ActiveDirectorySamba

This simple guide is a mostly accurate way to set up a **Samba** machine as a domain member in a **MicrosoftWindows** 2000 or Windows 2003 **ActiveDirectory** domain. For a REALLY short version, tested with Win2k3, see the Quick 'n' Dirty instructions at the bottom of the page.

Samba as an Active Directory Domain Member

The following setup is used:

192.168.0.1	**test1.thinclient.test.org**	the AD server, hereafter known as "the server"
192.168.0.209	**mail.thinclient.test.org**	samba3 "client" machine

The Samba system is based upon a stock standard **RedHat** 9 system with the samba software upgraded to Samba3 (using RPM)

The following steps are needed to get the system functioning:

1. install and check necessary packages
2. configure name resolution using either dns or a hosts file
3. configure samba and winbindd
4. configure kerberos
5. testing Samba and winbindd
6. good luck

Install and Check necessary packages

The following packages are required to sucessfully run all the commands detailed in this guide:

Samba:

1. redhat-config-samba (or system-config-samba)
2. samba-common
3. samba-client
4. samba

Kerberos:

1. pam_krb5

2. krb5-workstation
3. krb5-libs
4. krbafs

You can query if these packages are installed by running:

```
rpm -q package-name
```

Configure name resolution

Active Directory relies HEAVILY on DNS to resolve not only host names but services they provide as well. To set up DNS on the Linux box, see the DNSHowTo, otherwise consult necessary Windows documentation on setting up forward AND reverse DNS zones.

The first step is to configure name resolution for our systems. The kerberos authentication system, which we will configure later on, requires us to be able to do a reverse lookup on an IP address to get a fully qualified domain name (FQDN).

There are two ways to do this. The cheap and nasty method is to use a hosts file on both systems. Hosts based authentication, which is discussed here, is ugly and hacky, and should be avoided at all costs. If you want to do it anyway, you need entries similar to the following.

Samba machine

/etc/hosts

```
  127.0.0.1       localhost localhost.localdomain
  192.168.0.1     test1.thinclient.test.org test1
  192.168.0.209   mail.thinclient.test.org mail
```

Windows Active Directory server (see footnote 1)

%Systemroot%\System32\drivers\etc\hosts

```
  127.0.0.1       localhost localhost.localdomain
  192.168.0.1     test1 test1.thinclient.test.org
  192.168.0.209   mail mail.thinclient.test.org
```

The correct method is to setup DNS on the server which can be done through the DNS console in the Administrative Tools section of Windows 2000/2003 Server. (You shouldn't be runing an Active Directory without a well set up DNS; if you don't know how to do it, go away and learn RIGHT NOW). We won't go into the details of setting this up here, but we will specify the Linux side of that here.

A good way to set this up is to have a Linux-based BIND server doing name resolution for your site 'mydomain.tld', just as you normally would; then configure BIND to delegate the special Active Directory

sub-domains DomainDnsZones.mydomain.tld and so on to the Windows Server 2003 box. Then, configure Windows Server 2003 DNS to be a caching proxy using the Linux BIND box as its parent, except for the AD sub-domains for which it should be authoritative. All machines can then use the Linux box for DNS. This way, name resolution of normal names stays on good ole reliable Linux where it belongs, the Windows Active Directory crud goes on Windows where it belongs, and everything's happy. If the Windows Server is down, the AD stuff stops working (there's no avoiding that if the PDC is offline); however normal (non-AD) name resolution is unaffected. Thanks to Matthew Sanderson for the tip.

/etc/resolv.conf

```
  search      thinclient.test.org
  domain      thinclient.test.org
  nameserver  192.168.0.1
```

Configure Samba3 and Winbindd

This part is the easy one, we just create ourselves a default Samba configuration with at least the following entries (Note this is a completely empty and default configuration file, and you may wish to add more. A file share would be handy to add).

```
/etc/samba/smb.conf
  [global]
    # general options
    workgroup = THINCLIENT
    netbios name = MAIL

    # winbindd configuration
    # default winbind separator is \, which is good if you
    # use mod_ntlm since that is the character it uses.
    # users only need to know the one syntax
    winbind separator = +

    # idmap uid and idmap gid are aliases for
    # winbind uid and winbid gid, respectively
    idmap uid = 10000-20000
    idmap gid = 10000-20000
    winbind enum users = yes
    winbind enum groups = yes
    template homedir = /home/%D/%U
    template shell = /bin/bash

    # Active directory joining
    # "ads server" is only necessary if your kdc
    # can't be located using /etc/krb5.conf -- JamesSpooner
    #
    # Note that more recent Samba versions have renamed "ads server"
    # to "password server", so if /var/log/messages reports
    # 'Unknown parameter encountered: "ads server"' on restart,
    # change 'ads' to 'password' -- ChetHosey
    #
    # ads server = test1.thinclient.test.org
    security = ads
    # encrypt passwords = yes is now default in Samba3 -- Enigma
```

```
encrypt passwords = yes
realm = thinclient.test.org
# this handles the "ads server = " directive as well -- Enigma
password server = test1.thinclient.test.org
```

NB: The important things to pay attention to here are the name of our samba machine (netbios name), the workgroup, and the **ActiveDirectory** stuff.

