



---

# Intégration d'un poste Linux dans un domaine W2K

---

Pascal Gachet –EIVD  
pascal.gachet@eivd.ch  
mai 2003

---

## Table des matières

Introduction.....	2
Terminologie .....	3
Authentification Kerberos.....	4
Samba.....	5
Module PAM.....	6
Compte pour poste Linux.....	7
Accès à un fichier sur un poste Linux .....	7
Conclusion.....	8
Références .....	9

## Introduction

Les systèmes Linux devenant de plus en plus répandus dans les entreprises, le besoin d'interopérabilité entre système Linux et Windows est devenu une réelle nécessité.

Actuellement, la plupart des parcs informatiques utilisent largement W2K. Cet environnement met à disposition une authentification centralisée utilisant le protocole Kerberos, ainsi qu'un service d'annuaire basé sur le standard ldap<sup>1</sup> (Active Directory).

Un utilisateur disposant d'un *account* dans l'annuaire Active Directory et des *credential* (login, mot de passe) dans le domaine W2K pourra bénéficier d'une authentification unique pour tous les services W2K disponibles dans le domaine, et cela suivant la politique d'autorisation définie dans Active Directory.

En contre partie, pour accéder à un serveur Linux, l'utilisateur W2K devra s'authentifier à nouveau pour chaque service étant donné que chaque service Linux requiert une authentification propre. De même, un utilisateur Linux devra se ré-authentifier pour chaque service W2K, pour autant que le service en question le lui permette.

En résumé, la problématique est la suivante :  
L'administration d'un parc hétérogène est lourde.

- Il est nécessaire de créer sur chaque serveur Linux les *account* pour les utilisateurs W2K.
- Pour chaque service Linux, créer une politique d'autorisation pour chaque utilisateur
- Redondance d'information entre les *account* Linux et Windows, les *account* ne sont pas toujours à jour.
- Politique de sécurité rarement identique sur les deux systèmes.

L'utilisation de serveur hétérogène n'est pas transparente.

- L'utilisateur W2K doit se re-authentifier pour chaque serveur Linux.
- L'utilisateur Linux doit se re-authentifier pour chaque serveur Linux.

Il existe différentes méthodes pour résoudre cette problématique.

---

<sup>1</sup> Light Directory Access Protocol

- Administrer tous les serveurs Linux à l'aide d'un contrôleur de domaine Kerberos tournant sur un poste Linux et utiliser l'implémentation OpenSource de ldap (openldap), puis effectuer un trust entre le domaine Linux et W2K.
- Intégrer les machines Linux directement dans le domaine W2K en utilisant le package *Service for Unix* de Microsoft.
- Intégrer les machines Linux dans le domaine W2K en substituant les mécanismes propriétaires de Microsoft par des implémentations OpenSource (openKerberos, openldap, samba, pam).

Si la possibilité de disposer d'un domaine distinct pour chaque type d'environnement paraît séduisante, elle est toutefois envisageable uniquement avec un domaine Windows tournant uniquement sur NT4, étant donné que W2K ne permet plus de trust avec des domaines non W2K.

Utiliser le package *Service for Unix* est sans conteste la solution la plus efficace dans le cas d'une administration à large échelle, mais elle repose sur l'utilisation d'implémentations propriétaires de Microsoft non conformes à la politique des projets R&D.

La solution pour intégrer un poste Linux dans un domaine W2K se basera donc sur une intégration OpenSource de tous les mécanismes de Microsoft. C'est-à-dire :

- Authentification centralisée par Kerberos.
- Utilisation d'openldap pour accéder à Active directory.
- Utilisation de samba pour permettre une communication entre les postes Linux et Windows.
- Utilisation des modules Pam pour centraliser les credential dans le cas d'un utilisateur Linux.

## Terminologie

Avant d'entrer plus en détail dans la description de la solution, il est souhaitable de consacrer quelques lignes à la spécification de la terminologie qui sera utilisée dans ce document. Notamment la distinction entre poste client et poste serveur.

Dans la terminologie Microsoft, la distinction entre poste serveur et poste client est faite de façon évidente, étant donné qu'une licence particulière est nécessaire pour bénéficier d'un serveur Microsoft.

Les serveurs Microsoft du type IIS, FTP, AD etc. ne sont disponibles qu'avec la distribution *Microsoft Server Family*. Puisque les utilisateurs standard utilisant une distribution *Microsoft Professionnel* ne disposent pas de ce type d'application.

