreuses années. Cependant, plusieurs problèmes apparaissent : Cette solution n'est pas compatible qu'avec les Unix classiques. En particulier Windows a énormément de mal à l'utiliser, via *Services For Unix (SFU)* et ne sait synchroniser les mots de passe que de Windows vers NIS. L'autre problème est la conception inhérente à NIS : une usurpation d'identité (au niveau IP) et/ou la connaissance du domaine NIS permet de récupérer l'ensemble des mots de passe cryptés. En particulier lorsqu'on utilise NFS, ce qui est souvent le cas, l'introduction d'un client NFS compromis permet de passer outre bon nombre de restrictions d'accès. NIS a été une solution dans les débuts de l'informatique, mais actuellement, la sécurité ne peut plus être assurée dans de bonnes conditions pour continuer à l'utiliser : La facilité d'accès à une prise RJ45 ou a un réseau wifi, et les systèmes et outils actuels permettent de se faire passer pour un client légitime trop facilement.

Dans le monde Windows cette problématique a été prise en compte, et l'utilisation de réseaux virtuels appelés DOMAINE, où un client doit être ajouté par l'administrateur apporte une réponse partielle. Dans les domaines NT4, les contrôleurs de domaines primaires et secondaires gèrent les mots de passe, sous forme crypté LMHASH/NTHASH mais il s'est avéré que ce type de codage n'était pas très sécurisé [1]. Dans les domaines Windows 2000 et supérieur, ce codage est conservé que pour compatibilité. Par contre, les informations relatives à l'authentification restent peu connue de la plupart des administrateurs systèmes. D'autre part, le fait que Windows reste un système propriétaire fait qu'on ne sait pas facilement comment modifier de manière sûre le système d'authentification, ni que l'on ne peut s'assurer de son interopérabilité avec d'autres systèmes à long terme.

Une façon courante de faire cohabiter des postes Windows et linux s'est développée ces dernières années avec l'utilisation du logiciel *samba*[2] en particulier à partir de la version 3, où le domaine obtenu est plus riche en fonctionnalité que les domaines NT4 mais pas encore au niveau des domaines Windows 2000 : Le serveur samba utilise une base LDAP pour gérer ses utilisateurs, et stocke à la fois le couple LM/NTHASH et les systèmes Unix utilisent une authentification complètement basé sur LDAP [3]. Dans ce cas, le service rendu est largement suffisant pour bon nombre de composantes.

Une dernière façon de synchroniser les identifiants et mots

de passe est l'utilisations de scripts *maisons*. Plusieurs tentatives ont été mises en place mais on se retrouve toujours avec une configuration un peu «bricolée» [4]. La sécurité des synchronisations entre les différents acteurs n'est pas des plus évidentes, et on finit souvent par se retrouver avec un système ayant deux serveurs synchronisés, un pour la partie Unix, un pour la partie Windows.

Il pourrait sembler d'après ce qui a été présenté ici que la solution samba 3 est la plus intéressante à mettre en oeuvre. Pourtant, il reste plusieurs points problématiques :

Le premier est que le mode *fonctionnel* du domaine basé sur samba est très faible en fonctionnalités par rapport à un réseau Active Directory. Peu de logiciels actuellement requièrent l'utilisation d'un domaine Active Directory, mais ce nombre va surement augmenter, et cela finira par poser un problème. De plus, il ne faut pas fermer les yeux sur les évidences : Active Directory propose un certains nombre de fonctionnalités plus qu'intéressantes pour des administrateurs systèmes : Déploiement de logiciels à distances, verrouillages de fonctionnalités sur certains postes, ou configuration à distance de certaines applications.

Ensuite, il reste un problème de conception dans les authentifications basées sur LDAP : stocker des données sensibles (mots de passe) à coté de champs destinés à être rendues publiques (Nom, téléphone, fax...) reste une antinomie. Certes, il existe des ACL dans les annuaires LDAP pour choisir ce qui sera visible, mais quiconque les a déjà manipulées connaît la difficulté de la tâche.

Enfin, dans chaque transfert lors d'une authentification, le mot de passe est transmis, mais sous forme chiffrée. Il faut minimiser au maximum ces transferts, car même si le chiffrement est en théorie non réversible, la comparaison avec une table de mots déjà chiffrés est possible...

