

Using native LDAP, native Kerberos and Windows Server AD services and schema for cross platform identity management

Tech Note #4 in the "Identity Management" series

By Doug Miller (Interop Systems)

Scenario

The scenario for this tech note is as follows. You have Active Directory installed on Windows Server 2008 or Windows Server 2003 R2 and use Active Directory to provide directory and authentication services to Windows-based clients. You also have Linux and UNIX installed in the same datacenter and you wish to use Active Directory as your central system for providing consistent directory and authentication services to these systems as well. You have a desire to standardize directory services using LDAP technology and authentication using Kerberos. Ultimately you want to have one centralized account entry and password for each user and group in Active Directory and have users and systems be able to leverage the account information and credentials from Windows, UNIX and Linux client systems. You might also be using Samba on your Linux and UNIX systems for sharing files with your Windows systems and wish to leverage Active Directory for centralized user account information and authenticated access to shared files.

Solution Overview

This solution uses the industry-standard LDAP protocol for directory services and Kerberos for authentication. Since Microsoft Active Directory uses LDAP for directory services and most Linux and UNIX systems include LDAP software, it is logical to leverage this popular modern protocol for managing centralized user, group and system information. With the addition of Microsoft's Identity Management for UNIX component that is available for Windows Server 2003 R2 and Windows Server 2008 customers now have a Microsoft supported method for storing UNIX-specific user and group information in Active Directory. This is accomplished through the addition of UNIX schema extensions for user and group information in Active Directory and an additional UNIX Properties tab in the management tool for user and group properties.

Likewise, Kerberos is a standard technology for providing authentication across most modern operating systems and is the central authentication system for Active Directory. This is the centralized system we will use for storing and using one set of user passwords in Active Directory that can be used by both Windows and UNIX/Linux clients.

Finally, since many users need a simple way to share files between Windows and UNIX/Linux, we include instructions on how to set up the popular Open Source project, Samba, for SMB file serving from UNIX/Linux systems. This allows Windows users to use an authenticated "single sign-on" method for storing and retrieving files on UNIX/Linux systems using the native CIFS file sharing protocol for file. In addition, Samba is used to join the UNIX or Linux system to the Active Directory domain. Our Samba instructions leverage the underlying LDAP/Kerberos infrastructure in order to provide consistent centralized passwords and user / group information.

This solution will allow users to log into UNIX and Linux systems using their Windows account name and password via Active Directory. In this solution, UNIX specific information such as the default UNIX shell is stored centrally in Active Directory to simplify the management of users across multiple systems. This also means that a UNIX user will have a single consistent UID across all UNIX systems that use this setup. This solution can also lay the groundwork for enabling true single sign-on capabilities through the use of Kerberos-enabled utilities on UNIX.

Solution Implementation Details

To implement this solution we used Microsoft Windows Server 2003 R2 Standard Edition with Service Pack 2 as the

server environment. This solution should work equally well on Windows Server 2008. The server was configured with the following capabilities installed and enabled:

- Active Directory Domain Controller
- DNS Server
- DHCP Server
- Active Directory Services: Identity Management for UNIX

On the client side, we installed Fedora Core Linux 6. The key packages that were leveraged for this solution include:

- krb5-libs
- pam_krb5
- krb5-workstation
- openldap-*
- nss_ldap
- samba-client-3.0.23
- samba-common-3.0.23
- samba-3.0.23

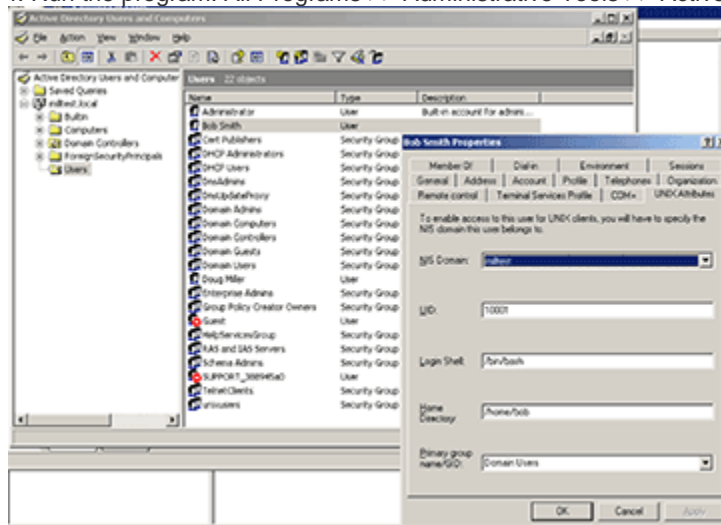
Normally, these components are installed by default on Fedora Core. For other systems, you may need to install Kerberos, LDAP and Samba software separately. It is also important to ensure that you are running a recent version of Samba and that the Samba daemon – *smbd* – has been built with Kerberos, LDAP and Active Directory support. The “[Samba by Example](#)” guide has details on how to ensure that your Samba software has the appropriate services built-in. This technote assumes that you are familiar with the basic administration concepts for both Windows Server and Fedora Core Linux.