Dans le monde Linux, cette distinction est moins évidente étant donné qu'il n'existe pas de licence limitative. Ainsi, un poste Linux pourra aussi bien être utilisé comme poste utilisateur que comme poste serveur.

Pour rester rigoureux dans ce document, le nom serveur ne sera jamais utilisé pour décrire un poste physique contenant des serveurs, le nom serveur ne sera employé que pour définir une application fournissant un service (du type daemon Unix). Pour définir la machine physique sur laquelle tournent ou ne tournent pas des serveurs, le nom **poste** sera employé.

## Authentification Kerberos

La première étape pour intégrer un poste Linux dans un domaine W2K consiste à intégrer sur le poste Linux les mécanismes d'authentification propre à Microsoft W2K.

Microsoft W2K utilise le protocole d'authentification centralisée Kerberos v5. Une implémentation libre de cette application, développée par le MIT, existe en OpenSource pour Linux sous le nom de `krb5`. Cette application permet d'être configurée comme client ou serveur. Dans le cas présent, seule la fonctionnalité cliente est nécessaire étant donné que le serveur Kerberos est assuré par le contrôleur de domaine W2K.

La configuration de ce client est relativement simple ; il est nécessaire d'indiquer le nom DNS du domaine et le nom DNS du serveur d'authentification kerberos.

L'utilitaire `kinit` permet d'effectuer une requête de ticket TGT<sup>2</sup> formatée auprès du contrôleur de domaine W2K, celui-ci répondant par un ticket TGT (figure 1).

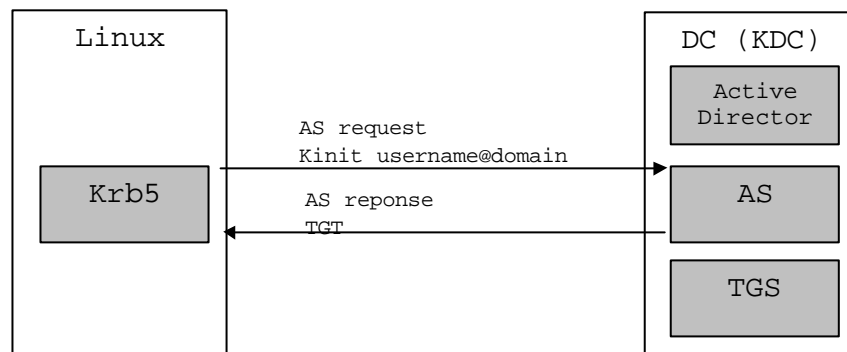


Figure 1 `kinit`

A ce stade, le poste Linux dispose d'un ticket TGT pour demander un ticket de service, mais le protocole Kerberos ne fournit pas d'autre mécanisme que l'authentification. Il est donc absolument nécessaire d'intégrer un protocole de communication entre les machines Windows et Linux.

Le protocole de communication entre les machines Windows se base sur l'implémentation propriétaire `smb`<sup>3</sup>. Ce protocole utilise une API, nommée GSS-API<sup>4</sup> qui utilise le ticket d'authentification obtenue précédemment pour authentifier tous les messages échangés entre les machines.

<sup>2</sup> Ticket Granting Ticket

<sup>3</sup> Simple Message Block

<sup>4</sup> Generic Security Service Application Programming Interface

## Samba

Pour rendre inter-opérable l'accès à des fichiers partagés entre Windows et Linux, une équipe de travail OpenSource a développé une version libre de smb, nommée Samba.

Cette application a été obtenue par *reverse engineering* en analysant et en implémentant le protocole propriétaire smb. La dernière version (samba 3.0) permet en plus d'utiliser les mécanismes d'authentification Kerberos. Samba, dans sa version 3, implémente GSS-API, c'est-à-dire qu'une fois authentifié au domaine W2K, l'utilisateur Linux est en mesure d'accéder à des fichiers partagés sur un poste W2K de façon transparente (Figure 2).