Configure Kerberos5

If your Kerberos setup is good, run `net ads join -U Administrator%password` and it will perform all the `ktpass` and `ktutil` stuff on the fly as mentioned in the SAMBA howto . Then you can skip to the winbind section below. Thanks to EnigMa**?** for the tip. If you don't specify %password, it will prompt you on the command line (for the security minded).

Configuring a Kerberos setup is much easier in the long run then generating the key and importing it.

Manual approach

We need to generate a key for our samba machine on the Windows server, and securely import this into our samba machine. To create the keyfile we run the following on the Windows server:

```
ktpass -princ host/mail.thinclient.test.org@THINCLIENT.TEST.ORG \
          -mapuser MAIL -pass MAIL1234PASSWORD -out mail.keytab
```

This, and many other tools for managing Kerberos in Windows 2000, are located in the support tools which are directly downloadable from Microsoft. Thanks to Jan Gerle for the tip.

We then transfer the mail.keytab securely to our samba machine by using something similar to SSH or another secure means. And then on the samba machine we will import the keyfile we just generated by using the ktutil program, which is part of the kerberos distribution. The unix commands for ktutil are as follows:

```
% ktutil
ktutil: rkt mail.keytab
ktutil: list
ktutil: wkt /etc/krb5.keytab
ktutil: q
```

See **ActiveDirectoryKerberos** on setting up Kerberos to talk to **ActiveDirectory**.

(Re)starting Samba and Winbindd

First we test our samba configuration and our winbind settings, before we modify our samba startup script.

```
/etc/rc.d/init.d/samba restart
```

```
/usr/sbin/winbindd
```

For some of our paranoid friends, we can check to see if our winbindd is actually running using

```
ps fax | grep winbindd
```

Now for a real test, and see if we can get some information off our Active Directory PDC.

```
/usr/bin/wbinfo -u
```

And we should get a list of users in the format THINCLIENT+<username>

```
THINCLIENT+Administrator
THINCLIENT+Guest
..
```

And we can do the same for our list of groups.

```
/usr/bin/wbinfo -g

THINCLIENT+Domain Admins
THINCLIENT+Domain Users
THINCLIENT+Schema Admins
..
```

We can now use the getent utility to get a unified list of both the local and PDC users and groups. These utilities will generate a list of data similar in format to the /etc/passwd and /etc/group files respectively.

add following entries in nssswitch.conf:

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

if you are compiling samba from source then you need to copy following files manually

```
cp /usr/src/samba-3.0.1/source/nsswitch/pam_winbind.so  /lib/security/
cp /usr/src/samba-3.0.1/source/nsswitch/libnss_winbind.so /lib/
cp /usr/src/samba-3.0.1/source/bin/pam_smbpass.so  /lib/security/
```

then run following command to get unified entries

```
/usr/bin/getent passwd
/usr/bin/getent group
```

It is now a good idea to test to ensure your Active Directory usernames are valid on the system. Try the following:

```
chown "THINCLIENT+username" filename
```

(where THINCLIENT is the active directory short name)

If 'wbinfo -u' and 'getent passwd' work fine but your chown says this is an unknown user, you probably have NSCD running. You should disable NSCD and restart winbind. (See http://us4.samba.org/samba/docs/man/winbind.html#id2958310 for more)

After this we can fix up our init.d startup scripts to automate the startup of winbindd and not start NSCD.

Configure PAM and Winbind

Before we do anything at all here, we need to make a backup of our /etc/pam.d/* files. And have a linux bootdisk available if possible. If anything goes wrong here, you may not be able to login to your system properly. (So don't reboot or logoff to test, but use a text console)

To have our **ActiveDirectory** users be able to login to our we have to modify our /etc/pam.d/login. We don't need to modify our /etc/pam.d/samba settings as it is already configured for winbind.

/etc/pam.d/login

```
#%PAM-1.0
auth        required     pam_securetty.so
auth        sufficient   pam_winbind.so
auth        sufficient   pam_unix.so use_first_pass
auth        required     pam_stack.so service=system-auth
auth        required     pam_nologin.so
account     sufficient   pam_winbind.so
account     required     pam_stack.so service=system-auth
password    required     pam_stack.so service=system-auth
session     required     pam_stack.so service=system-auth
session     optional     pam_console.so
```

After we save this file, we should now be able to login to our linux machine with the username THINCLIENT+Administrator, and get ourself a login prompt. Now the system may complain if you do not have the specified home directory created (in this case /home/THINCLIENT/Administrator)

SSH Support

Do the same additions that you made to /etc/pam.d/login to /etc/pam.d/sshd to support logins via SSH.