Dernière problématique pour les utilisateurs : les ré-authentifications. Il est de plus en plus mal vu de devoir faire des ré-authentifications pour accéder à certains services quand c'est le même couple (login/pass) qui doit être utilisé. Il faut fournir un service de *Single Sign On (SSO)*, c'est à dire que l'utilisateur ne doit s'authentifier qu'une seule fois, et qu'après, tous les services compatibles ne doivent pas redemander une authentification.

3 La solution déployée

Ce papier décrit exactement tout ce qui a été mis en place au département Informatique de l'IUT Robert Schuman d'Illkirch, à la rentrée 2004 pour la plus grosse partie, et à la rentrée 2005 pour la partie liée au projet SUPANN. L'ensemble de la centaine de postes clients, dont la plupart en dual-boot, a été reconfiguré pour s'intégrer à ce qui va être présentée : L'authentification se fait dans un domaine Active Directory en mode mixte, géré par deux serveurs DC (*Domain Controller*), et épaulés d'un serveur LDAP. Les utilisateurs ont un service SSO immédiat vers l'extranet maison, ainsi que vers une application web utilisée principalement en tant que système de gestion de contenu communautaire (Wiki).

4 Le principe utilisé

Le principe est de s'appuyer sur un système d'authentification Kerberos [5].

4.1 Kerberos

Sans entrer dans les détails, voici quelques informations sur Kerberos : Les serveurs KDC servent à stocker les mots de passe et à délivrer des tickets. Les tickets sont des accréditations chiffrées avec une durée limitée pour accéder à différents services. Le domaine d'action de Kerberos est nommé REALM (et est toujours en majuscule), on peut en faire cohabiter plusieurs dans le même réseau. Chaque client utilisateur, ordinateur possède l'équivalent d'un login : un *principal* précédé quelquefois d'un nom de service : `blindaue@DOMAINE.LOCAL` pour un compte utilisateur et `HOST/ibis.domaine.local` pour ordinateur. Le seul moment où le mot de passe transite sur le réseau est lors de la première authentification. Ensuite des *tickets* chiffrés (et qui ne contiennent pas de mot de passe) sont utilisés. Ce sont ces tickets, qui, présentés à certains services adaptés permettent d'authentifier la personne. Ils remplacent le couple (login/pass), et cela permet d'avoir des applications (classique ou web) totalement compatibles SSO.

Kerberos a été choisi car les domaines Active Directory (mixte ou natif) utilisent Kerberos pour effectuer les authentifications en plus d'intégrer un serveur LDAP. Attention lors des migrations vers un domaine Active Directory : seuls les mots de passe modifiés après la migration utilisent Kerberos, les anciens restent basés sur LM/NTHASH en attendant le prochain changement de mot de passe. Kerberos est également disponible dans le monde Unix puisqu'il y a été crée. Enfin, il permet de bénéficier du SSO dans sa conception, ce qui est un plus.

4.2 Configuration Windows

Au niveau des postes clients, il n'y a rien à configurer quand ils sont dans un domaine. Il est possible d'utiliser `klist.exe` pour avoir la liste des tickets de l'utilisateur connecté pour visualiser les tickets qui sont disponibles, et donc que Kerberos est bien utilisé.

Pour les contrôleurs de domaine, il faut savoir qu'il y a une contrainte forte : Le domaine DNS doit être le même que le REALM. En plus, le serveur DNS doit pouvoir supporter les mises à jours dynamiques par les clients eux même. Si on ne peut pas disposer de droits suffisants sur le serveur DNS en place, on doit pallier ce problème : Il faudra créer son propre domaine DNS interne pour que le domaine puisse fonctionner, (par exemple DOMAINE.LOCAL.). Il faudra obligatoirement que tous les postes clients aient comme serveur DNS, les contrôleurs de ce domaine. Dans le cas contraire, des dé-

lais de plusieurs minutes lors des ouvertures de sessions et des surcharges du réseau due au broadcast pour chercher le DC seront les problèmes les plus courants !

