Active Directory and Linux Identity Management

Published by the Open Source Software Lab at Microsoft. December 2007.
Special thanks to Chris Travers, Contributing Author to the Open Source Software Lab. Most current version will be maintained at http://port25.technet.com.

Abstract:
This paper is written for a technical audience and covers how the identity management expectations are different between Windows and Linux and how Windows Server can be used to manage both. I assume that the reader is familiar with general Windows administration tasks, such as user management.

1 Introduction

Windows has a reputation of being a bit difficult regarding interoperability of network log-in and identity management. Many of the problems have to do with security internals relating to how network access and identity has been managed on Windows since the early NT days and how this has differed from POSIX environments. While there are strong technical merits to Microsoft's approach, they have resulted in some degree of conflict with POSIX systems. However, with the development of Active Directory and the subsequent adoption of LDAP for directory services and Kerberos V for authentication, these problems are now quite manageable. In fact, Windows Server 2003 R2 vastly reduces the amount of complexity in implementing Windows-based identity management solutions for Linux.

Even so, there are a number of areas where the systems work in fundamentally different ways. Active Directory by default does not store the sorts of information UNIX and Linux systems expect to find associated with users, and there are a number of other areas where unexpected differences may surface.

This paper is written for a somewhat technical audience and covers how the identity management expectations are different for these two platforms and how Windows Server can be used to manage both. I assume that the reader is familiar with general Windows administration tasks, such as user management.

1.1.1 Understanding Windows Identity Management and Active Directory

Windows internally uses a value called a SID[1], which is a globally unique value. This value identifies each authenticated principal, whether a computer, user, or group, across not only an entire domain but world-wide. This is because UUID[2]'s are designed to be guaranteed unique world-wide, at least in theory (or at least the chance of a collision is negligible). SID's, due to their globally unique nature, do not need to be locally synchronized.

Internally, Active Directory attaches a second UUID to authenticating principal that is generally referred to as the GUID[3]. Unlike the SID, this UUID is valid for the life of the object. The SID's can change when users or machines are transferred from one domain to another in a forest, but GUID's remain the same.

All access control lists utilize SID's and GUID's. These are the basis for all aspects of identification management in Windows.

In Active Directory, LDAP stores all information including password hashes and a Kerberos interface is provided for domain authentication management. However, because it is possible to authenticate against the LDAP directory separately, Kerberos is not actually required to access an Active Directory identity. Yet, Kerberos is tightly integrated into the LDAP back-end in Active Directory, and the log-in credentials returned by the server include basic identity management information.

[1] Security IDentifier
[2] Universally Unique IDentifier
[3] Globally Unique IDentifier

1.1.2 Understanding Linux/UNIX Identity Management

In some ways, the approach of Linux and UNIX is the polar opposite of the Windows approach in these areas. Unlike a single integrated system, Linux and UNIX use modular components which can be reconfigured even on a running system. In particular, authentication and identity retrieval are fundamentally separate. While this approach leads to greater flexibility, it also can be more complex to implement. In some cases, bad implementations can have unfortunate security implications.

In Linux and UNIX, identity management are managed through integers known as UID (User ID) and GID (Group ID). These are not guaranteed to be unique across a network and must be synchronized through some authoritative source. Small networks may be able to replicate local files managing these things, but larger networks are likely to turn to other frameworks, such as NIS and LDAP to manage these numbers. Note that these non-unique integers are used to identify user and group rights, so if the numbers change, users may be unintentionally granted or denied permissions to various files.

The actual configuration on Linux for the identity look-up is handled by a utility called Name Service Switch or NSS. This module allows for an administrator to reconfigure the source of UID/GID and other information on the fly. This module is not to be confused with Network Security Services, which is an implementation of SSL maintained by the Mozilla Foundation and is also distributed with many Linux distributions. To pull identity information from Active Directory, we will use the LDAP NSS modules.

In Linux, authentication is managed using user-space programs called Pluggable Authentication Modules (PAM). PAM-enabled applications pass on authentication requests to the system, which is then able to pass on the requests to modules capable of handling the actual authentication. This allows the mechanism for authentication to be switched even while the system is running. We will use the PAM's for Kerberos to authenticate against Active Directory.

1.1.3 Windows System Configuration

Windows Server 2003 was configured for this example as a domain controller for the krbtest.local domain. It was configured to support pre-Windows 2000 clients. This allows for anonymous queries of some parts of the LDAP directory. The support tools were installed (via D:\Support\Tools\SupTools.msi, where D:\ is the root directory of the Windows Server 2003 install CD).