Setting up Windows Server

We begin by setting up the environment on the Windows Server system. Complete the following steps on your Windows Server system:

1. Install Windows Server and accept the standard options during the install.
2. Using the “Manage Your Server” wizards, set up the server as a first server which means installing and configuring the system as an Active Directory Domain Controller, DNS Server and DHCP Server. For this implementation, we called our domain “milltest”.
3. Using the “Add or Remove Programs” tool in the Control Panel, add the component, Active Directory Services: Identity Management for UNIX. Be sure that “Server for NIS” is installed as part of this package. This will ensure that the Active Directory schema is set up correctly for storing UNIX attributes and will enable the “UNIX Attributes” tab in the properties page for users and group within the “Active Directory Users and Computers” MMC. The Server for NIS program itself is not used and does not need to be configured or turned on.

4. Run the program: All Programs >> Administrative Tools >> Active Directory User and Computers.



For this example implementation, create a Security Group called “unixusers” and open the properties for this group. Click on the UNIX Attributes tab for “unixusers” and select the default domain name for your NIS Domain (which should be the same as your Active Directory domain) and enter a GID such as 10000. Click OK to save your changes to this new group. You could also enable an existing group such as “Domain Users” however some UNIX systems are not able to handle group names beyond a certain length or ones that have spaces in the name.

5. For each Active Directory user that you want to enable for UNIX logins, you will need to configure their UNIX attribute properties. For example, create a new user called Bob Smith with a user logon name of “bob”. Give “bob” a password and uncheck the “User must change password at next logon” option. Open the properties for user “Bob Smith” and select the UNIX Attributes tab. For the NIS Domain field, select the default domain name. The rest of the attributes should be automatically filled in; however you can change these to suit your needs. For example, you could change the Login Shell to “/bin/bash”. The primary group name should be set to “unixusers” or another group that is UNIX enabled. In our example, we set the primary group to “Domain Users” since Fedora Core Linux supports long group names and group names with spaces.

6. Finally, we need to create a special user account for handling LDAP requests from the UNIX/Linux clients. We used an account name of “ldap” and made the account a member of the “Domain Guests” group. This account does not need any special administrative privileges. We set the user password to “not24get!” and made sure that in Account options, “User cannot change password” and “Password never expires” are checked and all other options are unchecked.

7. There are no special steps required to enable Kerberos services for UNIX clients.

Your Windows Server system is now ready to provide LDAP directory services and Kerberos authentication services to UNIX and Linux clients.

Setting up Fedora Core Linux

The next step is to set up a Linux system as a client for Active Directory services. In this case, the Linux client can be either a Linux desktop or server. The term “client” refers to how the Linux system is configured to use Active Directory client services from a Windows Server system. In a previous technote we used Fedora’s graphical administration tools configure the system. For this tech note, we will document how to configure the system by manually editing

system files since this method requires more customization of certain system files. Complete the following steps on your Fedora Core Linux system:

1. Install a standard version of Fedora Core Linux. We used most of the defaults for our install. Make sure that the Kerberos (krb), Open LDAP and Samba packages are installed on your system.
2. Use DHCP to get an IP address from Windows Server by providing the IP address for the Windows Server system when prompted for the DHCP server. Alternatively, you can hard code an IP address in the same network as the Windows domain.
3. Make sure the DNS client is set up to use the Windows Server domain controller (normally automatic if DHCP is used). Check the `/etc/resolv.conf` file once you have installed Linux to be sure that DNS is configured correctly to use your Active Directory domain controller as a “nameserver” and that the Active Directory domain name is defined after the “search” parameter.
4. Make sure you have set the system host name for your Linux system. For example edit the file `/etc/sysconfig/network` and ensure that the `HOSTNAME` variable is set with your host name.
5. Also be sure to configure `/etc/hosts` so that your fully qualified domain name and host name are defined with either the local address or an IP address if using a static IP address. This should be the first entry in the definition list even if you currently have localhost defined for the local interface. You should also include a definition for the name and IP address for the Active Directory server being used for authentication.
6. Configure your Linux system to use the Network Time Protocol and set it to use the Windows Server IP address as the first NTP Server. This is done on Fedora by running System à Administration à Date & Time. It is essential to have the Linux system clock synchronized with the Windows Server clock, otherwise Kerberos authentication will not work correctly.
7. The file `/etc/pam.d/system-auth` needs to be configured to use both UNIX and Kerberos methods for authentication. It is also useful to add an extra “session” entry to enable the creation of home directories on first login to the Linux system by a user. We generated the following file by using the graphical Authentication tool included with Fedora (System >> Administration >> Authentication) and adding the `pam_mkhome` entry in the session section:

```
auth required pam_env.so
auth sufficient pam_unix.so nullok try_first_pass
auth requisite pam_succeed_if.so uid >= 500 quiet
auth sufficient pam_krb5.so use_first_pass
auth required pam_deny.so

account required pam_unix.so broken_shadow
account sufficient pam_localuser.so
account sufficient pam_succeed_if.so uid < 500 quiet
account [default=bad success=ok user_unknown=ignore] pam_krb5.so
account required pam_permit.so

password requisite pam_cracklib.so try_first_pass retry=3
password sufficient pam_unix.so md5 shadow nis nullok try_first_pass
use_authok
```

```
password sufficient pam_krb5.so use_authtok
password required pam_deny.so
```

```
session optional pam_keyinit.so revoke
session required pam_mkhome.so skel=/etc/skel umask=0022
session required pam_limits.so
session [success=1 default=ignore] pam_succeed_if.so service in crond
quiet use_uid
session required pam_unix.so
session optional pam_krb5.so
```