4.3 Configuration Linux - Authentification

Dans le schéma actuel d'authentification et d'autorisation de Linux, c'est le service `PAM` qui est utilisé. Ce service est modulaire et de nombreux composants sont disponibles. Ici, il faut utiliser pam_krb5 (ou tout autre système permettant l'authentification Kerberos sur d'autre Unix). A ce niveau, il n'y a rien de difficile, il faut simplement configurer `/etc/krb5.conf` :

```
[realms]
DOMAINE.LOCAL = {
  kdc = dc1.u-strasbg.fr:88
  admin_server = dc1.u-strasbg.fr:749
  default_domain = u-strasbg.fr
 }
```

pour indiquer le REALM (qui est aussi le nom de la forêt Active Directory) et que les serveurs sont ceux d'Active Directory. Ici le vrai domaine DNS est `u-strasbg.fr`, mais le domaine Active Directory est `DOMAINE.LOCAL` et le domaine des postes Windows sera `domaine.local`

Il faut cependant veiller à ne pas forcer les types d'encodages des tickets. (`default_tkt_enctypes` et `permitted_enctypes`), ou en cas de gros problème, se réferrer à la base d'informations de Microsoft pour vérifier les types utilisés, qui diffèrent dans Windows 2003 en particulier[6]. On peut facilement vérifier cette étape en demandant un ticket au serveur :

```
kinit userAD
```

en prenant un utilisateur `userAD` du domaine.

Kerberos se basant également sur l'heure pour délivrer des tickets, il ne faut pas hésiter à installer *ntp* sur les postes clients, afin de ne pas dépasser les 5 minutes de décalage qui invaliderait tout ticket.

4.4 Configuration Linux Name Service

Il manque encore une autre partie essentielle pour avoir un système fonctionnel sous linux. En effet, il n'y a pas encore d'informations dites NSS, c'est à dire son numéro d'utilisateur, son numéro de groupe, son répertoire de travail, etc. Il faut récupérer ces informations soit dans Active Directory, via LDAP ou par appel RPC, ou les inventer. Pour cela on va utiliser le programme *winbind* de samba. Ce programme est chargé d'effectuer la correspondances entre *SID* et *uid*, entre *SID* et *gid*, de fournir les informations *GECOS*, ainsi que le répertoire de base et le shell à utiliser pour chaque utilisateur.

winbind gère la correspondance entre *SID* et *uid* à la volée. Elle n'est pas forcement la même sur différentes stations. Dans certains cas, il est nécessaire d'avoir une correspondance identique sur tous les ordinateurs. Il existe pour cela un sous-système nommé `idmap_rid` qui crée ce mapping uniforme, mais ne gère qu'un seul domaine. Dans le cas de plusieurs domaines, ou si on ne souhaite pas utiliser `idmap_rid`, le sous-système `LDAP` permet de centraliser les correspondances. Ainsi on gardera des uid cohérents dans le cadre d'une utilisation de NFS par exemple.

```
idmap backend=ldap:ldap://ldapdom.u-strasbg.fr
idmap uid = 10000-20000
idmap gid = 10000-20000
ldap suffix = dc=domaine,dc=local
ldap idmap suffix = ou=Idmap
ldap admin dn = cn=admin,dc=domaine,dc=local

template homedir = /home/%U
template shell = /bin/bash
winbind use default domain = yes
winbind separator = +

winbind enum users = yes
winbind enum groups = yes
```

Cet exemple stocke les correspondances dans le serveur ldap `idmap backend=ldap :ladp ://ldapdom.u-strasbg.fr` et dans le cas de création d'un nouvel *uid*, en prend un de libre dans l'intervalle spécifié 10000-20000. Les répertoires utilisateurs sont décrit de manière générique et le shell est le même pour tous les utilisateurs (pour l'instant, il n'est pas possible de faire autrement). On enlève également le domaine `winbind use default domain` pour n'utiliser plus que le login pour le login des utilisateurs (sinon les logins seraient `DOMAIN+user`)

Pour l'initialisation, il faut d'abord joindre l'ordinateur linux au domaine avec :
`net ads join`

Si un sous-système LDAP est utilisé, ne pas oublier de mettre en place le mot de passe pour accéder au LDAP avec `smbpasswd -w lepassduldap` qui permettra d'écrire dans la branche indiquée les correspondances.