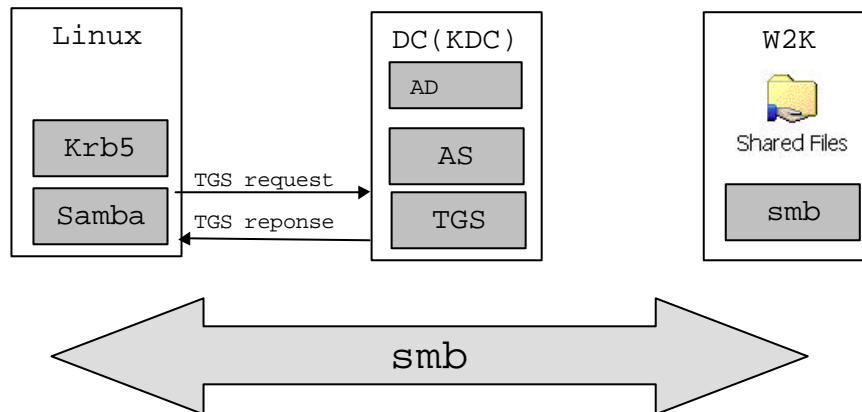


Figure 2 Samba

Pour accéder à un fichier partagé sur un poste W2K, Samba disposant déjà d'un ticket TGT (figure 1) utilise à nouveau krb5 pour effectuer une requête de ticket pour accéder au serveur smb en question. De cette manière un utilisateur Linux est en mesure de monter son disque W2K ou tout autre élément partagé de façon transparente, avec la commande suivante.

```
smbclient //w2k/Shared Files\$ -k
```

L'option `-k` spécifie une authentification kerberos. Si le client Linux dispose déjà d'un ticket pour le poste W2K en question, aucune authentification supplémentaire n'est requise. Dans le cas contraire, le contrôleur de domaine invite le client à s'authentifier.

L'utilisateur Linux est soumis au même droit d'accès que n'importe quel autre utilisateur Windows, étant donné que l'utilisateur fait partie intégrante d'active directory.

Si cette solution est déjà très séduisante, elle n'est pas encore parfaitement raffinée. Dans un domaine W2K, la plupart des utilisateurs possèdent un espace disque personnel sur le domaine. Cet espace est visible au moment où ces utilisateurs se sont logués. Pour un utilisateur Linux, l'accès à cet espace disque est un peu plus lourd. En effet, l'utilisateur Linux doit en premier lieu se loguer localement sur sa machine, puis monter son espace disque par la commande présente. L'étape suivante consiste naturellement à supprimer le login local pour un client Linux. Le client Linux ne disposera donc plus d'*account* local sur sa machine, mais tout devra être effectué par Kerberos.

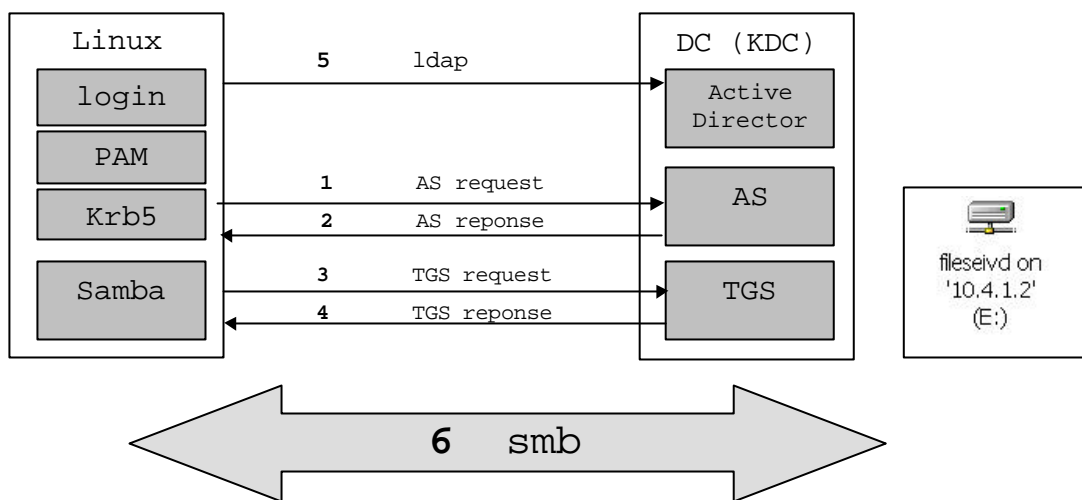
## Module PAM

L'environnement Linux fournit un mécanisme élégant permettant de gérer la problématique d'authentification. Il s'agit des modules PAM<sup>5</sup>.