8. The first Kerberos file to manually configure is the file `/etc/krb.conf`. Add two lines at the beginning of the file to define your new Active Directory Kerberos realm. For example, if the domain is `MILLTEST.LOCAL` and the Active Directory server is `ws2003r2.milltest.local`, then you would insert the following lines at the beginning of the file:

```
MILLTEST.LOCAL ws2003r2.milltest.local:88
MILLTEST.LOCAL ws2003r2.milltest.local:749 admin server
```

9. The next Kerberos file to edit is `/etc/krb.realms`. Insert your domain name as the first line of the file. For example, using the settings in the previous step, you would add:

```
.milltest.local MILLTEST.LOCAL
```

10. The final Kerberos file to edit is `/etc/krb5.conf`. Ensure that the `[libdefaults]`, `[realms]` and `[domain_realm]` sections have the correct entries. Using our example settings above, these sections would have the following entries:

```
[libdefaults]
default_realm = MILLTEST .LOCAL
dns_lookup_realm = true
dns_lookup_kdc = true

[realms]
MILLTEST .LOCAL = {
kdc = ws2003r2.milltest .local:88
admin_server = ws2003r2.milltest .local:749
kpasswd_server = ws2003r2.milltest .local:464
kpasswd_protocol = SET_CHANGE
default_domain = true
}

[domain_realm]
*.addomain.local = MILLTEST .LOCAL
.addomain.local = MILLTEST .LOCAL
```

11. The next step is to configure LDAP on Fedora. The central LDAP file to configure is `/etc/ldap.conf`. This file needs to be set up correctly with information related to the Active Directory Windows Server (the first 5 lines) as well as map the schema to be used for accessing user and group account information in Active Directory. Note on lines 4 and 5 we define the "ldap" user that was created in Active Directory and we set the password that is used for that account.

Only the first 5 lines should be modified for your environment. The following file shows how we set up our LDAP environment:

```
host 192.168.0.10
base dc=milltest,dc=local
uri ldap://ws2003r2.milltest.local/
binddn ldap@milltest.local
bindpw not24get!
scope sub
timelimit 120
bind_timelimit 120
idle_timelimit 3600
nss_initgroups_ignoreusers root,ldap
referrals no
ssl no
nss_base_passwd dc=milltest,dc=local?sub
nss_base_shadow dc=milltest,dc=local?sub
nss_base_group
dc=milltest,dc=local?sub?&(objectCategory=group)(gidnumber=*)
nss_map_objectclass posixAccount user
nss_map_objectclass shadowAccount user
nss_map_objectclass posixGroup group
nss_map_attribute uid sAMAccountName
nss_map_attribute homeDirectory unixHomeDirectory
nss_map_attribute gecos cn
nss_map_attribute shadowLastChange pwdLastSet
nss_map_attribute uniqueMember member
pam_login_attribute sAMAccountName
pam_filter objectclass=User
pam_password ad
```

12. The next system that needs to be set up is the nsswitch configuration file `/etc/nsswitch.conf`. The changes to this file direct the system to use LDAP to look up user and group information from Active Directory in addition to using local files. The following lines should be updated with the LDAP directive:

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

If you are planning on also using Samba, then you will also need to include Samba's `winbind` for name resolution. The `/etc/nsswitch.conf` file would read:

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

13. At this point, we have everything in place to enable logging into a Linux system using an Active Directory username and password and leveraging Active Directory, via Kerberos and LDAP, for directory services and authentication. If you had no plans to use Samba for file and print serving to Windows clients from the Linux server, then you would need no additional steps. However, adding Samba to the mix has advantages beyond enabling file

and print sharing. The main advantage is that by joining your Linux system to the Active Directory domain using Samba allows you to create a computer account for the Linux system in Active Directory and secure the Kerberos relationship between the two systems which will help to prevent Kerberos authentication spoofing. The bottom line is your environment will be more secure, so for this reason we highly recommend completing the last steps to enable Samba, even if you have no plans to use file and print serving from Linux.