Puis après avoir lancé le démon `winbindd`, on peut tester que on récupère bien les bonnes informations avec `wbinfo -t`
`wbinfo -u`

Enfin, il ne reste plus qu'à dire au système où trouver les informations NSS : Il faut rajouter `libnss_winbind` avec les autres librairies nss et rajouter `winbind` dans `nsswitch.conf`

```
passwd:     files winbind
group:      files winbind
```

On peut tester que le système trouve toutes ces informations : `getent passwd user` devrait renvoyer une ligne semblable celles de `/etc/passwd`

4.5 Postes en Dual-boot

Dans le cas des postes en dual boot, il faut veiller à ce que sous chaque système d'exploitation, un nom différent soit utilisé, car le SID doit correspondre à un nom unique dans le domaine Active Directory. Linux utilisant comme nom *netbios* le *hostname*, il sera préférable d'utiliser son nom DNS classique `pc.u-strasbg.fr` lors de l'adhésion au domaine et nommer autrement le poste sous Windows par exemple `pcW.domaine.local` (le domaine est de toute façon différent puisque Windows utilise par défaut celui du domaine Active Directory). Si les deux systèmes utilisent le même nom *netbios*, l'un des deux aura des problèmes d'authentification, et le DC remontera l'existence de noms en double dans son observateur d'événements.

Note : winbind sait également faire l'authentification dans le domaine Active Directory comme Kerberos, mais via des appels RPC. Il est donc possible de remplacer le module d'authentification `pam_krb5` par `pam_winbind` Cette façon est déconseillée, car les mises a jours Windows influencent beaucoup sur les appels RPC. En est la preuve, le dernier Service Rollup 1 du SP4 de Windows 2000, qui a nécessite de faire une mise à jour de tous les clients winbind...

5 Single Sign On

5.1 Données

Maintenant que nous avons une authentification unifiée, nous disposons déjà d'une solution de SSO : il suffit de tenter un montage d'un disque réseau sous Windows, ou utiliser `smbclient -k` depuis linux pour vérifier qu'on peut accéder à un service réseau juste en montrant un ticket Kerberos. C'est en fait un peu plus compliqué que cela : Le ticket initial (*Ticket Granting Tickets TGT*) ne permet que de demander d'autres tickets TGS (*Ticket Granting Services*), qui eux effectivement permettent l'accès aux services. Il y a donc une demande de ticket TGS faite par la station, en présentant le ticket TGT à un contrôleur de domaine. Celui ci, après vérification du TGT (il est le seul à pouvoir le faire) donnera un TGS au client qui servira uniquement à accéder au service demandé par ce client. Cet échange est transparent pour l'utilisateur.

5.2 Service Web apache

L'intégration d'un service web pour faire une authentification basée sur Kerberos, et être ensuite compatible SSO, nécessite quelques manipulations supplémentaires. Tout d'abord il faut déclarer le service HTTP dans le domaine Active Directory : Il faut, pour cela, d'abord créer un utilisateur fictif pour l'hôte dans le domaine Active Directory (*New - Users*) [6].

Ensuite il faut rajouter un service HTTP à cet utilisateur : Dans une fenêtre `cmd` :

```
ktpass -princ host/msw.domaine.local
@DOMAINE.LOCAL -pass password -mapuser
msw -out msw.host.keytab
```

```
ktpass -princ HTTP/msw.domaine.local
@DOMAINE.LOCAL -pass password -mapuser
msw -out msw.HTTP.keytab
```

Les deux fichiers créés sont à mettre sur le serveur web, le premier en tant que `/etc/keytab` et le second pour apache.