PAM, constitue une couche d'authentification générique supplémentaire qui peut être redéfinie. Ainsi chaque service compilé avec les bibliothèques PAM est en mesure de déléguer l'authentification à la couche PAM. PAM permet différentes méthodes d'authentification dont l'authentification par Kerberos définie dans le module `pam_krb5`.

Pour se loguer et accéder à son espace disque W2K, l'utilisateur suivra la séquence suivante :

1. L'application linux responsable du login local à la machine transmet l'ordre à la couche PAM. Cette couche s'authentifie auprès du serveur d'authentification AS.
2. Le serveur d'authentification AS vérifie les credential du client et transmet le ticket TGT.
3. La couche PAM effectue une requête auprès du TGS pour obtenir un ticket pour active directory.
4. Le serveur TGS<sup>6</sup> vérifie que l'utilisateur en question a les droits d'accès nécessaires et transmet le ticket pour Active directory.
5. Le module PAM utilise le module ldap pour vérifier que l'utilisateur existe dans Active Directory, si tel est le cas le poste linux est logué dans le domaine W2K.
6. L'espace disque correspondant à l'utilisateur est automatiquement monté en utilisant samba et le ticket d'authentification obtenu au point 2.



**Figure 3** Accès à un disque

Si cette interopérabilité est possible pour un utilisateur Linux, elle doit également être possible dans l'autre sens. C'est-à-dire un utilisateur W2K désirant accéder à un dossier partagé sur un poste Linux. Toutefois, avant de pouvoir effectuer une telle opération, il est nécessaire d'introduire un account pour le poste Linux dans l'annuaire Active directory

<sup>5</sup> Pluggable Authentication Modules for Linux

<sup>6</sup> Ticket Granting Service

---

## Compte pour poste Linux

Un élément marquant du protocole Kerberos est son mécanisme d'authentification centralisée, rendant possible la réutilisation des tickets pour différent service sans devoir se ré-authentifier. Si cette transparence est tout à fait visible pour l'utilisateur, un élément moins visible de ce protocole est lui aussi tout aussi intéressant. Il s'agit de l'authentification mutuelle des deux tiers, c'est-à-dire que le protocole définit aussi une authentification serveur. De façon traditionnelle, pour intégrer un nouveau service Linux dans l'annuaire Active Directory la procédure est la suivante.

- Créer un compte computer pour le poste Linux dans Active Directory
- Depuis le contrôleur de domaine, créer un fichier *keytab* contenant les *credentials* pour le service Linux. Les *credentials* pour le service Linux sont automatiquement ajoutés à la liste des mots de passe de Kerberos.
- Transmettre le fichier *keytab* sur le poste Linux, puis intégrer le fichier *keytab* à la liste des fichiers *keytab* de la machine Linux.

Le fichier *keytab* contient le nom du service ainsi que le mot de passe du service. Le contrôleur de domaine dispose également de ces informations. Le service est donc en mesure de s'authentifier de façon transparente.

Avec Samba, la procédure est simplifiée de façon sensible.

Elle suit exactement la même politique que pour intégrer un poste W2K dans un domaine W2K, en effectuant tout depuis le poste client.

- Samba fournit un outil permettant de se connecter par ldap à Active Directory, en tant qu'administrateur de domaine. Un compte computer est créé.
- Un fichier *keytab* est généré de façon transparente sur le contrôleur de domaine, puis intégré de manière tout aussi transparente dans la liste des *keytabs* du poste Linux.

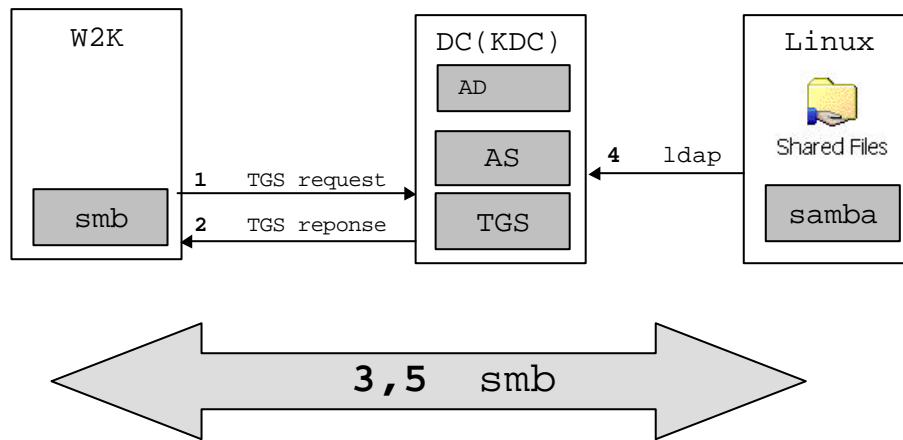
## Accès à un fichier sur un poste Linux

L'étape finale de cette intégration consiste à pouvoir accéder depuis un poste W2K à un fichier partagé sur un poste Linux, en utilisant l'authentification Kerberos.