14. The final file to set up is the Samba `smb.conf` file. This file is the main configuration file for Samba-3 and there are many possible combinations of settings that can be used to customize your Samba environment. Refer to the [Samba-3 documentation](#) for details on the various settings. For our environment, we want to enable our Fedora system as an Active Directory client, with Kerberos authentication support and use the LDAP services from Active Directory to provide consistent uid and gid mapping across all Linux machines in the domain. Since Active Directory user and group names will be enumerated when we use the `getent` command via the LDAP entries in `/etc/nsswitch.conf`, we do not need to turn on user and group enumeration via `winbind` – which will improve performance. We also want to automatically share out the home directory for Active Directory users who have logged into the Fedora system. This is an optional setting. Using our test domain settings, the `/etc/samba/smb.conf` file would have the following entries:

```
[global]
unix charset = LOCALE
workgroup = MILLTEST
netbios name = FEDORA100
realm = MILLTEST.LOCAL
server string = Fedora_6
security = ADS
use kerberos keytab = Yes
idmap backend = ad
ldap idmap suffix = dc=milltest,dc=local
ldap admin dn = cn=ldap,cn=Users,dc=milltest,dc=local
ldap suffix = dc=milltest,dc=local
idmap uid = 500-100000000
idmap gid = 500-100000000
log level = 1
syslog = 0
log file = /var/log/samba/%m
printcap name = CUPS
winbind use default domain = yes
winbind nested groups = Yes

[homes]
comment = Home Directories
valid users = %D\%U
read only = No
browseable = No
```

15. The final steps to enable this configuration are to reset the Samba environment and join the Linux system to the Windows Active Directory domain. In order to complete these steps, you may want to create and run a shell script with the following commands:

```
service smb stop
service winbind stop
rm -f /etc/samba/*tdb
rm -f /var/cache/samba/*tdb
```

```
rm -f /var/cache/samba/*dat
echo "Enter the password for the Active Directory Administrator"
net ads join -U Administrator
smbpasswd -w not24get!
service winbind start
service smb start
chkconfig smb on
chkconfig winbind on
```

Note on line 8 we use the “smbpasswd –w” command to store the ldap user password in the Samba secrets.tdb file.

Your Fedora Linux system should now be configured to use Active Directory for user and group information directory services and Kerberos authentication using the KDC on Windows Server. In order to test the setup, use the “getent passwd” command to confirm that you can see both local users that exist in /etc/passwd and Windows domain users from Active Directory.

You should also test whether the setup is working correctly by logging in to the Linux system using a standard Active Directory account name and password. If the user has no home directory on the system, it should be automatically created for them. If the login fails, double check that the clocks are synchronized to the same time on both Windows and Linux. If the clocks are off by even a few minutes, then Kerberos will not work correctly.

Summary

By implementing this solution you gain the same benefits described in [the previous NIS-based tech note](#). You will substantially reduce the overhead for account and authentication management for heterogeneous Windows and Linux environments. Users now have one account name and password to remember for logging into both Windows and Linux. Administrators can setup new user accounts for both Windows and Linux by simply adding a single new account entry into Active Directory. Administrators can also deny access to all systems for any user by simply disabling their Active Directory account. Policy settings such as password complexity and length requirements are now enforced for users on both Windows and Linux. Deploying new Linux systems is simplified since administrators no longer need to set up individual user accounts for every new Linux machine. While this method does involve making some changes to the Window Server Active Directory system, the changes are minimal and are fully supported by Microsoft. Given that this method uses the more modern LDAP protocol instead of the older NIS protocol for directory services, this method is recommended over the other methods. In addition, you get the added bonus of enabling Samba for secure file and print sharing between UNIX/Linux systems and Windows systems.

There are two other methods for solving the Windows/UNIX/Linux Active Directory integration challenge that are presented in this series. If you wish to use NIS instead of LDAP for storing user information, you should read the tech note titled: “[Using Server for NIS, IdMU and Kerberos for Unix/Linux directory and authentication services](#)”. If you wish to use a more limited Samba setup for integration with Active Directory, you should read the tech note titled: “[Using Samba client technology and Kerberos for AD-based identity management](#)”. For an overview of the pros and cons of each method, see the tech note: “[An introduction to Active Directory integration for Unix and Linux systems](#)”.

If you have comments on these tech notes or wish to provide enhancements or corrections, please feel free to post a question in the [Interop Community](#) forums.