Have fun

And congrats it works, if you want to configure further items such as mail and other things you may need to modify the apropriate PAM modules, and isn't covered here.

References

- Using Kerberos Clients section of the Microsoft : Step-by-Step Guide to Kerberos 5 (krb5 1.0) Interoperability
- Authentication to ADS
- The winbindd and Active Directory Domain Member sections of the Samba v3 Documentation

---

Quick 'n' Dirty setup for Samba 3 and Windows 2003

These are the absolute bare minimum steps to get your Samba server integrated as a member server in an AD controlled domain with Win2k3 as the DC.

1. ENSURE your samba box has an A record and associated PTR in DNS.

2. On your DC, disable signing: Run Domain Controller Policy tool and edit Account Policies -> Security Options -> Microsoft network client: Digitally sign communications (always) Set this to Disabled. Do the same in the Domain Policy tool. Note, you will need to reboot the server for this step, though it won't tell you to. Disable on your samba server as well with the following in smb.conf

*Note (PvtJoker**?**):* In my experience that wasn't needed, this tutorial concentrates on windows 2003, and works without disabling these options.

```
        client signing = no
        client use spnego = no
```

3. On your samba server, install kerberos5, and edit /etc/krb5.conf. It should contain:

```
[libdefaults]
        default_realm = YOUR.ADS.DOMAIN
        dns_lookup_kdc = false
        dns_lookup_realm = false

[domain_realm]
        .your.domain.name=YOUR.ADS.DOMAIN
        your.domain.name=YOUR.ADS.DOMAIN

[realms]
YOUR.ADS.DOMAIN = {
        default_domain = your.domain.name
        kdc = IP.OF.THE.DC
}
```

4. Ensure smb.conf contains

```
realm = YOUR.ADS.DOMAIN
workgroup = YOUR
security = ADS
```

5. Get a ticket using kerberos: kinit administrator (enter the administrator password when prompted). The klist command should then list a ticket.

6. Join the domain using 'net ads join'. This should use the credentials in your kerberos ticket.

7. Set up winbind - ensure the following is in smb.conf

```
winbind uid = 10000-20000
winbind gid = 10000-20000
winbind enum groups = yes
winbind enum users = yes
```

8. store your winbind credentials with wbinfo --set-auth-user=DOMAIN\\administrator%password

NOTE: This step may fail with one or more of the following errors:

```
could not obtain winbind separator!
could not obtain winbind domain name!
```

Should you receive either or both errors, it is because winbind is not currently running continue with the remaining steps and return to this step after winbind has been started.

9. modify /etc/pam.d/samba (on woody) or the appropriate pam file to add "sufficient" for auth and account using pam_winbind.so. These need to go BEFORE the pam_unix.so calls for samba. My /etc/pam.d/samba is as follows:

```
auth            sufficient      pam_winbind.so
auth            required        pam_unix.so nullok
account         sufficient      pam_winbind.so
account         required        pam_unix.so
session         required        pam_unix.so
password        required        pam_unix.so
```

10. Modify /etc/nsswitch.conf with the following:

```
passwd:         winbind compat
group:          winbind compat
shadow:         winbind compat
```

11. Restart samba and winbind.

12. All should work. :) Browse your server and see...

Samba and software deployment

Software deployment is a useful feature of a domain controller, as it allows to distribute software to many clients - and thus, the administrator doesn't have to walk from one workstation to another (10, 20, ... 100 machines...) to install the same piece of software (and uninstall it or upgrade a couple of days later).

One common misconception when comparing Samba to Active Directory, is that with Samba you can't deploy software to your Windows workstations. Another misconception, this time about Active Directory, is that with AD you can deploy software to your workstations. So, what's this all about?

Active Directory can only deploy packages in MSI format. This isn't very widely used; mostly software is available in EXE format.

With Samba, as in whole *NIX philosophy, one tool does the job, but does it well.

To distribute software with Samba, one can use WPKG - with this tool, you just configure the software which should be installed / upgraded / uninstalled on a given machine or a group of machines - and next time these Windows workstations are booted, the software you specified is installed / upgraded / uninstalled automatically.

Footnotes

1. %Systemroot% is a variable set by Windows NT and onward to mean "the location where Windows is installed", ie c:\winnt, c:\windows, etc.

CategoryInteroperability

Last edited on Monday, April 17, 2006 7:05:37 pm by "MichaelGronlund"

lib/ExternalReferrer.php:86: Notice: Array to string conversion (...repeated 9 times)