Il faut rajouter au serveur apache le module `mod_auth_kerb` pour faire l'authentification. Il suffit de rajouter dans son fichier de configuration le chemin d'accès au fichier keytab

```
AuthType Kerberos
KrbAuthRealm DOMAINE.LOCAL
Krb5Keytab "/etc/httpd/conf/keytab"
KrbServiceName HTTP
KrbMethodNegotiate on
KrbMethodK5Passwd on
```

Et nous voici avec un module permettant d'authentifier les utilisateurs via Kerberos. Pour finir notre SSO, il suffit de cocher la case «Transmettre l'authentification intégrée» dans la configuration de *Internet Explorer*, ou de rajouter l'adresse du serveur dans la variable `network.negotiate-auth.trusted-uris` de *Mozilla*

Autre service utilisé, le CMS *drupal* permet de faire une authentification automatique des personnes quand elles arrivent sur le site. Il faut juste autoriser l'auto-création des utilisateurs dans sa configuration, et désactiver la boîte d'authentification manuelle.

5.3 Autres services

Bien d'autres services savent tirer profit des tickets Kerberos. Le prochain gros projet sera sans doute NFS4 qui apportera enfin de la sécurité, et se basera sur Kerberos pour cela. Des programmes comme *putty* sous Windows sont compatible Kerberos et *openssh* version 4 et plus savent utiliser les tickets. les serveurs mails ne sont pas en reste avec wu-imap par exemple, et horde en tant que webmail.

6 Intégration SUPANN

Les projets retenus par le ministère dans le cadre des projets SUPANN ont un certain nombre de points volontairement communs. Le premier d'entre eux est l'utilisation de CAS pour effectuer les authentifications des applications web. CAS peut utiliser différents sous systèmes, le plus courant qui a été utilisé est un serveur LDAP (d'autres supports tels que LDAP+Kerberos ou Active Directory étaient possible mais aucun projet n'a retenu ces solutions).

En ce qui concerne le projet de notre université, c'est EPPUN [7] qui est le projet d'Espace Numérique de Travail. L'authentification utilise donc un mot de passe simplement crypté `userPassword` comme dans le fonctionnement des systèmes Unix. Dans le cadre de la simplification des services aux utilisateurs, il faudrait essayer de s'intégrer aux projets actuels pour faire un authentification commune et si possible compatible SSO.

Ce projet étant fédérateur, il n'est pas possible de faire des modifications dans leur authentifications, c'est donc de notre coté qu'il faut effectuer des modifications. Étant donné également qu'il manque un certain nombre d'informations nécessaires dans l'annuaire LDAP pour les utilisateurs, par exemple les répertoire de travails, les shells, uid, etc, à moins d'utiliser des *overlay* de openLDAP 2.3, le LDAP n'est pas utilisable directement.

6.1 Principe

Deux problèmes se posent : La synchronisation des logins , et celles des mots de passe.

La solution retenue au département d'informatique de l'IUT Robert Schuman a été de modifier la *GINA (Graphical Interface for Network Authentification)* [8] de Windows : On présente une mire de connexion avec deux champs login/passwd, et l'utilisateur s'authentifie sur le LDAP.

– Si cette authentification échoue, on transmet le couple (login/pass) à la GINA de Microsoft qui elle va tenter de faire une authentification dans le domaine. Si l'utilisateur s'est bien authentifié, une ouverture de session standard de Windows est effectuée.
– Si l'utilisateur s'authentifie correctement sur le LDAP, alors un compte avec le même login est crée dans le domaine, et le mot de passe associé est forcé au même mot de passe entré puis une seconde authentification est faite sur la GINA de Microsoft pour obtenir un ticket Kerberos (si le login existait déjà, seul son mot de passe est forcé).

Ainsi on a toujours synchronisations des utilisateurs du LDAP dans le domaine Active Directory. L'avantage est que une fois authentifié, l'utilisateur l'est aussi dans le domaine, donc garde tous les avantages décrits précédemment de l'authentification Kerberos.

6.2 Configuration et intégration EPPUN

Pour effectuer cela, nous avons utilisé *pGina* de XpaSystem [9] avec le plugin LDAPAUTH [10].

Il faut également un domaine Active Directory en mode *mixte* : la création des comptes dans *pGina* se fait par appel RPC ce qui n'est possible qu'en mode mixte. La création en mode natif est en théorie possible via LDAP, mais est beaucoup plus lourd à développer et à configurer aussi bien du côté de *pGina* que côté serveur.

