

Using Samba client technology and Kerberos for AD-based identity management

Tech Note #3 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 and use Active Directory to provide directory and authentication services to Windows-based clients. You also have Linux and Unix installed in the same enterprise and you wish to use Active Directory as your central system for providing consistent directory and authentication services to these systems as well. You currently use Samba on your Linux and Unix systems for sharing files with your Windows systems. 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.

Solution Overview

This solution uses the popular Open Source project, Samba, for user and group information look-ups. In addition, Samba is used to join the Unix or Linux system to the Active Directory domain. Samba-3 is a standard package in most Linux and Unix systems and the source code for Samba can be found at samba.org. Since Samba-3 designed to be able to talk directly to Active Directory, no special software is required on the Active Directory server in order to serve Samba-3 clients. Kerberos is used for authentication and password validation in this solution. Kerberos is the default system for authentication for modern Windows domains and is also widely used and included in most Unix and Linux systems. 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 on each Unix server so this solution is not quite as centralized as the other solutions presented in this series of tech notes. 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 with Windows Server 2008. The server was configured with the following capabilities installed and enabled:

- Active Directory Domain Controller
- DNS Server
- DHCP Server

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
- samba-client-3.0.23
- samba-common-3.0.23
- samba-3.0.23

Normally, Samba is installed by default on Fedora Core. For other systems, you may need to install 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 tech note 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 2008 or Windows Server 2003 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.local”.
3. There are no specific steps that need to be done in Active Directory in order to enable or configure Unix / Linux users to login using their Active Directory credentials. If a user is configured to allow logins to Windows clients then they are also now enabled to login to any Unix or Linux machine that has been joined to the Active Directory domain using Samba. Details on joining the domain are provided below.
4. There are no special steps required to enable Kerberos services for Unix clients.

Your Windows Server system is now ready to provide Active 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 the previous tech note 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. Both methods can accomplish the same result. Choose the method that you are most comfortable with. 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) 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 you 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 names 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 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
```

8. 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
```

9. 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
}
[domain_realm]
*.addomain.local = MILLTEST .LOCAL
.addomain.local = MILLTEST .LOCAL
```

10. 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 Samba's `winbind` to look up user and group information from Active Directory in addition to using local files. The following lines should be updated with the `winbind` directive:

```
passwd: compat winbind
group: compat winbind
hosts: files dns winbind
```

11. It is useful to add an extra entry to the file `/etc/pam.d/system-auth` to enable the creation of home directories on first login to the Linux system by a user. In the session section of this file towards the end, add the line:

```
session required pam_mkhomedir.so skel=/etc/skel umask=0022
```

12. 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 support and use `idmap_rid` to provide consistent uid and gid mapping across all Linux machines in the domain. We also want to allow all user and group names to be enumerated when we use the `getent` command. If you have a large domain, you will want to set these setting to “No”. We also want to automatically share out the home directory for Active Directory users who have logged into the Fedora system. Using our test domain settings, the `/etc/samba/smb.conf` file would have the following entries:

```
[global]
unix charset = LOCALE
workgroup = MILLTEST
netbios name = FEDORA6
realm = MILLTEST.LOCAL
server string = Fedora_6
security = ADS
allow trusted domains = No
idmap backend = idmap_rid:MILLTEST=500-100000000
idmap uid = 500-100000000 idmap gid = 500-100000000
log level = 1
syslog = 0
log file = /var/log/samba/%m
max log size = 50
template shell = /bin/bash
template homedir = /home/%U
winbind use default domain = yes
winbind enum users = Yes
winbind enum groups = Yes
winbind nested groups = Yes
printcap name = CUPS printing = cups

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

13. 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  
service winbind start  
service smb start  
chkconfig smb on  
chkconfig winbind on
```

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 now 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 and that all services mentioned above, such as winbind, are enabled. 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 as described in the previous [NIS-based tech note](#). You will substantially reduce the management 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. The main difference for this solution vs. the other solutions presented in this series is that you are not required to make any changes on the Windows Active Directory domain controller. If the Active Directory system is set up to authenticate users on Windows clients then it is also ready to authenticate Linux and Unix users with Active Directory using the Kerberos and Samba-3 packages that come with most Unix and Linux 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 Samba for storing user information then 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 LDAP for storing user information then you should read the tech note titled: “[Using native LDAP, native Kerberos and Windows Server Active Directory services and schema for cross-platform identity management.](#)”

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.