1.1.4 Installed Packages on Linux

The following packages and their dependencies were installed on Fedora Core 5 for this example[4] in order to make this configuration work:

- nss_ldap
- pam_krb5
- samba_client
- samba_common

[4] On other Linux and Unix distributions this will/may differ.

1.2 Initial Linux Setup

1.2.1 Configuring Kerberos

The following line was added to the /etc/krb5.conf file's [libdefaults] section:

```
default_keytab_name = FILE:/etc/krb5.keytab
```

The following two lines were added to the [realms] section of the same file:

```
KRBTEST.LOCAL = {
  kdc = kdc1.krbtest.local:88
  default_domain = krbtest.local
}
```

1.2.2 Joining the Domain

Add the following lines to the [globals] section of the /etc/samba/smb.conf file to configure the Samba client:

```
netbios name = chrislt
realm = KRBTEST.LOCAL
security = ADS
encrypt passwords = yes
password server = kdc1.krbtest.local
workgroup = KRBTEST
use kerberos keytab = yes
```

Join the domain with the following command:

```
bash# net ads join -U Administrator
```

Enter the Administrator password when prompted.

1.2.3 Generating the Keytab

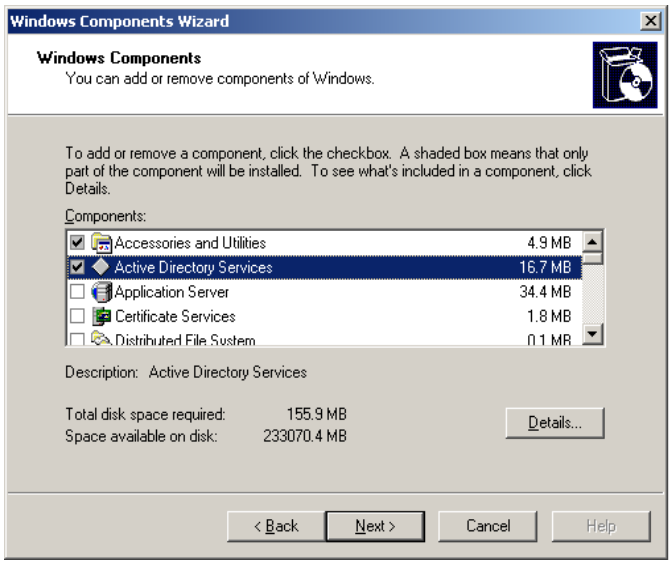
To generate the keytab, use the following command:

```
bash# net ads keytab create -U Administrator
```

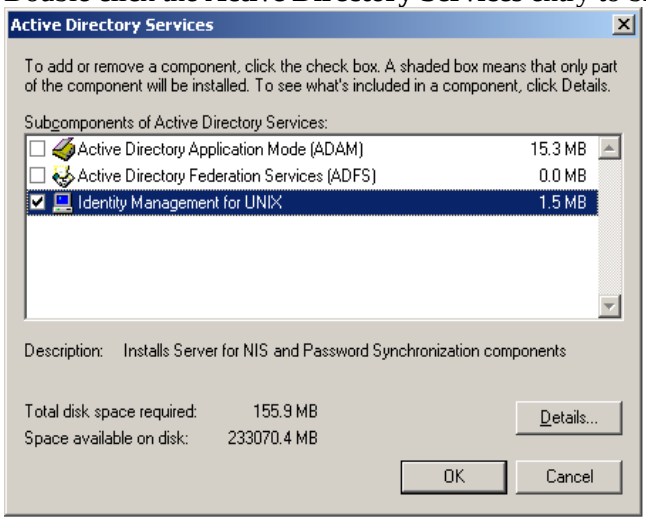
1.3 Windows Setup

1.3.1 Installing Identity Management for UNIX

In Windows Server 2003 R2, the Identity Management solution can be installed from the Add/Remove Programs applet in the control panel. Select the Windows Components option and the following screen will appear:



Double click the **Active Directory Services** entry to bring up the next screen.



Check the **Identity Management for UNIX** option and click **OK**. This will install the extensions for managing POSIX identity values in Active Directory. Unlike previous approaches involving Microsoft Windows Services for UNIX (SFU), this extension is standards-compliant and requires very little configuration on the Linux side to make work.