La figure met en évidence le processus :

1. Le client W2K dispose déjà d'un ticket TGT étant donné qu'il s'est déjà logué dans le domaine. Le client effectue une requête pour accéder à un service samba sur le poste linux.
2. Le contrôleur de domaine vérifie les droits d'accès du client, puis lui retourne un ticket pour le service samba.
3. Le client accède au service samba par le protocole smb.

4. Le service Linux vérifie que le client existe en interrogeant l'annuaire Active directory.
5. Le client accède au fichier partagé.



**Figure 4** Accès à un répertoire Linux

On constate dans cette figure que le client W2K n'a dû en aucun cas s'authentifier au service Samba. Le client W2K n'a donc aucun moyen de détecter que le poste auquel il accède est un poste non-W2K.

## Conclusion

L'intégration d'un poste linux dans un domaine W2K a permis de résoudre de façon significative l'administration d'un parc hétérogène en ce qui concerne la gestion des comptes utilisateurs et de leur mot de passe. Toutefois un grand nombre de problèmes administratifs restent en suspend, les postes Linux n'étant évidemment pas soumis à la même politique de sécurité que les postes W2K. Les GPO qui sont appliquées au poste W2K ne s'appliqueront pas au postes Linux. Il n'est pas non plus possible d'administrer (manage) les postes Linux depuis le contrôleur de domaine.

Dans ce document, la possibilité de rendre samba compatible avec Kerberos a été démontrée. Mais cette possibilité existe également pour un grand nombre de serveur Linux, notamment FTP, Telnet, SSH, etc. De plus, des classes java implémentant le client Kerberos et l'API GSS-API sont disponibles sur Internet. Il est donc possible de bénéficier d'une authentification centralisée pour toutes les nouvelles applications développées. Cette possibilité est de plus en plus prise en compte dans les applications client serveur de l'institut TCOM, notamment pour l'authentification aux serveurs Web ou plus récemment, pour authentifier les clients *Voice over Ip* basés sur le protocole SIP .

L'authentification centralisée de Kerberos soulève malgré tout un dilemme qui doit être mentionné. En déléguant l'authentification de tous les serveurs à Kerberos, on introduit un point critique dans toute la politique de sécurité du réseau. Le poste sur lequel repose Kerberos doit être particulièrement protégé, car il est la clé de voûte de tout le système

---

d'authentification. Un simple déni de service engendrerait une panne de tous les services ou pire une intrusion sur cette machine pourrait compromettre toute la confidentialité du système.

Dans le cadre d'une politique de sécurité plus élevée, il serait tout à fait envisageable d'ajouter une authentification forte PKI au protocole Kerberos. (Voir document *Intégration d'une PKI tierce(OpenCA) dans un domaine W2K*). L'authentification des clients aux différents services se ferait par signature *pkcs7*, ce passant donc complètement de Kerberos.

## Références

- [1] Samba Project Documentation  
<http://www.samba.org>
- [2] Samba 3.0 prealpha guide to Kerberos authentication  
<http://www.samba.org>
- [3] TechProGuild, Join a Linux server to Active Directory with samba 3.0  
<http://builder.com>
- [4] Microsoft, Step-by-Step guide to Kerberos 5  
<http://www.microsoft.com/WINDOWS2000/techinfo/planning/security/kerbsteps.asp>
- [5] MIT, Kerberos V5 Installation Guide  
<http://web.mit.edu/kerberos/www/krb5-1.2/krb5-1.2.3/doc/install.html>
- [6] Kerberos Testing MIT and Windows2000 Interoperability  
<http://www.colorado.edu/its/windows2000/itsresources/kerbtest3.pdf>
- [7] Kerberos and PAM  
[http://www.redhat.com/docs/manuals/linux/RHL-7.3-Manual/ref\\_guide/s1-kerberos/pam.html](http://www.redhat.com/docs/manuals/linux/RHL-7.3-Manual/ref_guide/s1-kerberos/pam.html)