Nous avons également activé le «Domain management» ce qui permet de créer les utilisateurs dans le domaine et d'authentifier des personnes non présentes dans les annuaires d'établissements, ce qui est très pratique pour les stagiaires ou autres invités ayant besoin d'un compte.

Seul point manquant : on fait une authentification vers l'annuaire LDAP et non pas directement sur le serveur CAS. Du coup, il n'y a pas de cookie pour le navigateur qui aurait pu permettre de faire du SSO directement sur les applications CAS-SSO. En attendant, on peut déjà profiter du SSO sur les applications classiques et web de la composante !

7 Conclusion

Le point principal à retenir est que les domaine Active Directory utilisent Kerberos pour faire l'authentification. Cela reste un point largement méconnu des ingénieurs. Kerberos en lui même apporte beaucoup d'éléments, de possibilités, de services, de sécurité. Il existe depuis des années mais n'a pourtant jamais vraiment percé (en fait si, il a percé grâce à Windows 2000 mais vous ne le découvrez que maintenant). Son principal inconvénient est sa configuration. C'est un outil Unix, et il faut énormement de configuration : chaque ordinateur doit avoir un compte, chaque service doit avoir un compte relié à l'ordinateur, et même en l'automatisant, cela reste compliqué. A l'époque, pour NIS, on ajoutait un utilisateur dans un fichier et il suffisait de reconstruire les tables.

Kerberos permet de faire ici ce qu'on a toujours voulu faire, et qu'on a fait de manière compliquée depuis des années : une authentification unique pour tous les ordinateurs. En prime, on bénéficie du SSO : une seule authentification lors de l'ouverture de session et on n'a plus à se ré-authentifier pour les autres applications compatibles. Un gain en terme de sécurité pour l'ingénieur, un gain de simplicité pour l'utilisateur.

Enfin l'intégration dans les projets SUPANN a été également possible sans perte de fonctionnalités. Notons que tous cela n'est possible que grâce à des programmes open source, (winbind de samba, pGina de XPASystem) ainsi qu'à l'implication et la disponibilité des auteurs de ces logiciels.

Pour finir, certains verront dans ce papier une place trop critique de Windows en tant que serveur d'authentification. Qu'ils se rassurent, il reste possible de faire déléguer la partie Kerberos de Active Directory à un serveur KDC Unix [11].

Références

[1] Denis Ducamp. Cassage et durcissement des mots de passe - première partie : Windows. Dans *Groupe SUR de l'OSSIR*. Hervé Schauer Consultants, 2002. http ://www.hsc.fr/ressources/articles/mdp_misc2.

[2] Samba Team. http ://www.samba.org/.

[3] John H. Terpstra Gerald J. Carter, Jelmer R. Vernooij. *The Official Samba-3 HOWTO and Reference Guide*. 2005. http ://www.samba.org/samba/docs/man/Samba-HOWTO-Collection.

[4] Christian Martin. Gestion des comptes et des groupes commune à unix et windows 2000. Dans *Actes JRES 2001*, 2001.

[5] MIT Kerberos. http ://web.mit.edu/kerberos/www/.

[6] Microsoft. Developing Heterogeneous Kerberos Security Solutions. Dans *Microsoft Solution Guide for Windows Security and Directory Services for UNIX*, 2004. http ://www.microsoft.com/technet/itsolutions/cits/ interopmigration/unix/usecdirw/07wsdsu.mspx.

[7] Service Interuniversitaire d'Informatique et de Gestion. Eppun. 2005. http ://eppun.u-strasbg.fr/.

[8] Keith Brown. Customizing the Graphical Identification and Authentication Component. 2005. http ://msdn.microsoft.com/msdnmag/issues/05/05/.

[9] Nathan Yocom. pGina. http ://www.xpasystems.com/.

[10] Micah Cooper. LDAP authentication plugin for pGina. http ://pgina.xpasystems.com/plugins/ldapauth.php.

[11] Microsoft. Using an MIT KDC with a standalone windows 2000 workstation. Dans *Step-by-Step Guide to Kerberos 5 (krb5 1.0) Interoperability*. http ://www.microsoft.com/windows2000/techinfo/planning/ security/kerbsteps.asp.