1.3.2 Configuring the Group

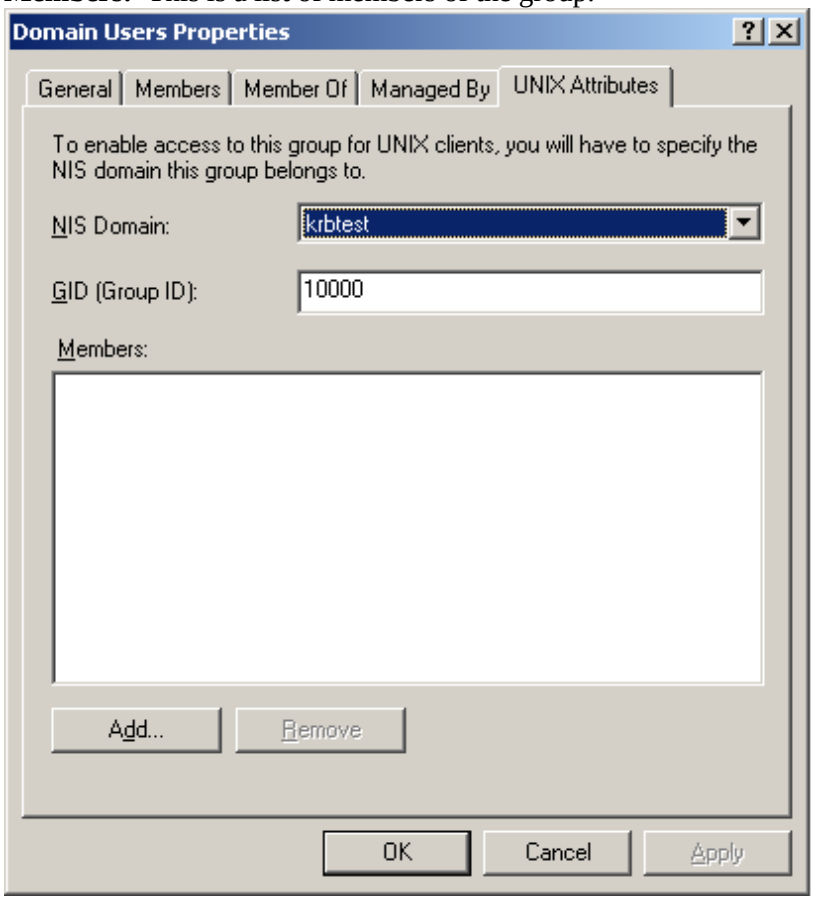
When the Identity Management for UNIX add-on is installed, a new **UNIX Attributes** tab appears in the Group properties applet. These should be configured before you configure your users because every user must be given a primary group.

Many Linux systems today create one group per user as the primary group. The actual design of the directory services though is beyond the scope of this paper. Specific environments may require specific solutions.

The fields in the new tab are as follows:
NIS Domain: This field is similar in function to the domain name for pre-Windows 2000 clients.

GID: This is the numeric value used to specify group ownership or file access rights in the Linux or UNIX system.

Members: This is a list of members of the group.



1.3.3 Configuring a User

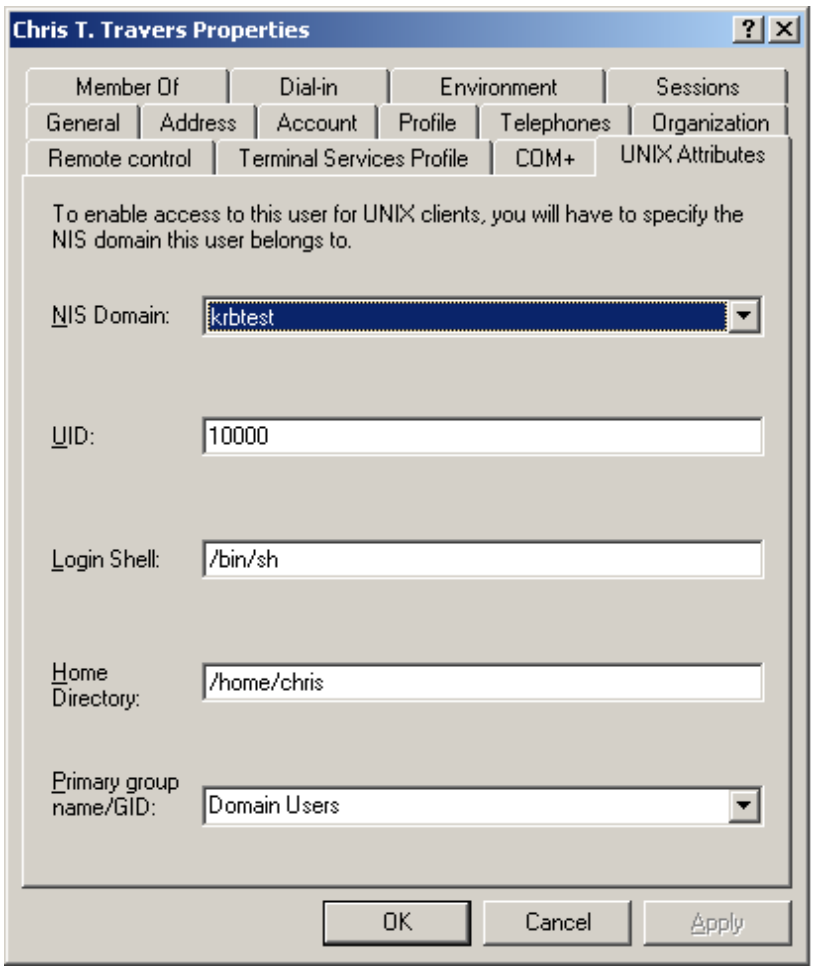
Similarly, the User Configuration screen contains a new **UNIX Attributes** tab which is used to store additional configuration information. The additional fields have the following meanings:

NIS Domain: This has a similar meaning to the old NT4 domain label. It is usually the final part in the fully qualified domain name.

UID: This is the numeric user ID used by Linux or UNIX for file ownership and system permissions enforcement.

Login Shell: This is the full path to the interactive shell for the account. Usually this is /bin/sh, but can be set to many other values depending on what the Linux system supports. To deny login on Linux systems, set this to /bin/nologin or /bin/false.

Primary Group Name/GID: When a user creates a file, those files are typically owned by the primary group of the user. This option specifies which group owns the files.



Note that if your UID and GID values should match any solution you already have in place. These are unique values, and once used, should not be changed.

1.4 Linux Final Setup

Once the above steps have been completed, all that is left to do is to configure Linux to authenticate properly against Windows and use Active Directory's LDAP mechanism for identity management.

1.4.1 Configuring PAM

To configure PAM to use Kerberos first and local accounts second, use the following command:

```
bash# authconfig --enablekrb5 --update
```

This utility will properly reconfigure the system, rewriting appropriate PAM configuration files.

1.4.2 Configuring NSS

I created an /etc/libnss-ldap.conf with the following configuration:

```
host kdc1.krbtest.local
base dc=krbtest,dc=local
ldap_version 3
```

```
binddn cn=User CN,cn=Users,dc=krbtest,dc=local
bindpw password
scope sub
timelimit 30
nss_map_attribute homeDirectory unixHomeDirectory
nss_map_objectclass posixAccount user
nss_map_objectclass shadowAccount user
```

"User CN" and "password" were changed to appropriate fields Note that the posixAccount and shadowAccount classes are merged into one "user" class in Active Directory. Furthermore, homeDirectory is replaced by unixHomeDirectory. hence the need for the mappings.

I changed the following lines in the nsswitch file:

```
passwd:     files
shadow:     files
group:      files
```

to:

```
passwd:     ldap files
shadow:     ldap files
group:      ldap files
```

These lines tell the name service switch to look up user or group information first in LDAP and if it is not found, consult local files.

1.4.3 A brief note on the Advances since Services for UNIX

In Microsoft Services for UNIX 3.5, Microsoft extended the Active Directory schema so that it could be used as the backend for a NIS server. In order to ensure that nothing conflicted, they added an msSFU prefix to the beginning of all the names. This meant that a great deal of mapping had to be done between the Linux host and the Windows host to make the identities work correctly.

In Identity Management for UNIX, a different approach is taken. The names of the object classes and attributes are defined in RFC 2309 for the most part. Although these are changed in a few cases, those are rare, reducing the number of mappings required to 3.

1.5 Final Thoughts

The Identity Management for UNIX component represents a great step forward in terms of security interoperability. It is fairly straight-forward to set up, and it is far more standards-compliant than past versions.

Identity management is a difficult problem in a heterogeneous environment, especially ones as diverse as Windows and Linux. n general, I think that Active Directory is a good solution to the problem and that Microsoft has shown a commitment to making the software work well as a universal directory infrastructure for a corporate organization.

1.6 About the Author

Chris Travers is the owner of Metatron Technology Consulting, a business specializing in helping businesses leverage open source software. He has extensive experience with security and identity management in both Linux and UNIX.